ADAPTIVE NEURO-FUZZY INFERENCE BASED PATTERN RECOGNITION STUDIES ON HANDWRITTEN CHARACTER IMAGES

A THESIS SUBMITTED BY

LAJISH. V.L.

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

IN

COMPUTER SCIENCE

UNDER THE

FACULTY OF SCIENCE

UNIVERSITY OF CALICUT
KERALA - 673 635
INDIA

FEBRUARY 2007

Dedication

To My Beloved Father Who Taught Me
the Rudiments of Mathematics
&
To My Loving Mother

CERTIFICATE

This is to certify that the thesis entitled "**ADAPTIVE NEURO-FUZZY INFERENCE BASED PATTERN RECOGNITION STUDIES ON HANDWRITTEN CHARACTER IMAGES**" is a report of the original work carried out by Mr. Lajish.V.L. under my supervision and guidance in the Computer Vision and Intelligence Research Centre, Post Graduate Department of Physics, Government College, Madappally, University of Calicut, Kerala and that no part thereof has been presented for the award of any other degree.

Dr. N.K. Narayanan
Professor & Chairman
School of Information Science & Technology
Kannur University
KERALA.

Kannur 670 567
February 14, 2007.

DECLARATION

I hereby declare that the work presented in this thesis is based on the original work done by me under the supervision of Dr. N.K. Narayanan in the Computer Vision and Intelligence Research Centre, Post Graduate Department of Physics, Government College, Madappally, University of Calicut, Kerala and that no part thereof has been presented for the award of any other degree.

Calicut 673 635,
February 14, 2007.

Lajish.V.L.

ACKNOWLEDGEMENTS

This thesis is the outcome of the education, guidance and inspiration I received from my research guide Dr.N.K..Narayanan, Professor & Chairman, School of Information Science & Technology, Kannur University, Kerala. I wish to express my deep gratitude to him for the insightful guidance and his invaluable help in steering the course of this study.

I am extremely grateful to Mr.K.Suresh Babu, Head of the Department and Dr.P.Ramakrishan, Computer Vision & Intelligence Research Centre, Post Graduate Department of Physics, Government College, Madappally, Kerala for their encouragement, helpful suggestions and constructive criticisms on this work.

I am indebted to Dr.P.Nagabhushan, Professor & Chairman, Department of Studies in Computer Science, University of Mysore, Karnataka who always extended unconditional support and motivation throughout this work. I would also thank Dr.G.Hemantha Kumar and Dr.D.S.Guru of the same Department for many enlightening discussions.

I thank Dr.Sargur N.Srihari, Distinguished Professor, University of Buffalo, New-York, Dr.Sankar K.Pal, Director, Indian Statistical Institute, Kolkata and Dr.K.Rajan, Principal Research Scientist, Indian Institute of Science, Bangalore for extending their valuable suggestions and support during the tenure of this work.

My special thanks are due to my colleagues Dr.R.K.Sunilkumar, Mr.P.Prajith and Mr.V.Kabeer for their wholehearted co-operation throughout this endeavor.

I am obliged to the Kerala State Council for Science, Technology & Environment, Government of Kerala, for providing financial assistance for carrying out this study.

TABLE OF CONTENTS

ABSTRACT

ADAPTIVE NEURO-FUZZY INFERENCE BASED PATTERN RECOGNITION STUDIES ON HANDWRITTEN CHARACTER IMAGES

BY

LAJISH.V.L

This study investigates the potential use of state-space parameters and zoning based vector distance information for handwritten character modeling and the credibility of Adaptive Neuro-Fuzzy Inference System (ANFIS) for pattern recognition applications. To start with, an image database for Malayalam handwritten character set consisting of eighty one elements is designed and developed. The database consists of a total of 57,996 BMP formatted and, 128x84 pixel sized image samples collected from 716 writers. The whole set of available data is split up into two categories namely *training set* and *test set* in 1:1 ratio. The database developed in this work is the first of its kind in Malayalam. Among the eighty one elements present in the database, forty four basic characters are selected for the recognition studies in this thesis.

In the pre-processing stage the handwritten character images are binarized using Otsu's global thresholding technique. The size normalization is also performed for unifying the character patterns into a standard 66x42

pixel size using Affine Transformation and Bilinear Interpolation algorithm. The size normalized binary images are then skeletonized using a rotation invariant rule based thinning algorithm.

In feature extraction stage, the handwritten character images are modeled using state-space parameters extracted directly from gray-scale images with out the usual step of binarization. For this purpose, the reconstructed state-space of each character image and the scatter plot of its trajectory matrix named State- Space Map (SSM) are generated. Then the State-Space Point Distribution (SSPD) parameters are extracted from the SSMs of the collected samples. These features are found to be promising and used later in the recognition experiments.

In the next stage, the shape analysis of the Malayalam handwritten characters is performed on the basis of existence ratio, aspect ratio and the centroid of each character pattern. From this study it is found that the region decomposition or zoning is the simplest and most suitable method to identify the information content regions in Malayalam handwritten characters. Then the Zoned Vector Distance (Z-VD) features and Fuzzy-Zoned Normalized Vector Distance (FZ-NVD) features are extracted from the binary skeletons of the pre-processed character images.

The cluster analysis based on *c*-Means clustering technique is performed on the extracted feature vectors and this clustering method is further used for classification purpose. The *k*-Nearest Neighbour statistical algorithm is also used for recognition purpose and the results indicate the credibility of the extracted features. Classification studies are also performed by means of Class-modular Multilayer Feed-forward Neural Network with Error Back-propagation learning algorithm. By comparing the recognition results obtained using the above classifiers, it is observed that the FZ-NVD parameters provide better recognition accuracy for Malayalam handwritten characters. These features are used further in the recognition experiments. The recognition results also indicate that the performance of the above used classification techniques are moderately adequate but necessitate definitely a modified combined classifier approach.

In the next phase of this study, the recognition experiment is conducted based on FZ-NVD features using Adaptive Neuro-Fuzzy Inference System (ANFIS). For this purpose fuzzy *if.....then* rules are framed from the FZ-NVD parameters. This is achieved by the fuzzification of each attribute of the feature vector using three fuzzy sets SMALL, MEDIUM and LARGE defined by *Gaussian* membership functions. This fuzzy rule set definitely has the capacity to handle the immense variations in distinct writing styles of individuals.

In this study, we used *Sugeno* fuzzy model (TSK model) for modeling the inference part of ANFIS. To perform parameter adjustments in learning, ANFIS needs partitioning of the input space according to the target classes. In order to facilitate this process a pre-processing operation is performed on the entire rule set using ID3 decision tree induction procedure. The reduced set of optimum rules obtained as a result of this pre-processing is given as input to the inference part of ANFIS. A gradient based learning algorithm, proposed by Jang.J-S.R is effectively used to estimate the parameter set in learning.

In the present study, we used a set of 15,752 samples of the forty four Malayalam handwritten characters collected from 358 different writers for training and a disjoint sample dataset of same size for testing. The entire system is developed and implemented using MATLAB 7. The recognition accuracy obtained for forty four Malayalam handwritten characters based on fuzzy-zoned normalized vector distance (FZ-NVD) parameters using the proposed ANFIS approach is found to be promising. Our research findings on handwritten character recognition, employing methodologies and algorithms on adaptive network based fuzzy inference system, are the first of its kind in Malayalam. Further it establishes that the neuro-fuzzy approach, a key component of *soft computing,* provides flexible information processing capability as regards to the recognition of ambiguous patterns.

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Background

One of the most impressive capabilities of human brain is the ability to recognize patterns in nature. Throughout the history of human search for knowledge, pattern recognition has been one of the main rationales behind the vast range of theories and concepts that we developed to understand natural world. The major goal of pattern recognition research is to make computers easier to communicate with people in a less constrained manner and thus to make its benefits available to the society. As a result machine simulation of human reading has been a subject of intensive pattern recognition research in the last three decades. In the infant stages of this research, investigations were limited by the memory and computational power available at that time. With the explosion of Information Technology, there has been a dramatic increase of research in this field since the beginning of 1980s.

Document Image Analysis refers to algorithms and techniques that are applied to images of printed or handwritten documents to obtain a computer readable description from it. Research in this field dates back to the earliest experiments in computer vision, and remains prominent by the close and productive ties between the academic and industrial communities. A well-known document image analysis product is the Optical Character

Recognition (OCR) system. This software converts the scanned images of printed text (numerals, letters or symbols), into computer processable format (such as ASCII). At present, reasonably good OCR packages are available for most of the languages. But these packages are able to recognize only high quality printed text documents. It is worth noticing that despite the emergence of some specific application specific OCR systems for printed documents, progress is still to be made in the case of handwriting processing. In this background an exhaustive research is needed in the area of Handwritten Character Recognition (HCR) to deal with unconstrained handwritten documents.

According to the way in which handwriting data is generated, two different approaches are present in HCR: on-line and off-line. In the former, the data are captured during the writing process by a special pen on an electronic surface. In the latter, the data are acquired by a scanner after the writing process is over. The recognition of off-line handwriting is more complex than the on-line case due to the presence of noise in the image acquisition process and the loss of temporal information such as the writing sequence and the velocity. Off-line and on-line recognition systems are also categorized according to the applications they are devoted to. The off-line recognition is dedicated to bank check processing, mail sorting, reading of commercial forms etc., while the online recognition is mainly dedicated to pen computing industry and security domains such as signature verification and author authentication. The present study, based on pattern recognition, is specifically interested in off-line recognition of isolated Malayalam handwritten characters.

1.2 Motivation

The motivation for the spurt of activities in the field of handwritten character recognition is the increasing need for common people to interact with the computing machines in their natural mode of communication. Another important motivation is to design and make intelligent machines. The interest devoted to this study is not only due to the exciting challenges involved but also by the expected huge benefits that a HCR system, designed in the context of a commercial application, could bring.

Our visual system is capable of recognizing image patterns in an impressively fast and accurate way. We recognize various patterns and objects in the daily environment without much effort. The task of handwritten character recognition, that is simple for human, is still a problem to a machine. It depends on various constraints and parameters and still without a definite solution. Even though tremendous advances have been made for the recognition of printed characters recently, the recognition of handwritten characters pose some difficult and interesting challenges.

The recognition of numerals, a sub field of character recognition, is subject of much attention since the beginning of research in the field of HCR and has achieved almost saturation with more than 99% recognition accuracy. In the case of recognition of alphabetic characters, the problem becomes more complicated since the ambiguity is high and the number of classes to be distinguished is more.

Many promising research results are reported in the area of handwritten character recognition for languages like English, Chinese, Korean, Japanese and Arabic. In Indian languages studies are active in Devanagari and Bangla. Some promising research findings are also reported in South Indian (Dravidian) languages like Tamil, Telungu and Kannada, where as HCR research in Malayalam is still in its infancy. So extensive research works are highly essential in Malayalam handwritten character recognition.

Selection of a feature extraction method is probably the single most important factor in achieving high recognition performance. At present there is no perfect mathematical model that can describe the extreme variations present in handwriting and hence it is impossible to find characteristic features that are invariant with different writing styles. It is also noted that the existing features like Fourier descriptors, Wavelet Transformation based parameters, Zernike moments, Invariant moments etc. are not easy to implement because of their computational complexity. So in the present study an investigation is carried out for designing a computationally simple feature extraction method in order to extract parameters that are reasonably insensitive to variations caused by individual writing styles.

Another intention of this study is to identify features to represent Malayalam handwritten characters with reduced dimensions and design computationally simple algorithms to extract them. The *curse of dimensionality* phenomenon [Duda.R.O and Hart.P.E, 1973], [Jain.A.K and Chandrasekaran.B, 1982] cautions us that with a limited training set, the size of feature must be kept reasonably small to get a better recognition result. The problem with existing features (e.g., chain code histograms/directional chain codes, contour profiles, stroke based features etc.,) is that, in almost all the cases, size of the feature vector is very large. At present, the computational complexity of feature extraction technique and the large size of the feature vector both play major role in limiting the performance of the recognition systems. To develop a considerably small sized feature vector with an easy to implement feature extraction technique, gray-scale based state-space parameters and the perceptual fuzzy-zoned normalized vector distance parameters are presented. These features are computationally less complex and easy to implement on hardware.

In each and every phase of a pattern recognition system, uncertainty can arise either implicitly or explicitly. This is mainly due to the incomplete, imprecise or ambiguous input information and the ill defined and/or overlapping boundaries among the classes of regions. At preset there is no HCR system that has sufficient provision for representing and handling these uncertainties. The best solution for this problem is the fuzzy inference system. Its significance in the realm of pattern recognition is adequately justified in

- Representing linguistically phrased input features for processing
- Providing an estimate (representation) of missing information in terms of membership values.
- Representing multi-class membership of ambiguous patterns and in generating rules and inferences in linguistic form
- Extracting ill-defined image regions, primitives and properties and describing relations among them as fuzzy subsets.

This is the main motivation of the study towards the development of a HCR system with fuzzy inference based classifier architecture. The aim is to incorporate generic advantages of artificial neural networks (massive parallelism, robustness and learning) and fuzzy logic (handling of uncertainty and impreciseness) into the recognition system. As a result an integration of neural networks and fuzzy theory, commonly known as neuro-fuzzy computing has been adequately investigated in this study. Often the neuro-fuzzy models are found to perform better than either a neural network or a fuzzy system considered individually. In this perspective the present study proposes a suitable pattern recognition scheme based on adaptive neuro-fuzzy inference model for the recognition of unconstrained isolated Malayalam handwritten characters.

1.3 Outline of the Work and Main Results

The intend of the chapter 2 is to establish the necessary background for the following chapters. The former part of the chapter presents a summary of the research findings reported in the pre-processing techniques with a review of reported methods and main investigations towards the feature extraction. The later part describes various successful recognition strategies adopted in handwriting recognition followed by a special review on handwritten character recognition research in Indian languages.

Chapter 3 deals with the data acquisition methods and creation of an image database for Malayalam handwritten character set including eighty one elements consisting of vowels, consonants, *chillukal*, special symbols and numerals. The character samples are digitized by scanning at 300 dpi and stored as 128x84 pixel size in the BMP file format. The database consists of a total of 57,996 samples of eighty one character elements collected from 716 writers. The whole set of available data is split up into two categories namely training set and test set in the ratio 1:1. The developed database has the potential to be used as a resource for the handwritten character recognition research and this work is the first of its kind in Malayalam.

Chapter 4 describes various pre-processing techniques that are necessary for character images including binarization (thresholding), size normalization and thinning (morphological skeletonization) with their application outcomes. The objective of binarization is to automatically choose a threshold that separates the foreground and background information. The character images are binarized, using Otsu's iterative global thresholding technique. The second pre-processing is the size normalization, which is highly essential because of the large size variations present in the handwritten character samples in the database. To this end all the character image samples are mapped into a standard window of 66x42 pixel size. For this mapping, Affine Transformation is performed with the Bilinear Interpolation algorithm. Thinning is applied on character images for obtaining a one pixel wide skeleton of the original pattern. To perform thinning, a rotation invariant rule based thinning algorithm proposed by Ahmed.M and Ward.R is applied on the size normalized binary character image sample. The character samples obtained after these pre-processing operations are then considered for feature extraction.

A novel method for modeling Malayalam handwritten characters based on the state-space pixel distributions of their gray-scale images is described in chapter 5. In the proposed method the reconstructed state-space for each character pattern is modeled based on the information obtained from the pixel intensity distribution of both the foreground and the background of the gray-scale image. The scatter plot of the row vector of the trajectory matrix of embedding dimension d=2 named State-Space Map (SSM) is generated from the reconstructed state-space of character images. These SSMs are constructed by incorporating one, four and eight directional space variations, by considering one nearest neighbour, 4-neighbours and 8-neighbours of each pixel in the character image. The experimental results show that the SSMs constructed with eight directional space variations are well informed and hence it can be effectively used for modeling handwritten characters. The SSMs are also found to be similar for different samples of the same character and differs from character to character. In the next stage, the State-Space Point Distribution (SSPD) graph is plotted using the SSMs with eight directional space variations and consequently the SSPD parameters are extracted from it. The SSPD parameters are found to be promising and it is further used in recognition experiments.

In chapter 6, the shape analysis of Malayalam handwritten characters is carried out based on the existence ratio, aspect ratio and the centroid of each character pattern. From this study it is found that the region decomposition or zoning is the simplest and most suitable method to identify the information content regions of Malayalam characters. Two different perceptual zoning based features are specially designed for Malayalam characters. At first, the zoned vector distance (Z-VD) features are extracted from the pre-processed character images. These features are then modified by introducing fuzzy-zone boarders and normalized vector distance measures to obtain fuzzy-zoned normalized feature vector (FZ-NVD) parameters. The recognition studies based on the above mentioned features using different pattern classifiers are performed in the succeeding chapters.

In Chapter 7, the cluster analysis and classification based on *c*-Means clustering algorithm is performed based on the extracted feature vectors. The well-known non-parametric statistical algorithm named *k*-Nearest Neighbour classifier is also employed for recognition purpose. Then the classification studies are performed by means of Class-modular Multilayer Feed-forward Neural Network with Error Back-propagation learning algorithm. The recognition accuracies obtained using FZ-NVD features are found to be more promising and it is used for developing a recognition scheme based on neuro-fuzzy networks to achieve better performance.

Chapter 8 describes the design and development of a handwritten character recognition system based on the adaptive network based fuzzy inference mechanism. For this purpose, a fuzzy *if......then* rule set is framed from the FZ-NVD features. This rule set is capable of handling the immense variations in distinct writing styles of individuals. This is achieved by the fuzzification of each attribute of the feature vector using the fuzzy sets SMALL, MEDIUM and LARGE defined by *Gaussian* membership functions. For recognition we used Adaptive Neuro-Fuzzy Inference System (ANFIS), the inference part of, which is based on *Sugeno* fuzzy model. To perform parameter adjustments in learning, ANFIS needs partition of the input space according to the target pattern classes. For this purpose a pre-processing operation is performed on the entire rule set using ID3 decision tree induction procedure. The reduced set of ID3 driven optimum rules is given as input to the inference part of ANFIS. The network is then trained using a gradient based learning algorithm proposed by Jang.J.-S.R. Here we used a set of 15,752 samples of the forty four Malayalam handwritten characters for training and a disjoint sample dataset of same size for testing. The entire system is developed and implemented using MATLAB 7. The overall recognition accuracy obtained for forty four Malayalam handwritten characters using the proposed ANFIS approach based on FZ-NVD features is 82.35%. This result is as good as the highest recognition accuracy reported in various other Indian languages. Since the FZ-NVD feature vector size (Nine) is considerably small and the ill-defined inputs are efficiently treated using fuzzy inference part of ANFIS, this system can be easily implemented for high-speed recognition.

Finally, chapter 9 concludes this work and suggests a few directions for future research.

CHAPTER 2

REVIEW OF PREVIOUS WORK

2.1 Introduction

Automatic recognition of handwritten text was regarded as an important and challenging research area by scientists and engineers even before the advent of modern electronic computers. The initial research works to automate the recognition of printed characters were reported after the development of the first operational digital computer [Glauberman.M.H, 1956], [ERA, 1957], [Chow.C.K, 1957]. During the 1960s and 1970s, the recognition systems for printed text were developed to an operational level [Balm.G.J, 1970]. But the reliable system performance was achieved only when the special fonts were used in printing. With the explosion of Information Technology, there has been a dramatic increase of research in this field since the beginning of 1980s.

The important research works that has been reported so far include the development of handwritten character recognition systems for English, Japanese, Chinese, Arabic, Korean [Bortolozzi.F *et al.*, 2005] and for a few Indian Languages like Devanagari and Bangla [Pal.U and Chaudhuri.B.B, 2004]. The highest recognition accuracy reported is 92.3% for English lowercase letters (26 classes) [Dong.J.X *et al.*, 2001] and 87% for English upper/lowercase letters (52 classes) [Britto S.Jr. *et al.*, 2004] evaluated on

NIST database. In the case of characters in Indian languages the problem becomes more complicated and it is still an open problem.

This chapter presents a summary about the recent advances, new trends and important contributions in the area of handwritten character recognition. Generally, a handwritten character recognition system includes three main modules: pre-processing, feature extraction and classification. This review is also organized and presented on the basis of these tasks. Section 2.2 presents a summary of the research findings in the pre-processing techniques. Section 2.3 gives a review of methods and main investigations towards feature extraction. Section 2.4 describes various recognition strategies adopted in handwriting recognition. Section 2.5 presents a special review on handwritten character recognition research in Indian languages and finally section 2.6 concludes this review.

2.2 Review of Pre-processing Techniques for Handwritten Character Images Focused on Thresholding, Size Normalization and Thinning

Major research works carried out in pre-processing techniques, applied on character images are discussed in this section. In pre-processing, one normally performs thresholding (binarization), size normalization, morphological skeletonization (thinning), skew correction and slant normalization. In the case of isolated handwritten character images if the samples are collected in separate boxes provided for writing, the skew and slant variations present are limited to a tolerable range. In such cases, skew correction and slant normalization are not relevant. So only the foremost pre-processing operations namely thresholding, size normalization and thinning are reviewed in this instance.

Existing binarization techniques can be categorized as either global or local. Global thresholding algorithms use a single threshold, while local thresholding algorithms compute separate threshold for each pixel (or group of pixels) based on the neighborhood of the pixel. Global techniques are usually effective in character recognition because the majority of the character images have relatively constant contrast [Nakagawa.Y and Rosenfeld.A, 1979], [Mardia.K.V and Hainsworth.T.J, 1988], [Yanowitz.S.D and Bruckstein.A.M, 1989], [Trier.O.D and Taxt.T, 1995]. Local adaptive thresholding techniques are appropriate in some special instances, especially for the documents with locally varying foreground and background intensities such as engineering drawings, maps, newspapers, forms and mail envelops [Sahoo.P.K *et al.*, 1988], [Parker.J.R, 1991].

Many global thresholding techniques were proposed over the last few years [Ridler.T and Calvard.S, 1978], [Otsu.N, 1979], [Kapur.J.N *et al.*, 1985], [Kittler.J and Illingworth.J, 1986], [Pal.S.K and Rosenfeld.A, 1988], [Abutaleb.A.S, 1989], [Pal.N.R and Pal.S.K, 1989], [Kurita.T *et al.*, 1992], [Huang.L.K and Wang.M.J.J, 1995], [Brink.A.D and Pendock.N.E, 1996]. Sahoo *et al.*, [Sahoo.P.K *et al.*, 1988] compared the performance of more than twenty global thresholding algorithms including all the above cited methods.

Their comparison results show that Otsu's method [Otsu.N, 1979] gives best performance.

In Trier and Jain's study [Trier.O.D and Jain.A.K, 1995], four global thresholding techniques were compared using a performance criterion based on the capability of an OCR module developed for recognizing numbers from hydrographic images. In their study Otsu's technique [Otsu.N, 1979] performed best, followed, in order, by Kapur *et al.*'s [Kapur.J.N *et al.*, 1985] algorithm, which uses entropy of the histogram to compute the appropriate threshold value, Abutaleb's [Abutaleb.A.S, 1989] method based on two dimensional entropy and Kittler and Illingworth's [Kittler.J and Illingworth.J, 1986] minimum error technique.

Solihin and Leedham [Solihin.Y, Leedham.C.G, 1999] report a new class of histogram based global thresholding technique called *Integral Ratio*. Two integral ratio thresholding techniques, Native Integral Ratio (NIR) and Quadratic Integral Ratio (QIR) are presented in their work. This method separates the handwriting from the background by selecting an optimal threshold using histograms modified by integral ratio techniques. This algorithm guarantees following three tight requirements.

i. Retain all details of the handwriting, including faint skate-on and skate-off pen strokes at the beginning and end of strokes.
ii. Remove the background, which may contain dark colours and/or patterns.

iii. Retain handwriting produced by a wide variety of pens such as a fountain pen, ballpoint pen, fiber-tip pen, and pencil.

This technique is based on a two stage thresholding approach that necessitates each pixel of a handwritten image to be placed into one of the three areas: foreground, background and a fuzzy area (where it is hard to determine whether a pixel belongs to the foreground or to the background). Experimental results show that the QIR method is better than the well-known thresholding techniques (including Otsu's technique and Kapur's entropy thresholding technique) for certain applications, such as forensic document examination, where the above said tight requirements should be strictly observed.

Although many thresholding techniques have been proposed, none were designed specifically for handwriting images. Most of the comparative studies also consider the problem of thresholding in a more general image processing perspective. An appropriate choice of thresholding technique can be made in between Otsu's technique (suitable for the character images with relatively constant contrast) and QIR technique (suitable for forensic exercises where the document is to be examined in a more accurate way).

Size normalization of handwritten character is another important pre-processing operation addressed in this review. In a work on normalization for hand written numerals, Nagy *et al.* [Nagy.G and Tuong.N, 1970] studied a family of pattern normalization techniques based on geometric projections.

These techniques are applied to a set of 13,000 hand printed numerals obtained from sale clerks in the course of routine sales operations. The principle involved in his algorithm transforms a quadrilateral specified in terms of convex hull of each character pattern into a square. The results of normalization achieved in this method are compared and found better with respect to other published methods including size normalization through moments [Ugawa.K *et al.*, 1967].

As presented by Oliveira *et al.*, [Oliveira de J.J. Jr. and Veloso.L.R Jr., 2000], to perform size normalization, the character images can be linearly mapped into a standard plane by interpolation/extrapolation. Here the size and position of the character pattern is controlled such that the *x* and *y* dimensions of the normalized plane are filled.

In a recent research work, Battiato *et al.*, proposed a new edge adaptive algorithm for zooming digital images [Battiato.S *et al.*, 2000]. In their work, the character image is binarized and the height and width of the character pattern found out. The original gray-scale image is then cropped into the minimum rectangle enclosing the character curve. After that the character image is transformed in to a fixed sized window using Affine Transformation and Bilinear Interpolation algorithm. All the above studies indicate that size normalization of handwritten character images simplifies and enhances the performance of recognition systems.

As a result of its central role in pre-processing of data images, the design of thinning or skeletonization algorithms has been a very active research area since its inception. Thinning procedure transforms digital binary patterns into connected skeletons of unit width. Much of the handwritten character recognition works reported are based on thinned characters [Sherman.H, 1959], [Deutsch.E.S, 1968], [Alcorn.T.M and Hoggar.C.W, 1969], [Triendl.E.E, 1970], [Beun.M, 1973], [Tamura.H, 1978], [Suen.C.Y *et al.*, 1980], [Ogawa.H and Taniguchi.K, 1982], [Smith.R.W, 1987], [Lam.L and Suen.C.Y, 1988], [Abdulla.W.H *et al.*, 1988], [Lam.L and Lee.S.W, 1992]. Integrated Circuits have also been designed for this purpose [Rahier.M.C and Jespers.P.G.A, 1980].

One of the earliest experiments on thinning of character patterns was conducted by Dinnen [Dinnen.G.P, 1955]. In his work an averaging operation over a square window with a high threshold is used for thinning the input image. Kirsch *et al.*, [Kirsch.R.A *et al.*, 1957] proposes a cluster operation to obtain a thin line representation of character patterns.

There are two basic approaches for thinning a digital pattern: iterative boundary removal and distance transformation [Rosenfeld.A and Pfaltz.J, 1968], [Danielsson.P.E, 1980], [Toriwaki.J.I and Yokoi.S, 1981], [Davies.E.R and Plummer.A.P.N, 1981], [Borgefors.G, 1984], [Arcelli.C and Baja.G.S.D, 1985]. The former method is widely used for handwritten character images, since it removes boundary pixels of a connected component that neither preserve the connectivity of the pattern nor represent any of its significant geometrical features. Two basic implementation techniques have been used for this operation. One is the sequential approach, in which pixels are examined and transformed depending on the previous processed results. The other is parallel algorithm, which operates on all pixels simultaneously.

Sequential algorithms generally operate by processing only the contour pixels [Beun.M, 1973], [Pavlidis.T, 1980], [Chu.Y.K and Suen.C.Y, 1986] or by raster scanning [Yokoi.S *et al.*, 1973]. Parallel algorithms use four sub-iterations [Stefanelli.R and Rosenfeld.A, 1971], [Arcelli.C, 1979], two sub-iterations [Zhang.T.Y and Suen.C.Y, 1984], [Suzuki.T and Abe.K, 1987], [Chin.R.T *et al.*, 1987], [Chen.Y.-S and Hsu.W.-H, 1988], [Guo.Z and Hall.R.W, 1989] or one sub-cycle [Holt.C and Stewrt.A, 1987] for thinning.

In a recent work, Ahmed and Ward [Ahmed.M and Ward.R, 2002] propose a novel rule based algorithm for thinning. The important characteristic of this algorithm is that, it thins symbols to their central lines. So the shape of the symbol is preserved and it is rotation invariant in nature. This algorithm uses twenty rules for deleting the pixels that lie on the outer boundaries of the symbol (treating as a connected graph). The simultaneous application of these twenty rules to each pixel, at each iteration, result in peeling off the outer and inner boundaries of the symbol. Drawback of the above procedure is that it results in discontinuous central lines when a part of a symbol is two pixels wide in the horizontal or vertical direction. If we do not delete pixels from this two pixel wide symbol, there arise many extraneous pixels. The solution for this problem is also presented in their work. This same drawback of the Ahmed and Ward's algorithm is also pointed out by Rockett.P.I [Rockett.P.I, 2005]. He illustrated it by means of some examples where the algorithm fails on two-pixel wide lines and proposed a modified method based on graph connectivity, which corrects this shortcoming. They experimentally showed that the modified algorithm efficiently thins a set of 12,227 handwritten digits, taken from the standard CEDAR database, when the rejection of data is permitted, with an error rate of 9.32%. This results of thinning performed on English, Chinese and Arabic characters and different geometrical shapes disclose that this algorithm is very efficient in preserving the topology of symbols and letters written in any language.

In summary, the research findings reported so far on pre-processing techniques are good enough to deal with the printed or neatly written document images. The invention of effective and robust application-specific pre-processing techniques, capable of handling degraded and unconstrained handwritten document images, is to be seriously addressed.

2.3 Review of Feature Extraction Methods for Handwritten Character Recognition Focused on Gray-scale Based Features and Perceptual Zoning

The investigation of feature extraction methods has gained considerable attention since a discriminative feature set is considered the most important factor in achieving high recognition performance. There are two methods existing in feature extraction: first one is directly from the gray-scale images and other is from binary images. The overview of feature extraction methods for various representation forms namely gray-level, binary and vector or skeleton (derived from binary images) as examined by Trier *et al.*, [Trier.O.D *et al.*, 1996] is shown in Table 2.1.

Gray-scale sub image	Binary		Vector (skeleton)
	Solid character	Outer contour	
Template	Template		Template
Deformable templates			Deformable templates
Unitary Transforms	Unitary Transforms		Graph description
	Projection histograms	Contour profiles	Discrete features
Zoning	Zoning	Zoning	Zoning
Generic moments	Generic moments	Spline curve	
Zernike moments	Zernike moments	Fourier descriptors	Fourier descriptors

Table 2.1 Overview of feature extraction methods for various representation forms

As shown in table, the selection of the representation form and features are neither totally interdependent nor independent. After the former has been fixed the latter still has some options. The succeeding sections deal with the overview of feature extraction directly from gray-scale images and from the zoned binary images.

2.3.1 Feature Extraction from Gray-scale Images

As early as in 1975, Peuker and Douglas [Peuker.T.K and Douglas.D.H, 1975] had proposed a direct feature extraction method from gray-scale images. Their algorithm detects the topographic structures from the gray-scale image patterns. The gray-scale based feature extraction approaches mainly include: (1) topographical feature extraction or edge detection [Wang.L and Pavlidis.T, 1993][1], [Kahan.S *et al.*, 1987], [Rocha.J and Pavlidis.T, 1992] and [Shi.M *et al.*, 2002], (2) global feature extraction by Discrete Cosine Transform (DCT) or moment transform [Wang.X *et al*, 2001]. These methods can achieve improvements in some cases such as blur character recognition but work poorly in other cases. For example, topographical features have poor performance on images with noise or dirty background. Methods such as DCT and moment transform are very sensitive to illumination variance and character distortion.

Nackman [Nackman.L.R, 1984] discusses about the surface description of gray-scale images with the help of Critical Point Configuration Graphs (CPCG's). Three remarkable research works dealing with corner detection in contours from the gray-scale images are also reported [Zuniga.O.A and Haralick.R.M, 1983], [Rangarajan.K *et al.*, 1988], [Chen.M.H, Lee.D and Pavlidis.T, 1991]. Haralick *et al.* [Haralick.R.M *et al.*,. 1983] give precise mathematical descriptions of topographic structures called Topographic Primal Sketch (TPS) and propose a method to extract these structures from the gray-scale images.

Wang and Pavlidis [Wang.L and Pavlidis.T, 1993][2] describe a method for feature extraction directly from gray-scale character images (either machine-printed or handwritten) without the usual step of binarization. In this method, a digitized gray-scale image is treated as a noisy sample of underlying continuous surface and the desired features based on the topographic characteristics of this surface are extracted.

Bimbo *et al.*, [Bimbo.A.D *et al.*, 1994] propose methods to identify deformable templates from gray-scale images of credit card slips with poor print quality for recognition purpose. Andrews [Andrews.H.C, 1971] applies a unitary transform to gray-scale character images and obtained features that preserve the most important information about the character curves. In the transformed space, the pixels are ordered by their variance and the pixels with the highest variance are measured as features.

In a zoning based feature extraction from gray-scale images, Bokser [Bokser.M, 1992] superimposes a grid on the character image. The average gray level values computed from each of the resultant zones constitute a feature vector.

Hu [Hu.M.-K., 1962] introduced the use of moment invariants as features for pattern recognition. Hu's absolute orthogonal moment invariants (invariant to translation, scale and rotation) have been extensively used in various applications [Reiss.T.H, 1991], [Belkasim.S.O *et al.*, 1991], [Reiss.T.H, 1993]. Zernike moments have also been used by several authors for the recognition of binary solid character symbols [Khotanzad.A and Hong.Y.H, 1990][1], [Khotanzad.A and Hong.Y.H, 1990][2]. However the initial experiments suggest that all the above techniques are well suitable for gray-scale character sub-images as well.

Recently, Wang *et al.* [Wang.X, *et al.*, 2005], proposed a gray-scale based feature extraction method for character images. In their work, features are extracted directly from gray-scale character images by Gabor filters, which are specially designed from statistical information of character structure. An adaptive sigmoid function is applied to the output of Gabor filter to achieve better performance on low-quality images. In order to enhance the credibility of the extracted features, the positive and the negative real parts of the outputs from the Gabor filter are used separately to construct histogram features. This method is tested on printed Chinese characters and produced better results.

To summarize, only a few research works are reported in the area of gray-scale based feature extraction for handwritten character recognition. The major challenge in any gray-scale based feature extraction method is identifying the candidate character locations. The main shortcoming observed in all the above mentioned methods is their computational complexity. So more research is needed to evolve suitable gray-scale based features for modeling handwritten characters with less computational complexity.

2.3.2 Zoning Based Features from Binary Images

The feature extraction methods for character recognition are generally based on two types of features: statistical and structural. The statistical features derived from statistical distributions of pixels are estimated by employing the techniques like zoning, moments, projection histograms and direction histograms. Structural features are based on topological and geometrical properties of the character like strokes and their directions, end points, intersections of segments and loops [Ali.F and Pavlidis.T, 1977], [Yamamoto.K and Mori.S, 1978]. Since these two types of features are considered complementary in nature, many researches have explored the integration of structural and statistical information to highlight different character properties. This section presents a review of research works carried out in zoning based feature extrication methods on binary handwritten character images.

In fact, zoning (region decomposition) is not a feature extraction method but it is an auxiliary procedure used in combination with actual feature extractors [Tucker.N.D and Evans.F.C, 1974], [Trier.O.D *et al.*, 1996]. In this technique, the area of the input character image is split typically into number of sub-images and appropriate feature extraction method is applied to each of these regions separately. The final feature vector is formed by concatenating these measures from each of the sub-images. Bokser [Bokser.M, 1992] describes the commercial OCR system CALERA that uses zoning on binary character images. The system is designed to recognize machine printed characters of non-decorative fonts. Here zoning is used to compute the percentage of black pixels in each zone.

Kimura and Shridhar [Kimura.F and Shridhar.M, 1991] implemented zoning on character counter curves. In their work, in each zone, the counter line segments between neighboring pixels are grouped by horizontal orientation (0^0), vertical orientation (90^0) and two diagonal orientations (45^0,135^0). Then the number of line segments of each orientation is counted.

In another work, Holbaek-Hanssen *et al.*, [Holbaek-Hanssen.E *et al.*, 1986] measured the length of the chain graph in each zone. In this work the features are made size independed by dividing the graph length in each zone by the total length of the line segments in the graph. The presence or absence of junctions or endpoints in each zone is also used as additional features.

Takahashi [Takahashi.H, 1991] uses orientation histograms from vertical, horizontal and diagonal zones as shown in figure 2.1. The orientations are extracted from inner contour (if any) in addition to the outer contour when making the histogram. Further he identified high curvature points along the inner and outer contours present in each zone. For each of these points, the curvature value, the contour tangent, and its zonal position are extracted which form the final feature vector.

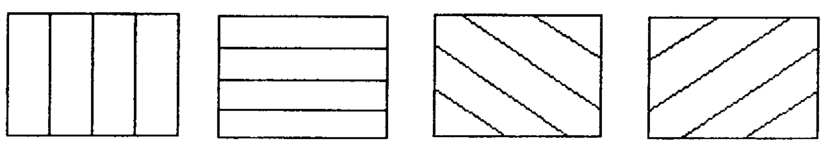


Figure 2.1 Slice-zones used by Takahashi

Suen *et al.*, [Suen.C.Y. *et al.*, 1994] presents a zoning mechanism in their experiments for hand written characters. They analyzed circumscribed character patterns by a minimum rectangle, which is partitioned into Z parts, say Z=2 (left-right), 4 and 6. Experiments on other non uniform zone models, such as Z=5 horizontal, 5 vertical and 7 as shown in figure 2.2 are also performed and achieved good recognition rates for printed English characters.

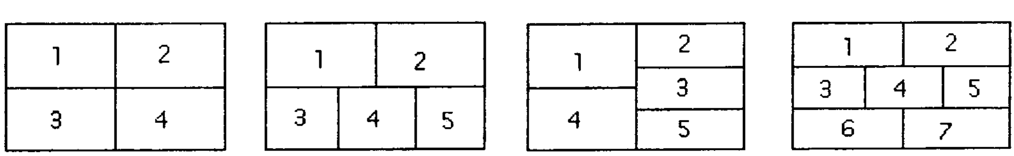


Figure 2.2 Zoning with 4, 5, & 7 vertical and horizontal zones

Both the benefits and shortcomings of the zoning method are quite obvious. Generally, it is advantageous that the characteristics of the individual parts of the image are not mixed, but processed separately. On the other hand, the fixed artificial borders that divide the character patterns to its subparts may lead to variations in the extracted features. In the same direction Cao *et al.*, [Cao.J *et al.*, 1994] pointed out that, when the character curve was close to the zone borders, small variations observed in the character curve could lead to large variations in the extracted features. They tried to compensate for this by using fuzzy borders. The points in the fuzzy-zone borders are given membership values corresponding to two or four neighboring zones such that the total membership values sum to one. Figure 2.3 shows the zoning with fuzzy borders. In the figure, pixel p_1 has a membership value 0.25 in each of the four zones 1, 2, 4 and 5. Pixel p_2 has 0.75 membership for zone 5 and 0.25 membership value for zone 6.

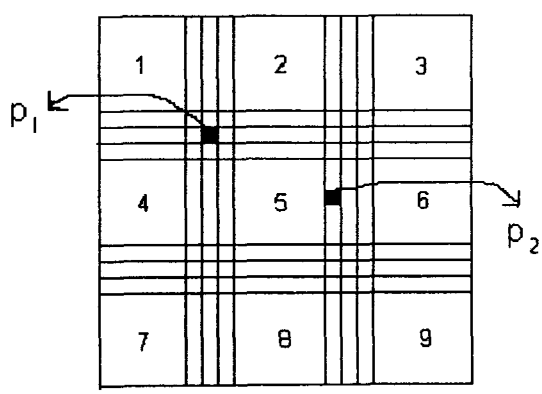


Figure 2.3 Zoning with fuzzy borders

In summary, various zoning based feature extraction techniques applied for binary character images are reviewed. Before selecting a specific feature extraction technique, one need to consider the total character recognition system, including the classifier, in which it will operate. The feature extraction based on zoning can be easily implemented for any of the character representation forms (gray-level, binary or skeleton) with comparatively less computation. It gives information about the confusion parts of the character curve but should be appropriately designed (uniform/ non-uniform/ fuzzy) by considering the shape of the script. Then the structural or statistical (or combination of both) features can be extracted from each zone for composing the feature vector for handwritten characters.

2.4 Review of State of the Art of Recognition Strategies for Handwritten Characters Focused on Neural Network and Neuro-Fuzzy Approaches

Existing character recognition strategies are generally classified based on the four approaches of pattern recognition including template matching, statistical technique, structural technique and connectionist models (neural networks). Template matching operations determine the degree of similarity between two vectors (groups of pixel, shapes, curvatures etc.,) in the feature space. Various template matching algorithms based on direct matching, deformable templates, elastic matching and relaxation matching are normally applied in the area of handwritten character and digit recognition [Xie.S.L and Suk.M, 1988], [Gader.P.D, 1991], [Dimauro.G, 1997], [Jain.A.K. *et al.* 2000].

Statistical techniques are concerned with statistical decision functions and set of optimal criteria, which determine the probability of the observed pattern belonging to a certain class. Popular handwriting recognition algorithms belonging to this domain include Bayesian classifier [Duda R.O *et al.*, 2001], *k*-Nearest Neighbour classifier (k-NN) [Mico.L and Oncina.J, 1999], [Guillevic.D and Suen.C.Y, 1995], [Zhang.B and Srihari.S.N, 2004], [Pernkopf.F, 2005], Polynomial Discriminant classifier [Schurmann J. 1996], Hidden Markov Model (HMM) [Rabiner.L.R, 1989], [Cai.J and Liu.Z.Q, 1999], [Britto Jr., *et al.*, 2001], [Kundu.A *et al.*, 2002], Fuzzy Set based reasoning [Abuhaiba.S.I and Ahmed.P, 1993], [Sasi.S and Bedi.J.S, 1994], [Gader.P.D *et al.*, 1995], [Lazzerini and Marcelloni.F., 2000], [Hanmandlu.M *et al.*, 2003], and Support Vector Machines (SVM) [Vapnik.V, 1995], [Ayat.N.E *et al.*, 2002], [Oliveira.L.S and Sabourin.R, 2004].

In structural techniques, the characters are represented as unions of structural primitives [Simon.J.C *et al.*, 1980]. It is assumed that the character primitives extracted from handwriting are quantifiable, and one can also find the relationship among them. Significant research results are also reported in handwritten character recognition using the different classes of structural methods including the grammatical and graphical methods [Heutte.L *et al.*, 1998], [Kim.H.Y and Kim J.H, 1998].

It is seen that, the above approaches are neither necessarily dependent nor disjoint from each other. Succeeding section of this review is focused on the investigations reported in the area of off-line handwriting recognition using artificial neural networks and neuro-fuzzy (a combination of adaptive network and fuzzy set based reasoning) approaches.

2.4.1. Review of the Application of Neural Networks for Handwritten Character Recognition

Artificial Neural Networks (ANN) have been extensively applied to the recognition of printed and isolated handwritten characters with widely accepted recognition results. The main advantages of neural networks lie in the ability to be trained automatically from examples, good performance with noisy data, possible parallel implementation. It serves as an efficient tool for learning large database. The most widely studied neural network model is Multi-Layer Perceptron (MLP) [Bishop.C.M, 1995]. This network architecture trained with Back-propagation algorithm is among the most popular and versatile forms of neural network classifiers and is also among the most frequently used traditional classifiers for handwriting recognition [Zhang.G.P, 2000].

Other successful network models used for handwritten character recognition include Convolutional Network (CN) [LeCun.Y *et al.*, 1998], Self Organized Maps (SOM) [Zhang *et al.*, 1999], Time Delay Neural Network (TDNN) [Lethelier *et al.*, 1995], Recurrent Neural Network [Senior.A.W and Robinson.A.J, 1998], Hopfield Network [Ling.M *et al.*, 1997] and Class-modular Neural Networks [Oh.I-S and Suen.C.Y, 2002].

In a recent work on printed Telungu (an Indian script) characters, Pujari *et al.*, [Pujari.A.K *et al.*, 2004] propose a Dynamic Neural Network (DNN) architecture for recognition. The DNN has been demonstrated to be efficient for associative memory recall and this model overcomes the inherent difficulties of memory limitations and spurious states in the Hopfield Network. In this study, the overall recognition rate reported is 91%.

Bhowmik *et al.*, [Bhowmik.T.K et al., 2004] proposed a recognition scheme for basic Bangla (an Indian script) handwritten characters using MLP classifier and stroke features. The network is trained using a variant of the Back- propagation algorithm that uses self-adaptive learning rates. Here the recognition accuracy reported for Bangla (50 class) handwritten characters is 84.33%.

The conventional Feed-Forward Neural Networks (FFNN) have been designed to classify a large number of classes with one large network structure. This inevitably poses very complex problem of determining the optimal decision boundaries for all the classes involved in a high-dimensional feature space. Oh.I-S and Suen.C.Y [Oh.I-S and Suen.C.Y, 2002] present a class-modularity concept to the FFNN classifier to overcome this limitation.

In their work the original *k*-classification problem is decomposed into *k* 2-classifcation sub problems. Here a modular architecture is adopted which consists of *k* sub-networks, each responsible for distinguishing a class from the other *k*-1 classes. Several cases have been studied using this network structure including recognition of numerals (10 classes), touching numeral pairs (100 classes), English upper case letters (26 classes), and Korean characters in postal addresses (352 classes) and showed very good test results. This experiment confirms the superiority of the class-modular network over others.

Kapp *et al.*, [Kapp.M.N *et al.*, 2003] evaluated the use of the conventional feed forward MLP architecture and class-modular neural network for the handwriting recognition. They simulated these architectures for the recognition of handwritten names of the months of the year in Brazilian Portuguese language. The recognition rate obtained for conventional MLP was 77.08% and for the class-modular neural network the recognition rate obtained was 81.75%.

The class-modularity has also been tested using neural network classifiers by many of the researchers with the experimental data [Anand.R *et al.*, 1995], [Mui.L *et al.*, 1994]. In all these works the most prominent features of class modularity, which can be effectively utilized for large class HCR problem, is highlighted in two ways

i. *Low classifier complexity*: Each of 2-classifiers has a much smaller number of parameters to be estimated by the training process. Since a 2-classifier uses the whole training set for its training, the process of parameter estimation can be accomplished in a more precise and stable manner.

ii. *Independence of classes*: Each of 2-classifiers can be designed and trained independently from the other classes. Generally, in a k classification problem, the 2-classifier module for a specific class can be viewed as a black box by the other k-1 classes and the sole purpose of the 2-classifier is to discriminate the input samples with a very high precision. One can design 2-classifier modules so that they can have their own feature set and classifier architecture.

In recent years, many researchers have combined different classifiers with neural networks and applied in handwritten character recognition applications [Hull.J.J *et al.*, 1990], [Nadal.C *et al.*, 1990], [Kimura.F and Shridhar.M, 1991], [Xu.L *et al.*, 1992], [Aksela.M and Laaksonen.J, 2007]. In Kittler's work, various classifier combination schemes have been used with neural networks and experimentally demonstrated that some of them consistently outperform a single best classifier [Kittler.J. *et al.*, 1998].

The following section reviews a special category of combined classifier approaches, where the neural networks and fuzzy systems are integrated for handwritten character recognition.

2.4.2 Review of the Application of Neuro-Fuzzy Networks for Handwritten Character Recognition

Numeric analysis approach of fuzzy system was first presented by Takagi and Sugeno [Takagi.T and Sugeno.M, 1985]. Then the studies on fuzzy neural work have been reported since around the beginning of 1990s. This is successfully applied to a wide range of areas, such as automatic control, expert systems, time series prediction, data classification and pattern recognition [Devaraj.D and Yagnanarayana.B, 2000], [Jamadagni.S and Ghadialy.Z, 2000], [Engin.M, 2004].

As a representative example, Jang.J.-S.R [Jang J.-S.R, 1993] proposed Adaptive Network-Based Fuzzy Inference System (ANFIS) architecture in 1993. This is a fuzzy inference system implemented in the framework of adaptive networks. He also presented an efficient learning procedure that can construct an input-output mapping based on fuzzy if-then-rules and stipulated input-output data pairs. As an extension, Kim.C.-H and Lee.J.-J [Kim.C.-H and Lee.J.-J, 2003] proposed a modified fuzzy modeling method based on the ANFIS with a pruning technique with appropriate measure named impact factor. Consequently, number of works has been carried out in the analysis and design of various simple and effective neuro-fuzzy architectures [Sushmita.M and Pal.S.K, 1995], [Lin.Y and Cunningham.G.A, 1995], [Nishina.T and Hagiwara.M, 1997], [Joshi.A *et al.*, 1997], [Song.F and Smith.S.M, 2001], [Iyatomi.H and Hagiwara.M., 2004].

Lin *et al.*, [Lin.C.-T and Lee.C, 1991] proposed one of the current models which decides the initial fuzzy model by Kohonen's self-organizing algorithm [Kohonen.T, 1990] and carry out parameter adjustment by Back-propagation algorithm. Kwan and Cai [Kwan.H.K and Cai.Y, 1994] proposed a fuzzy neural network for the recognition of 26 English characters and 10 Arabic numerals and obtained a good recognition result. In another work, Anasuya Devi [Anasuya Devi.H.K, 2003] proposed a fuzzy neural network based recognition system for Brahmi characters. The network was used to recognize 43 Brahmi characters and the best recognition accuracy obtained was 79.12%.

In a recent study on printed English upper case letters, Venishetti *et al.*, [Venishetti.K *et al.*, 2006] proposed a recognition system based on fuzzy min-max neural networks. The feature vectors consisting of normalized moment of inertia and eight radial coding features are used in this study and obtained 98% for single font printed character set.

To summarize, even though lot of studies are reported based on neural network in HCR, the use of combined classifiers such as neuro-fuzzy architecture is very limited. The large ambiguities present in handwritten character patterns demand an efficient network that is invariant to the distortions in the input without involving much training effort. In this context, neuro-fuzzy networks find an appropriate place in handwritten character recognition tasks and more research is needed in this direction.

2.5 Review of Handwritten Character Recognition in Indian Languages

Although many pieces of work have been done on the recognition of printed characters in Indian languages, only a few attempts have been made towards the recognition of handwritten characters. Significant research results are reported on machine printed or constrained handwritten character recognition in Devanagari script (Indian script used for writing Sanskrit, Hindi and some other languages). Sethi and Chaterjee [Sethi.I.K and Chatterjee.B, 1976] have presented a Devanagari numeral recognition system in which the presence or absence of four basic primitives, namely, horizontal and vertical line segments, right and left slant, and their interconnections are used for the recognition with a decision tree. Later the authors attempted recognition of constraint hand printed Devanagari script using the similar method [Sethi.I.K, 1977].

Sinha *et al.*, [Sinha.R.M.K and Mahabala.H, 1979] presented a syntactic pattern analysis system with an embedded picture language for hand printed Devanagari script recognition. The system stores structural descriptions for each symbol of the script in terms of primitives and their relationships. The recognition involves a search for the unknown character primitives based on the stored description and context. Sinha *et al.*, [Sinha.R.M.K, 1987], [Bansal.V and Sinha.R.M.K, 2000] later suggested knowledge based contextual post-processing systems for Devanagari script.

In a recent study, Arora.S *et al*, [Arora.S *et al.*, 2006] proposed a method for recognition of segmented handwritten Devanagari characters. In this study they used a feature vector constituted by accumulated directional gradient changes in different segments, number of intersection points, type of spine and *shirorekha* (upper line) present in the character. Multilayer perceptrons with conjugate-gradient training is used to classify these feature vectors and achieved 88.2% accuracy for 25 characters.

Comparatively large number of major works are reported for Bangla character recognition [Dutta.A.K and Chaudhuri.S, 1993], [Rahman.A.F.R. *et al.*, 2002], [Bhattacharya.U. *et al.*, 2002], [Pal.U and Chaudhuri.B.B, 2004], [Roy.K *et al.*, 2005]. Ray and Chatterjee [Ray.A.K. and Chatterjee.B, 1984] presented a nearest neighbour classifier for Bangla handwritten character recognition based on the features extracted by a string connectivity criterion.

Chaudhuri and Pal have done remarkable research work on Bangla character recognition [Pal.U and Chaudhuri.B.B, 2000]. A complete OCR system for printed Bangla script is proposed by Chaudhuri and Pal [Chaudhuri.B.B and Pal.U, 1998]. In this work, basic, modified and compound characters are separated for the convenience of classification and recognized by a structural feature based tree classifier. Here a lexicon based error correction scheme is used and obtained 95.50% word level accuracy for single font clear document. Chaudhuri.B.B *et al.*, [Chaudhuri.B.B *et al.*, 2002] also proposed a character recognition system in Oriya (an Indian language). In a related work, Pal and Sarkar [Pal.U and Sarkar.A, 2003] proposed a OCR system for printed Urdu (a popular Indian script) characters. In their work the individual characters are recognized using a combination of topological, contour and water reservoir concept based features with 97.8% average recognition accuracy.

Siromoney *et al.*, [Siromoney.G. *et al.*, 1978] attempted machine recognition of Tamil handwritten characters using an encoded character string dictionary. In another work, Chinnuswamy [Chinnuswamy.P and Krishnamoorthy.S.G, 1980] proposed a hand written Tamil character recognition system employing labeled graphs to describe structural composition of characters in terms of line-like primitives.

A two stage recognition system for Telungu handwritten characters have been proposed by [Rajasekharan.S.N.S and Deekshatulu.B.L, 1977] and a few works in this language also include [Chaudhuri.B.B *et al.*, 1991], [Sukhaswami.M.B *et al.*, 1995], [Pujari.A.K *et al.*, 2004]. Important research works are also reported for Brahmi (a script widely used all over India during third century BC) handwritten character recognition [Siromoney.G et *al.*, 1983] and [Anasuya Devi.H.K, 2003].

Nagabhushan *et al.*, [Nagabhushan.P and Radhika M.Pai, 1999] proposed a recognition scheme for Kannada characters using decision tree, by bounding the character with a rectangle that grows in horizontal and vertical directions to fit the character. In another study Nagabhushan *et al.* [Nagabhushan.P *et al.*, 2003] presented a promising fuzzy statistical recognition scheme based on seven invariant central moments for Kannada character recognition. It is also important to notice that, no favorable results have been reported for Malayalam handwritten character recognition so far.

2.6 Conclusions

In summary, the current stage in the evaluation of handwriting processing results from a combination of several elements, such as versatility of the data base used, the credibility of the pre-processing algorithms, the use of different strategies for feature selection, the performance of different classifiers and their combinations. It is clear that the maturity is reached just for isolated numerals and hence more research is needed to improve the recognition rates for alphabetic characters. It is also evident that not many research works have been carried out in Indian languages other than Devanagari and Bangla in this field. Recent advances reveal that, the neuro-fuzzy approach has a very good role in providing flexible information processing capability by devising methodologies and algorithms on a massively parallel system capable of handling infinite intra-class variations for representation and recognition of handwritten character patterns.

CHAPTER 3

HANDWRITTEN CHARACTER DATA ACQUISITION AND IMAGE DATABASE CREATION

3.1 Introduction

The availability of a dataset that contains an appropriate number of representative samples is a critical part of any pattern recognition research. Handwritten character image data acquisition and creation of database have been of great interest to computer scientists and engineers for the last few decades [Suen.C.Y *et al.*, 1980], [Garris.M.D, Wilkinson.R.A, 1992], [Hull.J.J, 1994]. The accuracy of the results of character recognition research highly depends upon the versatility (presence of moderately large and representative samples) of the database used.

The CEDAR and NIST are the two important publically available databases containing the largest collections of isolated numerals and alphabetic characters [Suen.C.Y *et al.*, 1992], [Hull.J.J., 1994], [Guyon.I *et al.*, 1997]. The use of significant sets of labeled samples of handwritten image data in languages like English, Chinese, Korean, Japanese and Arabic have been reported in the literature [Srihari.S.N. *et al.*, 1989], [Govindaraju.V *et al.*, 1994], [Lee.S.W. and Park.J.S., 1994], [Tsukumo.J. and Tanaka.H, 1998], [Xie.S.L. and Suk.M., 1988], [Kim.H.Y. and Kim.J.h., 1998], [Sekita.I. *et al.*, 1988], [Amin.A. and Al-Sadoun.H.B, 1994], [Amin.A, 1998]. However in

Indian languages, development and use of standard handwritten character image database has been done only for a few scripts.

India is a multi-lingual, multi-script country of more than one billion population using eighteen popular languages with twelve different scripts. From the literature it is found that extensive research work on developing handwritten image database has been done only for Devanagari and Bangla scripts [Sethi.I.K. 1977], [Chaudhuri.B.B. and Pal.U, 1998]. Most of the hand written character recognition studies in South Indian scripts were reported on the basis of data collected in laboratory environment [Chinnuswamy.P and Krishnamoorthy.S.G, 1980], [Chandrasekharan.M. *et al.*, 1984], [Pal.U and Chaudhuri.B.B, 2004]. At present there is no Malayalam handwritten image database available for research purpose.

Malayalam is one of the languages in Indian subcontinent and is the youngest and most dynamic one in the Dravidian language family. Malayalam is ranked as the eighth in the list of eighteen popular languages in India, used by about 30 million people. So far no remarkable research results have been reported in Malayalam handwritten character recognition. Consequently a standard database is not available in the language. This chapter presents the work carried out to create a moderately large and representative database for Malayalam handwritten characters.

3.2 Data Acquisition

Collecting a large amount of sample patterns is as important as developing recognition methods for pattern recognition applications. In the present work sufficient attention is given to collect large set of character patterns. The samples were collected with minimum writing constraints so that it reflects the real life environment, and can be used to train and evaluate recognition systems. The character patterns collected include fifteen Malayalam vowels (long and short), thirty-six consonants, five special characters named *chillukal*, fifteen symbols and ten Arabic numerals (Malayalam numerals are not in use) as shown in Table 3.1 (a-e).

Malayalam Vowels					
1	2	3	4	5	6
അ [-ah]	ആ [-aah]	ഇ [-yi]	ഈ [-yee]	ഉ [-uh]	ഊ [-ooh]
7	8	9	10	11	12
ഋ [-eru]	എ [-eh]	ഏ [-aeh]	ഐ [-ai]	ഒ [-oh]	ഓ [-ohoh]
13	14	15			
ഔ [-ow]	അം [-am]	അഃ [-aha]			

Table 3.1 (a) Malayalam Vowels (long and short)

Malayalam Consonants				
16	17	18	19	20
ക [ka]	ഖ [kha]	ഗ [ga]	ഘ [gha]	ങ [nga]
21	22	23	24	25
ച [cha]	ഛ [chha]	ജ [ja]	ഝ [jha]	ഞ [nja]
26	27	28	29	30
ട [ta]	ഠ[tta]	ഡ [da]	ഢ [dda]	ണ [nha]
31	32	33	34	35
ത [tha]	ഥ [thha]	ദ [dha]	ധ [dhha]	ന [na]
36	37	38	39	40
പ [pa]	ഫ [pha]	ബ [ba]	ഭ [bha]	മ [ma]
41	42	43	44	45
യ [ya]	ര [ra]	ല [la]	വ [va]	ശ [sha]
46	47	48	49	50
ഷ [shha]	സ [sa]	ഹ [ha]	ള [lha]	ഴ [zha]
51				
റ [rha]				

Table 3.1(b) Malayalam Consonants

Malayalam *Chillukal*				
52	53	54	55	56
ൺ [-enh]	ൻ [-en]	ർ [-er]	ൽ [-el]	ൾ [-elh]

Table 3.1 (c) Malayalam special characters (*Chillukal*)

Symbols							
57	58	59	60	61	62	63	64
ാ	ി	ീ	ു	ൂ	െ	േ	ൈ
65	66	67	68	69	70	71	
ം	ഃ	്ര	്യ	്വ	ൃ	്	

Table 3.1(d) Symbols

Numerals									
72	73	74	75	76	77	78	79	80	81
0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**

Table 3.1(e) Numerals (Arabic)

Table 3.1 (a-e) Malayalam handwritten character set with eighty one elements included in the image database

In data collection process, each writer is provided with white papers, boxes of equal size printed on it and the paper supplied was of different quality. The writers are requested to write all the eighty one character patterns listed in Table 3.1(a-e), one character per box. If any mistake is made while writing he/she can use the extra boxes provided. The purpose of data collection was not disclosed to them so that the samples reflected their natural handwriting style. The writer was permitted to use his/her own writing instrument and to write in normal mood. No other restrictions were imposed on writers.

There are several factors that influence the handwriting styles. They include age, sex, educational qualification, profession, writing instrument, writing surface and mood of the writer. In data collection, the above factors are taken into account to incorporate all possible variations of the handwriting scripts. The entire dataset is collected from seven hundred and sixteen (329 males and 387 females) individuals (writers). The writers belong to 17-52 age groups. The average age of male writers is thirty and that of female is twenty eight. An image database of handwritten Malayalam characters is created from the collected samples. Image digitization and data definition techniques used for this process are discussed in the following sections.

3.3 Image Digitization and Database Creation

The collected handwritten documents containing the character samples are digitized at 300 dpi resolution using a flatbed scanner. The character images are then separated in 128 x 84 pixel size. In order to make sure that the prepared sample data are perfect, all images of isolated characters have been manually checked and necessary corrections were made using an image editor. Sample character images (128x84 pixels) of Malayalam handwritten characters separated from the scanned document are shown in figure 3.1. These image samples are then stored in the database as gray-scale BMP file format with one byte per pixel. This format will definitely help the researchers to experiment with various pre-processing techniques including thresholding, thinning or feature extraction methods in gray-scale domain.

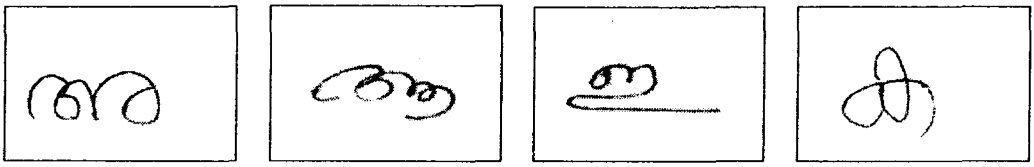


Figure 3.1 Malayalam handwritten character image samples (128x84 pixel size) separated from the scanned document

Each image data sample is labeled with a unique name in the following format. This unique labeling system will allow individual researchers to compare the performance of different algorithms on specific image sample.

LLXXXXYZZRRR

LL : Language code, e.g., *ML* for Malayalam

XXXX : Writer identification number, e.g., 0013

Y : Gender of the writer, e.g., *M*/*F*

ZZ : Character number, e.g., 16 for character “ക [ka]”

RRR : Repetition from a single writer, e.g., 015

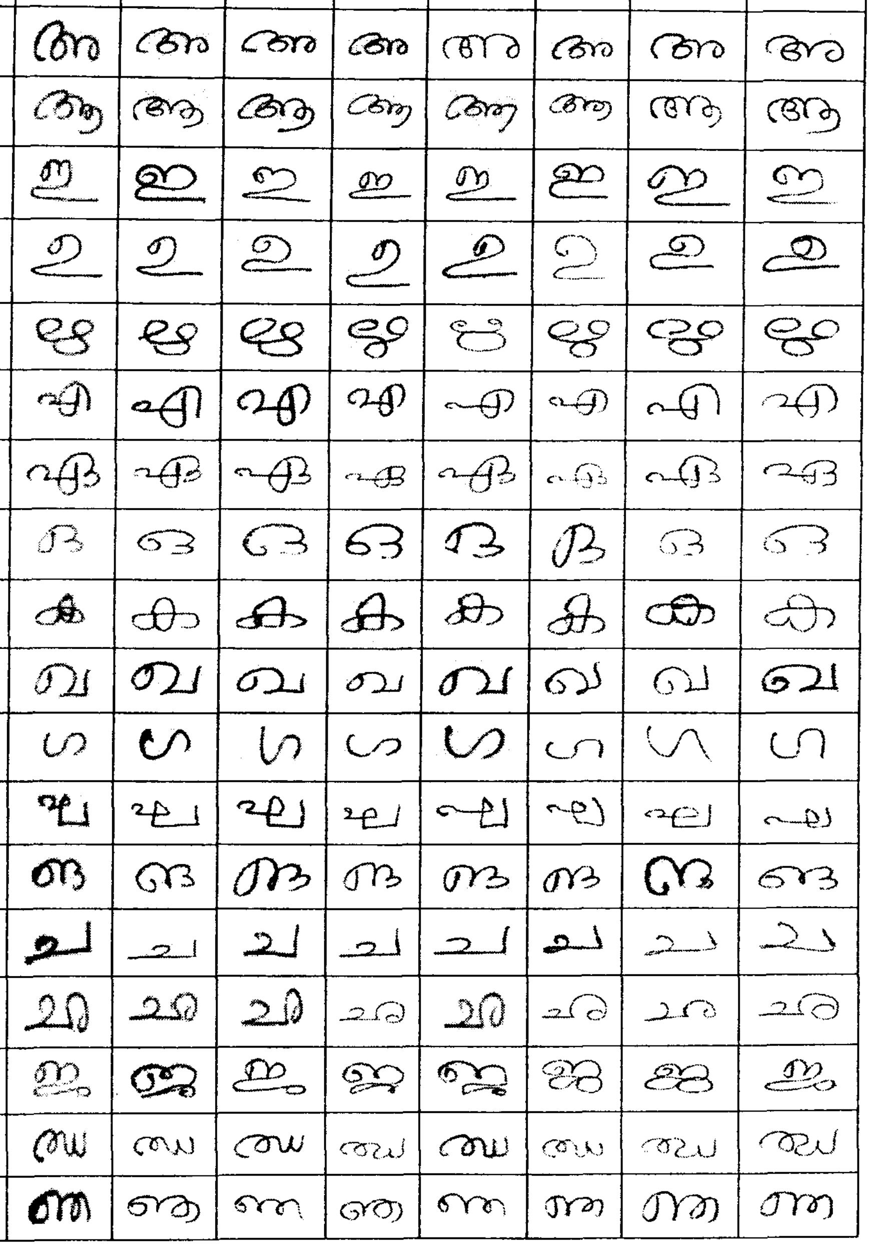
The database consists of a total of 57,996 samples of eighty one character elements collected from 716 writers. The whole set of available data is split up into two categories namely *training set* and *test set* in 1:1 ratio consisting of 28,998 samples written by 358 writers.

As discussed above the developed database contains eighty one character patterns including fifteen Malayalam vowels (both long and short), thirty-six consonants, five special characters (*chillukal*), fifteen symbols and ten numerals. Among this eighty one character patterns we selected forty four basic characters excluding the modifiers, special characters and numerals for the present recognition study. The forty four basic characters selected for the present study are given in Table 3.2. The prime reason for selecting only the basic characters for this study is that, typically, in recognition experiments the alphabetic characters are treated separately from the numerals and modifier symbols. This is because of the significant differences found in structural and statistical features of the characters with numerals or symbols. Moreover it is also noted that the recognition accuracy of numerals has achieved almost saturation and more challenges are still reported in the case of isolated characters. Hence special attention is needed in the case of isolated characters. Selected samples of the forty four basic characters taken from the database are shown in Table 3.3.

1	2	3	4	5	6	7	8	9
-ah	-aah	-yi	-uh	-eru	-eh	-aeh	-oh	ka
10	11	12	13	14	15	16	17	18
kha	ga	gha	nga	cha	chha	ja	jha	nja
19	20	21	22	23	24	25	26	27
ta	tta	da	dda	nha	tha	thha	dha	dhha
28	29	30	31	32	33	34	35	36
na	pa	pha	ha	bha	ma	ya	ra	la
37	38	39	40	41	42	43	44	
va	sha	sa	shha	ha	lha	zha	rha	

Table 3.2 Forty four basic Malayalam handwritten characters used for recognition studies

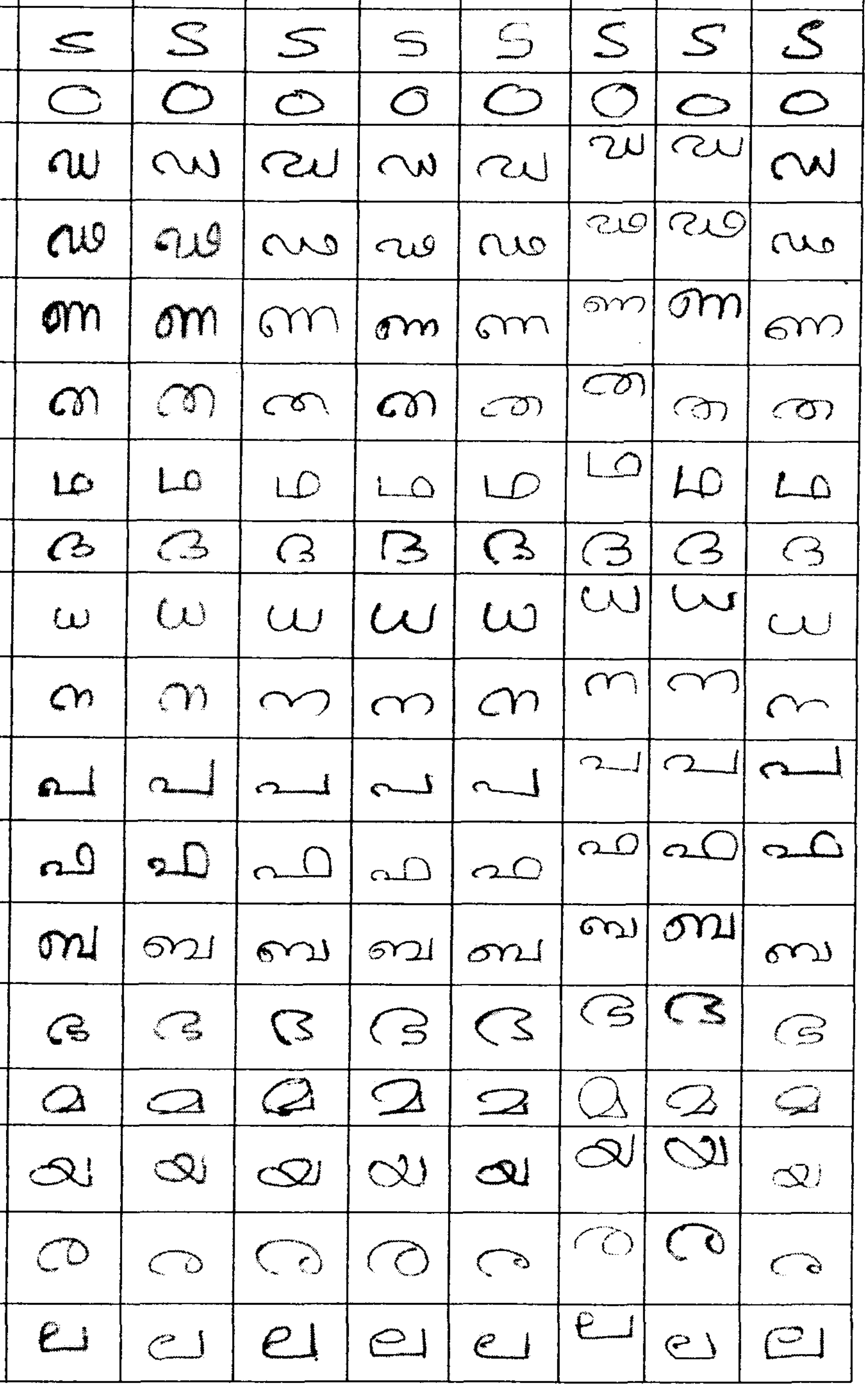
Character Number	Character	Samples							
		1	2	3	4	5	6	7	8
1	അ [-ah]								
2	ആ [-aah]								
3	ഇ [-yi]								
4	ഉ [-uh]								
5	ഋ [-eru]								
6	എ [-eh]								
7	ഏ [-aeh]								
8	ഒ [-oh]								
9	ക [ka]								
10	ഖ [kha]								
11	ഗ [ga]								
12	ഘ [gha]								
13	ങ [nga]								
14	ച [cha]								
15	ഛ [chha]								
16	ജ [ja]								
17	ഝ [jha]								
18	ഞ [nja]								



(Continued)

Table 3.3 Malayalam handwritten character image samples

Character Number	Character	Samples							
		1	2	3	4	5	6	7	8
19	ട [ta]								
20	ഠ [tta]								
21	ഡ [da]								
22	ഢ [dda]								
23	ണ [nha]								
24	ത [tha]								
25	ഥ [thha]								
26	ദ [dha]								
27	ധ [dhha]								
28	ന [na]								
29	പ [pa]								
30	ഫ [pha]								
31	ബ [ba]								
32	ഭ [bha]								
33	മ [ma]								
34	യ [ya]								
35	ര [ra]								
36	ല [la]								



(Continued)

Table 3.3 Malayalam handwritten character image samples

Character Number	Character	Samples							
		1	2	3	4	5	6	7	8
37	വ [va]								
38	ശ [sha]								
39	ഷ [shha]								
40	സ [sa]								
41	ഹ [ha]								
42	ള [lha]								
43	ഴ [zha]								
44	റ [rha]								

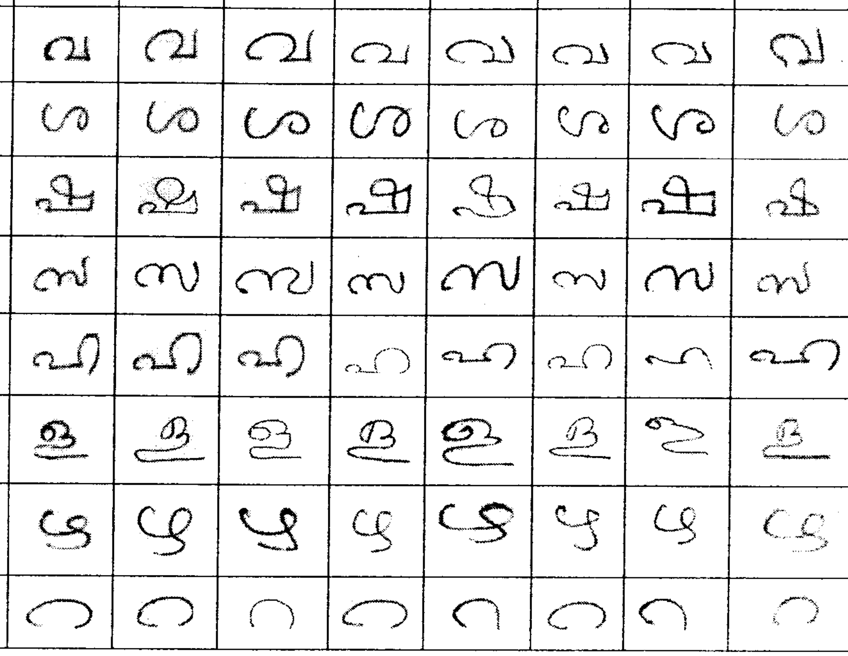


Table 3.3 Malayalam handwritten character image samples

3.4 Conclusions

An attempt is made to collect moderately large and representative samples of Malayalam handwritten characters. A well-structured and standard database of isolated Malayalam handwritten character images, which facilitates research in off-line handwritten character recognition, is developed. The key features of this database are

i. All image samples are stored in gray-scale using BMP file format providing maximum possible information.

ii. As the data were collected under minimum constraints it incorporates maximum variability.

iii. It has balanced representative data among age, sex, educational qualification and profession of the writer.

iv. The database is divided explicitly into training and testing sets to facilitate the sharing of results among researches.

The constructed database has the potential to be used as a benchmarking resource for the handwritten character recognition research and this work is the first of its kind in Malayalam. Hence the database developed in the present work will certainly help researchers to develop and validate various recognition schemes for Malayalam handwritten characters.

CHAPTER 4

PRE-PROCESSING ON HANDWRITTEN CHARACTER IMAGES

4.1 Introduction

Computer vision experiments like optical character recognition, handwritten character recognition, signature verification, forensic document examination, face recognition etc. require pre-processing (pixel level processing or low level processing) of optically scanned gray scale images acquired from a scanner or a camera arrangement. Different pre-processing algorithms are applied on the captured images to prepare them for further analysis [Ha.T.M and Bunke.H, 1997].

Since the handwritten character images are gray-scale in nature, binarization (thresholding) is usually performed prior to further processing [Mavdia.K.V and Hainsworth.T.J, 1988], [Pal.S.K and Rosenfeld.A, 1988], [Abutaleb.A.S, 1989], [Trier.O.D and Taxt.J, 1995]. The objective of binarization is to automatically choose a threshold that separates the foreground and background information. This process converts a gray-scale image in to its binary format. This process is also considered as a signal denoising technique or a method for background removal. In the present study, we choose Otsu's iterative global thresholding technique [Otsu.N, 1979] for binarizing the character image samples in the database. Otsu's technique has been compared with other binarization techniques proposed in literature by many researchers and is reported to be the best one suitable for handwritten character images [Sahoo.P.K *et al.*, 1988], [Trier.O.D and Jain.A.K, 1995].

The second pre-processing operation required is the size normalization. It is highly essential because large size variations are present among the handwritten character samples in the database. For this purpose Affine Transformation with Bilinear Interpolation algorithm is used and found suitable for character image samples in the database.

Finally, Thinning (Morphological Skeletonization) is applied to enable easier detection of pertinent features from the handwritten characters. A rotation invariant rule based thinning algorithm proposed by Ahmed and Ward [Ahmed.M and Ward.R, 2002] is used for this purpose. This algorithm is applied on the binarized character images of uniform size to obtain one pixel wide skeleton of it. The character image samples obtained after performing these pre-processing operations are then considered for feature extraction.

This chapter is organized in three sections. First section presents the character image binarization technique used for handwriting extraction. Second section deals with the geometrical spatial transformations applied on character images for size normalization. The third section describes the thinning of character images with a rotation invariant rule based algorithm.

4.2 Character Image Binarization for Handwriting Extraction

Thresholding is one of the simplest and most commonly used pre-processing techniques applied to separate the foreground information (object) of an image from its background (noise). The role of binarization in HCR is to remove the background as much as possible without modifying the handwritten information.

One obvious way to extract the objects from the background is to select a threshold T that separates these modes. Then any point (x, y) for which $f(x, y) > T$ is called an object point; otherwise, the point is called a background point. Thresholding may be viewed as an operation that involves tests against a function T of the form

$$T = T[x, y, p(x, y), f(x, y)]$$

Where $f(x, y)$ is the gray level of point (x, y) and $p(x, y)$ denotes some local property of this point, for example, the average gray level of a neighbourhood centered on (x, y). A thresholded image $g(x, y)$ is defined as

$$g(x, y) = \begin{cases} 1 & \text{if} f(x, y) > T \\ 0 & \text{if} f(x, y) \leq T \end{cases}$$

When T depends only on $f(x, y)$ (that is, only on gray level values) the threshold is called global. If T depends on $f(x, y)$ and $p(x, y)$, the threshold is called local. If in addition, T depends on the spatial co-ordinates x and y, the threshold is called dynamic or adaptive.

Before selecting an appropriate thresholding technique, a detailed analysis is performed on the intensity histograms of character images in the database. Figure 4.1(a-b) shows an example of a gray-scale handwritten character image taken from the database and its intensity histogram. From the detailed analysis of the histograms of different character samples, it is observed that the small peaks representing the foreground are usually found approximately in the intensity range 100 to140. At the same time the location of the distinct single peak of the background is often found approximately in the white intensity range 210 to 240. It is also observed that in all most all the cases the upper bound for the foreground peak is near 140 and the lower bound of the background peak end up with intensity value near 210. Presence of these widely separated foreground and background in the intensity histogram indicates that the image samples have relatively constant contrast.

To compute a good threshold value range suitable for the character image samples in the database, an experiment is conducted by thresholding the image samples using different threshold values. The outputs of the thresholding operation performed on the Malayalam handwritten character ണ[-ah] using different threshold (T) values taken in the intensity range 120-230 are shown in figure 4.2 (a-h)

Figure 4.1(a) Malayalam handwritten character image ണ [-ah]

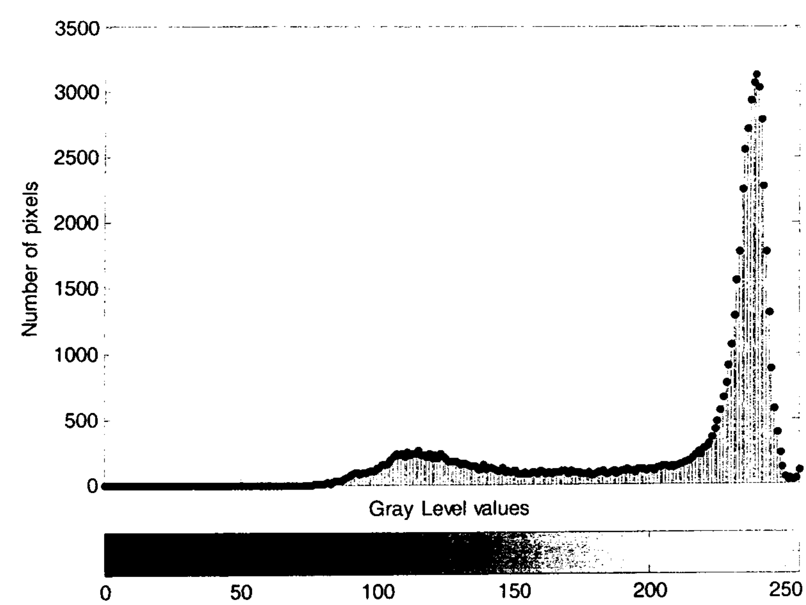
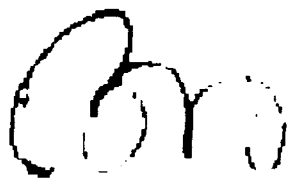


Figure 4.1 (b) Intensity histogram of image ണ [-ah]



(a) Threshold $T = 120$

(b) Threshold $T = 140$

(c) Threshold $T = 160$

(d) Threshold $T = 180$

(e) Threshold $T = 200$

(f) Threshold $T = 210$

(g) Threshold $T = 220$

(h) Threshold $T = 230$

Figure 4.2(a-h) Binarized character images using different threshold values

Since the foreground and background peaks are widely separated in the intensity histogram, there appear to be many acceptable threshold values between these peaks approximately in the range 140-210. But visually inspecting the output binary images in the figure 4.2 (a-h), it is observed that the acceptable threshold value comes in the 180-200 range. So it is clear that the information obtained from the intensity histogram is not much sufficient for accurately choosing a good threshold value.

The important characteristics of handwritten character images observed from the above analysis are stated below.

i. The foreground (handwriting) has far fewer pixels compared to the background and they tend to be more scattered.

ii. The distribution of the foreground depends heavily on the type of writing instrument.

iii. From the intensity histograms, it is observed that the small peaks of the foreground usually vary approximately in the range 120-140, while the location of the peak of the background is often distinct near the intensity range 210-240.

iv. The acceptable threshold values for handwritten images taken from the database lie approximately in the range 180-200, which is near to the background peak than the foreground peak.

In global thresholding, foreground (handwriting) extraction is accomplished by scanning the image, pixel by pixel, and labeling each pixel as background or foreground depending on whether the gray level of that pixel is greater than or less than the value of the threshold *T*. Figure 4.3 shows the traditional global thresholding approach to separate a gray-scale image into two sub images, foreground and background, using the intensity histogram of the image. In this figure, *i* denote the intensity value and *h*(*i*) denote number of pixels with intensity *i*. The foreground-background separation point is defined by a threshold value *T* and it is selected using different criteria according to the thresholding algorithms specified.

Although many thresholding techniques have been proposed none were designed specifically for handwritten images. Based on the above analysis it is observed that the majority of handwritten character image samples in the present study have relatively constant contrast for background and foreground. In this context it is decided to restrict our consideration here to global thresholding techniques, which are computationally simple and capable of giving best results on the handwritten image samples.

Numerous global thresholding techniques have been proposed in the literature to obtain threshold value *T* automatically [Ridler.T and Calvard.S, 1978], [Otsu.N, 1979], [Kapur.J.N, *et al.*, 1985], [Kittler.J and Illingworth.J, 1986], [Pal.N.R and Pal.S.K, 1989], [Kurita.T, Otsu.N and Abdelmalek.N, 1992], [Brink.A.D and Pendock.N.E, 1996]. In the present study Otsu's iterative global thresholding technique is applied on the character image samples. The following section deals with the implementation details of this algorithm.

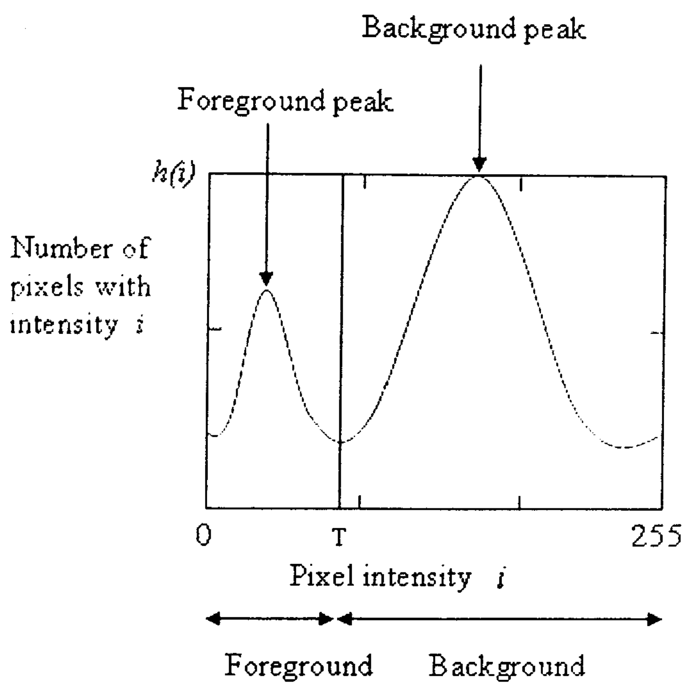


Figure 4.3 Illustration of Traditional Global thresholding

4.2.1 Otsu's Threshold Selection Technique for Handwritten Character Images

Otsu's thresholding technique has been cited as an efficient thresholding technique and has been used in numerous document-processing applications [Sahoo.P.K *et al.* 1988], [Trier O.D and Jain A.K, 1995]. The implementation details of the Otsu's algorithm are given below.

Algorithm: Otsu's threshold selection technique

Step 1: Read an input image sample.

Step 2: Take a random intensity value in the range 0-L-1 as threshold value k and divide the total pixels S in the image in two groups.

$$C_0 = [0, 1, \ldots, k\text{-}1]$$

$$C_1 = [k, k+1, \ldots, L\text{-}1)$$

where L is the possible total intensity levels.

Step 3: Compute

$$W_0 = \sum_{q=0}^{k-1} P_q(r_q)$$

$$P_q(r_q) = \frac{\text{Total pixels with intensity } q}{\text{Total pixels in the image}}$$

Step 4: Compute W_1, u_0, u_1 and u_t with the following formulas.

$$W_1 = \sum_{q=k}^{L-1} P_q(r_q)$$

$$u_0 = \sum_{q=0}^{k-1} q\, P_q(r_q) / W_0$$

$$u_1 = \sum_{q=k}^{L-1} q\, P_q(r_q) / W_1$$

$$u_t = \sum_{q=0}^{L-1} q\, P_q\,(r_q)$$

Step 5: Find the class variability λ^2 between C_0 and C_1 where

$$\lambda^2 = W_0\,(u_0 - u_t)^2 + W_1\,(u_1 - u_t)^2$$

Step 6: Repeat the above steps (for different k) until maximum class variability is obtained.

Step7: Choose Otsu's threshold value (k) with maximum class variance.

The experiential results show that the Otsu's thresholding algorithm can be effectively applied on the Malayalam handwritten image samples for threshold selection. The result of the Otsu's thresholding algorithm applied on the Malayalam handwritten character image sample is shown in figure 4.4

Threshold $T = 199$

Figure 4.4 Input gray-scale image and its binary output obtained using Otsu's technique

4.3 Geometrical Spatial Transformation for Size Normalization

Standard pre-processing methods in handwritten character recognition also include size normalization through linear geometrical transformations [Ugawa.K, Toriwaki.J and Sugino.K, 1967], [Nagy.G. and Tuong.N, 1970]. Geometrical transformations modify the spatial relationship between the pixels in an image. Suppose that an image f, defined over a (w, z) co-ordinate system, undergoes geometrical transformation to produce an image g, defined over a (x, y) co-ordinate system. This transformation (of the co-ordinates) can be explained as

$$(x, y) = T\{(w, z)\}$$

One of the most commonly used forms of spatial transformation is the Affine Transformation [Wolberg.G, 1990]. This transformation can be represented in matrix form as

$$[x \;\; y \;\; 1] = [w \;\; z \;\; 1]\,\boldsymbol{T}$$

$$= [w \;\; z \;\; 1] \begin{bmatrix} t_{11} & t_{12} & 0 \\ t_{21} & t_{22} & 0 \\ t_{31} & t_{32} & 1 \end{bmatrix}$$

This can scale, rotate, translate or sheer a set of points depending on the values chosen for the elements of $\boldsymbol{T}$.

Hence for scaling of the handwritten character images in varying sizes to a fixed window size, we apply Affine Transformation using the Affine Matrix ***T*** with the co-ordinate equation $x = s_x w$ and $y = s_y z$

$$T = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Along with the Affine Transformation we use Bilinear Interpolation algorithm, to accomplish gray level assignments of each output pixel, on, size enlargement. The basic nature of the interpolation is as follows.

Let (x', y') denote the co-ordinates of a point in the final window and let $v(x', y')$ denote the gray level assigned to it. For bilinear interpolation, the assigned gray level is given by [Gonzalez.R.C and Woods.R.E, 2002].

$$v(x', y') = a x' + b y' + c x' y' + d$$

where the four coefficients are determined from the four equations in four unknowns that can be written using the four nearest neighbours of point (x', y'). The implementation details of the size normalization algorithm are explained in the following section.

4.3.1 Size Normalization of Handwritten Characters Using Affine Transformation and Bilinear Interpolation

The high variability in the character size and shape pose serious problems in designing handwritten character recognition schemes. By observing the character samples in the image database it is clear that the collected patterns vary significantly in size. So it is decided to apply normalization for size invariance on character images prior to feature extraction. For this purpose, at first, the character images are binarized using Otsu's algorithm. The binarized character images are then cropped into a minimum rectangle, which has the same height and width of the original character pattern. It is then mapped into a standard window of 66 x 42 pixel size. For this mapping the Affine Transformation technique is applied on the character images with the Bilinear Interpolation algorithm using four nearest neighbours for interpolation. The proposed algorithm for finding the transformation matrix T is given below.

Algorithm: To find the transformation matrix T

Step 1. Initialize minimum rectangle co-ordinates ($x_{min}, x_{max}, y_{min}, y_{max}$) and final window size m x n.

Step 2: Read the character image to be size normalized from the database.

Step 3: Binarize the image sample using Otsu's threshold selection technique.

Step 4: To determine the minimum rectangle co-ordinates, $(x_{min}, x_{max}, y_{min}, y_{max})$ do the following steps.

Step 4.1: Scan the character image in row wise and find the first and last row indices x_{min} and x_{max} which has a non-white pixel.

Step 4.2: Scan the character image in column wise and find the first and last column indices y_{min}, y_{max} which has a non-white pixel.

Step 5: Compute the height (ImH) and width (ImW) of minimum rectangle

$$\text{Im}H = (x_{max} - x_{min}) + 1$$

$$\text{Im}W = (y_{max} - y_{min}) + 1$$

Step 6: Crop the original gray-scale image to the minimum rectangle such that the image fits exactly to the rectangle boundary.

Step 7: Compute the scaling factors s_x and s_y using the expressions

$$s_x = 1 + \frac{(m - \text{Im}W)}{\text{Im}W}$$

$$s_y = 1 + \frac{(n - \text{Im}H)}{\text{Im}H}$$

Step 8: Form the transformation matrix $\boldsymbol{T}$ and stop.

$$T = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

While implementing the above procedure for size normalization of the handwritten character images, in some cases, due to the rounding off error caused by the computation of scaling factors s_x and s_y, the final image size may vary in a small factor from the standard size. In order to overcome this shortcoming and to fix the final output window size as exactly 66 x 42 pixels we have copied the resultant image obtained after the transformation in to the centre of the standard image with white background. The detailed description of this white pixel padding process is given below.

Algorithm: Padding on character images after the transformation.

Step 1: Create an image (2*D* array) of size m x n initialized to zero intensity values.

Step 2: Complement the image.

Step 3: Read the character image sample obtained after transformation an find its size.

Step 4: Find the difference in row size (RowDi*ff*) and column size (ColDi*ff*) of the input image with the standard image window (66 x 42 pixel size).

Step 5: *If* row difference RowDi*ff* is EVEN *then*

Map the input image to the standard output window from the row

starting from $\left(\frac{\text{RowDiff}}{2}\right)+1$ and row end at $m-\left(\frac{\text{RowDiff}}{2}\right)$

Else

Map the input image to the standard output window from the row

starting from $\left\lceil\left(\frac{\text{RowDiff}}{2}\right)\right.$ and ends at m - $\left\lfloor\left(\frac{\text{RowDiff}}{2}\right)\right.$

($\lceil$ - is the *ceiling* operation and $\lfloor$ - is the *flooring* operation)

Step 6: *If* the column difference ColDiff is EVEN *then*

Map the input image to the standard output window from the column starting from $\left(\frac{\text{ColDiff}}{2}\right)+1$ and column ends at $n-\left(\frac{\text{ColDiff}}{2}\right)$

Else

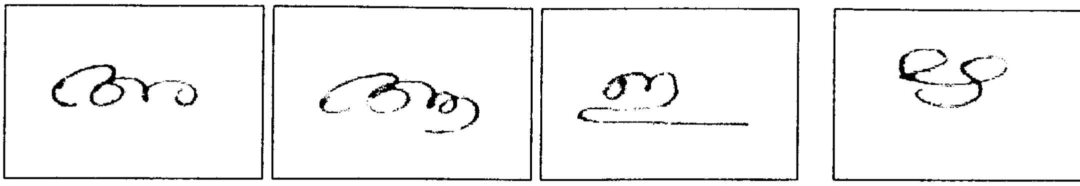
Map the input image to the standard output window from the column starting from $\left\lceil\left(\frac{\text{ColDiff}}{2}\right)\right.$ and column ends at $n-\left\lfloor\left(\frac{\text{ColDiff}}{2}\right)\right.$

Step 7: Save the final standard character image sample and stop.

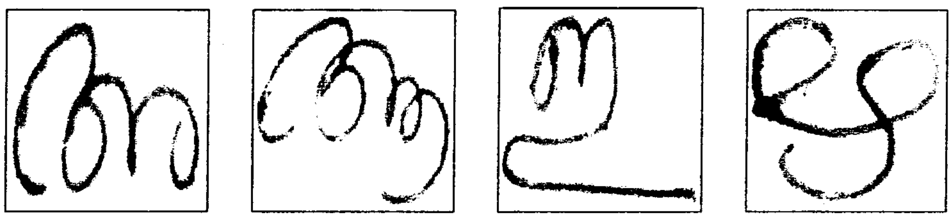
4.3.1.1 Experimental results

The character image samples of varying sizes are normalized into the standard output size using Affine Transformation and Bilinear Interpolation algorithms. The Figure 4.5 (a-d) shows the input character images of size 128x84 pixels and the size normalized images of different output size obtained after the above explained size normalization operations.

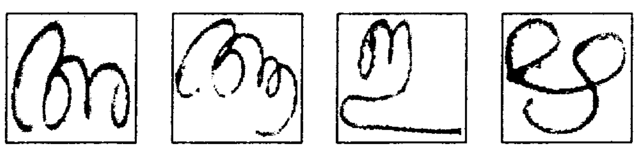
These experimental results indicate that the normalized images of 66x42 pixel size show comparatively less geometrical distortions after the transformation operation. This most favorable output window size (66x42 pixels) is fixed by the trial and error experiments.



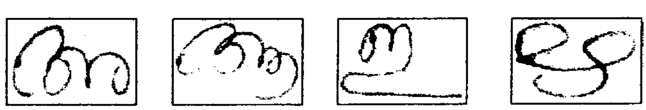
4.5 (a) Input character images of size 128 x 84 pixels



4.5 (b) Size normalized images (100 x 100 pixel size)



4.5 (c) Size normalized images (64 x 64 pixel size)



4.5 (d) Size normalized images (66 x 42 pixel size)

Figure 4.5(a-d): Input character images of size 128 x 84 pixels and and size normalized images of different output sizes (100 x 100, 64 x 64 and 66 x 42 pixels)

4.4 Thinning of Handwritten Character Images Using Rotation Invariant Rule Based Algorithm

Thinning is an image processing operation in which the binary valued image regions are reduced to lines that approximate the centre lines or skeletons of the regions. The purpose of thinning is to reduce the image components to their essential information so that further analysis and recognition are facilitated. Thinning is also identified as a morphological operation that is used to remove selected foreground pixels from binary images obtained as a result of thresholding. It is found that thin-line representations of character image patterns would be closer to human conception of patterns. The thinned handwritten character patterns permit a simpler structural analysis and more intuitive design of recognition algorithms. So in the present study, prior to feature extraction, it is decided to perform thinning on character images. The following section describes the method adopted for thinning the binarized Malayalam handwritten characters.

Many thinning algorithms, their performance, and evaluation have been studied and reviewed by Lam and Suen [Lam.L *et al.*, 1992]. These algorithms are designed to reduce the binary objects or shapes in an image to a single pixel wide pattern [Deutsch.E.S, 1968], [Govindan.V.K and Shivaprasad.A.P, 1987]. In the present study a rule based rotation invariant thinning algorithm proposed by Ahmed and Ward [Ahmed.M and Ward.R, 2002], [Rockett.P.I, 2005] is used to thin the character patterns to their central line. The reason for choosing this algorithm is that the other algorithms reported in the literature do not address the problem of rotation invariant thinning and many are specific to digits or characters written in English, Chinese, Arabic or any other specific script. Three major merits of using Ahmed and Ward's algorithm on handwritten character images are

i. This is a general system that thins character patterns to their central lines and thus the system is invariant to rotations in the original pattern.

ii. It has the desirable feature of preserving the topology of the character patterns, thus it can be applied to thin symbols, digits or characters irrespective of the scripts they are written in.

iii. This algorithm does not produce extraneous pixels or discontinuities.

This algorithm proceeds by deriving a set of twenty rules over the eight neighbours of the pixel, which is a candidate for deletion. It is iterative in nature. In each iteration it deletes every point that lies on the outer boundaries of the character pattern, as long as the width of the pattern is more than one pixel wide. This results in twenty rules, which are applied simultaneously at every iteration to each pixel. The iterations are repeated until no further changes occur. If the resultant pattern, at some point, has width (measured in any direction) equal to one pixel, then this pixel belongs to the central line of the symbol and will not be deleted. If the width at any point is two pixels, then the central line passes between these two pixels. This case is separately treated in the algorithm. The implementation steps of this algorithm are discussed below.

Consider a binary image of an isolated handwritten character. It can be considered as a graph of black and white pixels. The white pixels represent the background and are denoted by 0's. The black pixels represent the character curve and are denoted by 1's. Every pixel x has eight neighbours with a 3x3 pixel neighbourhood as shown in figure 4.6

x_6	x_7	x_8
x_5	x	x_1
x_4	x_3	x_2

Figure 4.6 Pixel x and its eight neighbours

The pixel x is then deleted if any one of the twenty conditions listed below is satisfied.

List of twenty rules used in thinning

1. $x_3 = x_4 = x_5 = x_6 = 1$ and $x_1 = x_8 = 0$

2. $x_4 = x_5 = x_6 = x_7 = 1$ and $x_1 = x_2 = 0$

3. $x_1 = x_6 = x_7 = x_8 = 1$ and $x_3 = x_4 = 0$

4. $x_5 = x_6 = x_7 = x_8 = 1$ and $x_2 = x_3 = 0$

5 $x_5 = x_6 = 1$ and $x_1 = x_2 = x_3 = x_8 = 0$

6 $x_6 = x_7 = 1$ and $x_1 = x_2 = x_3 = x_4 = 0$

7 $x_2 = x_3 = x_4 = x_5 = x_6 = x_7 = x_8 = 1$ and $x_1 = 0$

8 $x_1 = x_2 = x_4 = x_5 = x_6 = x_7 = x_8 = 1$ and $x_3 = 0$

9 $x_4 = x_5 = 1$ and $x_1 = x_2 = x_7 = x_8 = 0$

10. $x_3 = x_4 = 1$ and $x_1 = x_6 = x_7 = x_8 = 0$

11. $x_7 = x_8 = 1$ and $x_2 = x_3 = x_4 = x_5 = 0$

12. $x_1 = x_8 = 1$ and $x_3 = x_4 = x_5 = x_6 = 0$

13. $x_2 = x_3 = 1$ and $x_5 = x_6 = x_7 = x_8 = 0$

14. $x_1 = x_2 = 1$ and $x_4 = x_5 = x_6 = x_7 = 0$

15. $x_1 = x_2 = x_3 = x_4 = x_6 = x_7 = x_8 = 1$ and $x_5 = 0$

16. $x_1 = x_2 = x_3 = x_4 = x_5 = x_6 = x_8 = 1$ and $x_7 = 0$

17. $x_1 = x_2 = x_3 = x_8 = 1$ and $x_5 = x_6 = 0$

18. $x_1 = x_2 = x_7 = x_8 = 1$ and $x_4 = x_5 = 0$

19. $x_1 = x_2 = x_3 = x_4 = 1$and $x_6 = x_7 = 0$.

20. $x_2 = x_3 = x_4 = x_5 = 1$ and $x_7 = x_8 = 0$

After each iteration, the parallel application of the twenty rules to each pixel in the image can be easily shown to result in peeling off the outer and inner boundaries of character curve. Drawback of the above procedure is that, when a part of a pattern is two pixels wide in the horizontal or vertical direction, these two pixels may be deleted and results in disconnected central lines.

Figure 4.7(a) shows the results of thinning applied on Malayalam handwritten character ഞ[-ah] with thinning stages leading to disconnected central lines. If we do not delete a pixel when the horizontal and vertical width equals to two, then we get a thinned image with many extraneous pixels. Figure 4.7(b) shows the results of thinning applied on Malayalam handwritten character ഞ [-ah] with thinning stages leading to connected central lines but not one pixel wide. In order to solve this problem, for each iteration, we must first check whether the width of the pattern is two pixels or not. This can be achieved by some modification of above algorithm. The following section describes the implementation details of the modified rule based thinning algorithm and presents the results of thinning obtained for different Malayalam handwritten characters based on it.

Figure 4.7 (a) Thinning stages leading to discontented central lines

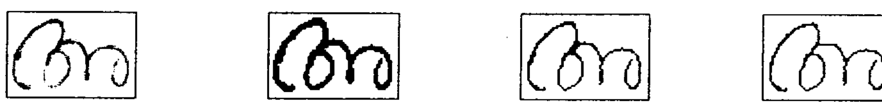


Figure 4.7 (b) Thinning stages leading to connected central lines but not one pixel width

Algorithm: Modified rule based thinning

Repeat the following steps until no changes occur from one iteration to the next.

For iteration ***i***, for every pixel ***w*** in the character image do the following:

Step 1:

If *w* belongs to two pixels wide in the vertical direction, go to step 2.

If *w* belongs to two pixels wide in horizontal direction, go to step 3.

Otherwise go to step 6.

Step 2:

If *w* belongs to

x	0	x
1	w	1
1	1	1
x	0	x

Stop calculations for the pixel *w*, otherwise, check if *w* belongs to

x	0	x
1	1	1
1	w	1
x	0	x

If yes, delete *w* and stop calculation for the pixel *w*, otherwise, go to step 4

Step 3:

If *w* belongs to

x	1	1	x
0	w	1	0
x	1	1	x

Stop calculation for the pixel *w*, otherwise, check if *w* belongs to

x	1	1	x
0	1	w	0
x	1	1	x

If yes, delete *w* and stop calculations for the pixel *w*, otherwise, go to Step 5

Step 4:

If *w* belongs to

0	0	0
0	w	0
1	1	0
1	0	0

OR

0	0	0
0	w	0
0	1	1
0	0	1

Stop calculation for the pixel *w*, otherwise, check if *w* belongs to

1	0	0
1	1	0
0	w	0
0	0	0

OR

0	0	1
0	1	1
0	w	0
0	0	0

If yes, stop calculation for the pixel *w*, otherwise go to step 6.

Step 5:

If *w* belongs to

0	0	0	0
0	w	1	0
0	0	1	1

OR

0	0	1	1
0	w	1	0
0	0	0	0

Stop calculations for this pixel, otherwise check if *w* belongs to

0	0	0	0
0	1	w	0
1	1	0	0

OR

1	1	0	0
0	1	w	0
0	0	0	0

If yes, stop calculation for the pixel *w*, otherwise go to step 6

Step 6: Apply twenty rules and stop calculation for the pixel *w*.

4.4.1 Experimental Results

Figure 4.8 shows the various stages of modified thinning procedure performed on the Malayalam handwritten character ഞ[-ah] rotated in different angles. The results obtained by applying the above thinning algorithm on various isolated Malayalam handwritten character images show that the method is effective and can be used to thin any handwritten character in Malayalam irrespective of the direction of rotation.

Figure 4.8: Results of thinning performed on Malayalam handwritten character ഞ[-ah] rotated in different angles, with the original gray-scale and binary samples (Final thinned image is shown in the last column)

Since the resultant thinned characters are formed on the central lines of the pattern, it is also capable to preserve the topology (shape) of the character curve. Figure 4.9 shows the results of the modified rule based thinning algorithm applied on the different Malayalam handwritten character images.

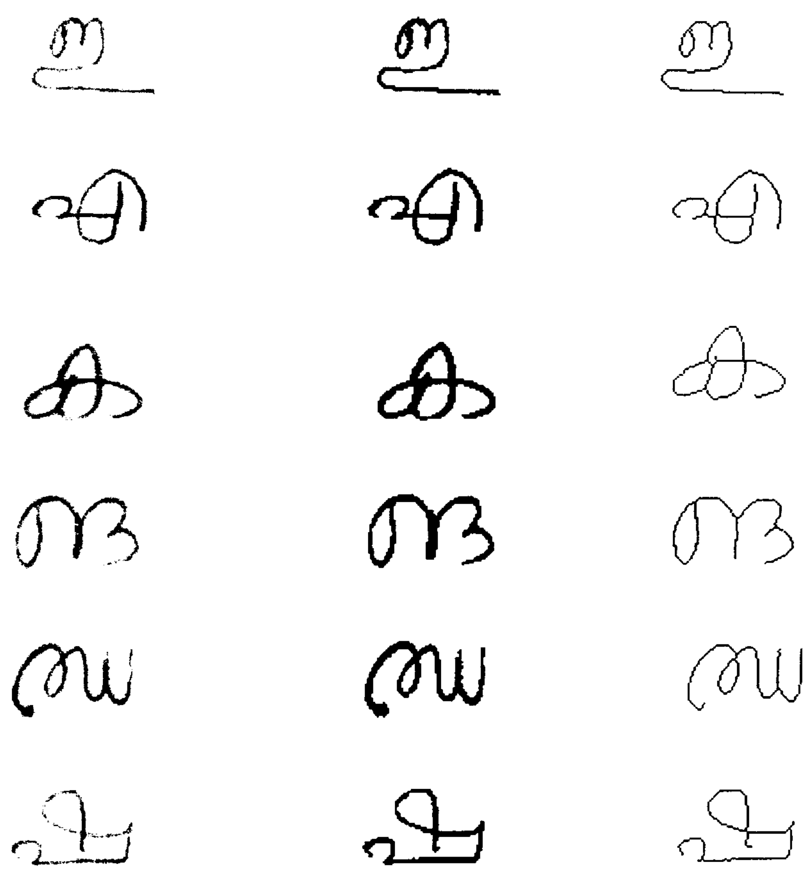


Figure 4.9 Thinned images of selected Malayalam handwritten characters with the original gray-scale and binary samples

4.5 Conclusions

Different pre-processing techniques that are capable of preparing the original gray-scale images in the database for further feature extraction studies are discussed in this chapter. At first the foreground extraction from the gray-scale character image samples is performed using Otsu's global thresholding technique. To bring about size uniformity, the character image samples are then normalized in to 66x42 pixel size using Affine Transformation and Bilinear Interpolation algorithms. Finally a modified version of rotation invariant rule based thinning algorithm proposed by Ahmed.M and Ward.R is applied on Malayalam handwritten character images to acquire efficiently thinned images for further processing.

CHAPTER 5

HANDWRITTEN CHARACTER MODELING USING STATE-SPACE PARAMETERS BASED ON GRAY-SCALE IMAGES

5.1 Introduction

This chapter presents a novel method for modeling Malayalam handwritten characters directly from gray-scale images without the usual step of binarization. For this purpose the reconstructed state-space for each character pattern is generated based on the information obtained from the pixel intensity distribution of both the foreground and the background of the gray-scale image. Then the trajectory matrix of embedding dimension d=2 is formed from this reconstructed state-space of each character image. After that the scatter plot of the row vector of the trajectory matrix named Sate-Space Map (SSM) is constructed for each character image with one, four and eight directional space variations (space delay). The State-Space Point Distribution (SSPD) parameters are extracted for each character pattern from the state-space map. The extracted parameters are found to be promising and can be effectively used for the recognition purpose.

5.2 Gray-scale Based Features for Modeling Handwritten Characters

In the present study we propose a promising approach to perform feature extraction from gray-scale images of the handwritten characters. Direct feature extraction from gray-scale image is not a new idea [Wang.L and Pavlidis.T, 1993][2]. As early as in 1975, Peuker and Douglar [Peuker.T.K and Donglar.D.H, 1975] had proposed methods for the detection of topographic structure in a gray-scale image. Figure 5.1 shows the block diagram of the direct feature extraction method from the gray-scale image. A major challenge in gray-scale image based feature extraction is to identify the candidate character locations in the image [Kahan.S *et al.*, 1987]. Here we introduced a new method of feature extraction based on the state-space representation by trying all possible locations in a gray-scale image.

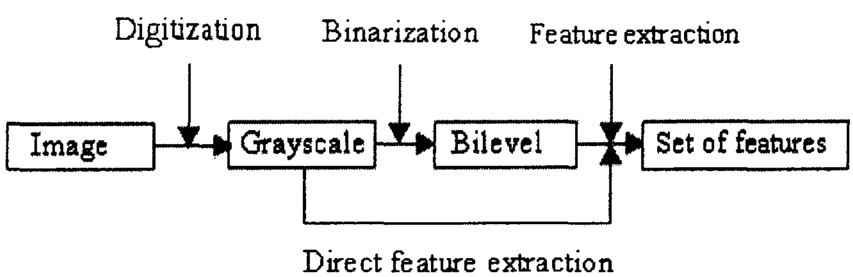


Figure 5.1 Direct feature extraction from gray-scale image

5.3 The Reconstructed State-space for Handwritten Character Images

In the case of purely deterministic systems, once it's state is fixed, then the states at all future times can be determined as well. Thus by all accounts it is significant to establish a vector space called State-Space or Phase-Space for the system such that, specifying a point in this space specifies the state of the system. This will definitely help us to study the dynamics of the system by studying the dynamics of the corresponding state-space points.

The concept of the state of a system is powerful for non-deterministic systems also. *Takens theorem* states that under certain assumptions, state-space of a dynamical system can be reconstructed through the use of time delayed (space varying) versions of the original scalar measurements [Takens.F, 1981]. This new state-space is commonly referred in the literature as a reconstructed state-space. It is also proven that the reconstructed state-spaces are topologically equivalent to the original phase-space of the dynamical system, as if all the state variables of that system would have been measured simultaneously [Ott.E, 1993], [Baker.G.L and Gollub.J, 1996]. So a reconstructed state-space can be treated as a powerful signal-processing domain, especially when the dynamical system of interest is nonlinear or even chaotic [Kantz.H and Schreiber.T, 1998], [Broomhead.D.S and King.G,1986].

A reconstructed state-space for a dynamical system can be produced from a measured state variable, $\boldsymbol{I_n}$, $n = 1, 2, 3 \ldots N$, via the method of delays by creating vectors given by

$$\boldsymbol{I_n} = [\, i_n \;\; i_{n+\tau} \; i_{n+2\tau} \; \ldots \; i_{n+(d-1)\,\tau} \,]$$

where d is the embedding dimension and τ is the chosen time or space delay value.

The row vector, $\boldsymbol{I_n}$, defines the position of a single point in the reconstructed state-space. The row vectors then can be compiled into a matrix (called a trajectory matrix) to completely define the dynamics of the system and create a reconstructed state-space.

$$I = \begin{bmatrix} i_1 & i_{1+\tau} & i_{1+2\tau} & \cdots\cdots & i_{1+(d-1)\tau} \\ i_2 & i_{2+\tau} & i_{2+2\tau} & \cdots\cdots & i_{2+(d-1)\tau} \\ i_3 & i_{3+\tau} & i_{3+2\tau} & \cdots\cdots & i_{3+(d-1)\tau} \\ \cdot & & & & \\ \cdot & & & & \\ \cdot & & & & \\ i_N & i_{N+\tau} & i_{N+2\tau} & \cdots\cdots & i_{N+(d-1)\tau} \end{bmatrix}$$

A handwritten character image, denoted by a two-dimensional function $f(x, y)$, can be treated as a dynamical system with the original scalar measurements including the pixel intensity values or amplitude of f and the spatial co-ordinates (x, y). Based on the above theory, this study investigates a method to model a reconstructed state-space for handwritten character images, through the use of space varying versions of the original scalar measurements. Here trajectory matrices I_1 with embedding dimension $d = 2$ and space delay or space variation $\tau = 1$ and I_2 with embedding dimension $d = 3$ and space delay $\tau = 1$ are constructed by considering the pixel intensity values i taken from all the spatial co-ordinate points of the image. The matrices I_1 and I_2 thus obtained are given below.

$$I_1 = \begin{bmatrix} i_1 & i_{1+\tau} \\ i_2 & i_{2+\tau} \\ i_3 & i_{3+\tau} \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ i_N & i_{N+\tau} \end{bmatrix}$$

$$I_2 = \begin{bmatrix} i_1 & i_{1+\tau} & i_{1+2\tau} \\ i_2 & i_{2+\tau} & i_{2+2\tau} \\ i_3 & i_{3+\tau} & i_{3+2\tau} \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ i_N & i_{N+\tau} & i_{N+2\tau} \end{bmatrix}$$

A visual representation of the system dynamics is evident from figure 5.2(a) and Figure 5.2(b) obtained by plotting the row vectors of the above given trajectory matrix I_1 and I_2, constructed for the Malayalam handwritten character ഞ [-ah].

The state-space reconstruction techniques are not specific to any particular production model of the underlying system with a finite dimension. Theoretically reconstructed state-spaces (in the case of an image it can be treated as space varying vectors of a two dimensional signal) are capable of capturing the full dynamics of the system including the non-linear information.

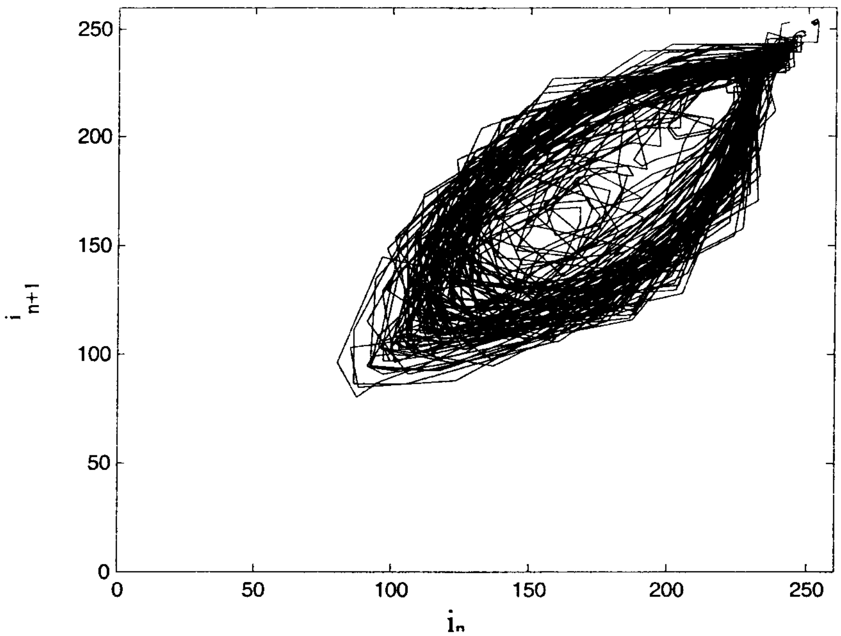


Figure 5.2 (a) Plot of the two dimensional trajectory matrix I_1 of Malayalam handwritten character ഞ [-ah]

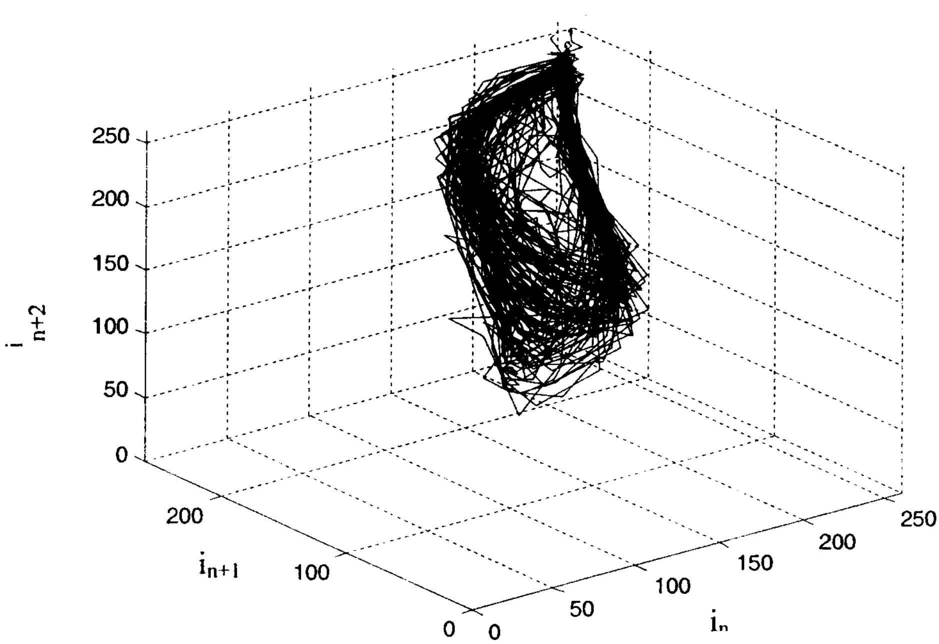


Figure 5.2 (b) Plot of the three dimensional trajectory matrix I_2 of Malayalam handwritten character ഞ [-ah]

In the present study the reconstructed state-space maps for segmented Malayalam handwritten characters are generated for one, four and eight directional space variations by considering one neighbour, 4-neighbours and 8-neighbours of each pixel with space delay $\tau=1$ as discussed in the following sections.

5.3.1 State-Space Map (SSM) of Handwritten Characters with One Directional Space Variation

In this section state-space map of handwritten characters are constructed with one directional space variation (space delay). At the outset, the total 2772 pixel intensity values obtained from the size normalized handwritten character image of size 66x42 pixels, ranging from 0 to 255 gray level values, are treated as a state vector denoted by $[i_1, i_2, \ldots.., i_{2772}]$. For each image sample, a trajectory matrix is formed by taking intensity values of each pixel $p(x, y)$ and its neighbor $p(x+1, y)$ denoted by i_n and $i^1{}_{n+1}$ respectively. In all the cases the embedding dimension'd' is two and the space delay τ is one. Now a scatter plot named State-Space Map (SSM) is generated by plotting the row vectors of the above constructed trajectory matrix (ie by plotting i_n versus $i^1{}_{n+1}$, where i^1 indicates the one directional space variation). Figure 5.3(a-f) shows the state-space maps of the selected Malayalam characters ണ [-ah], ഗ [ga], ട[ta], ഥ[thha], പ [pa] and റ[rha] constructed for one directional space variation.

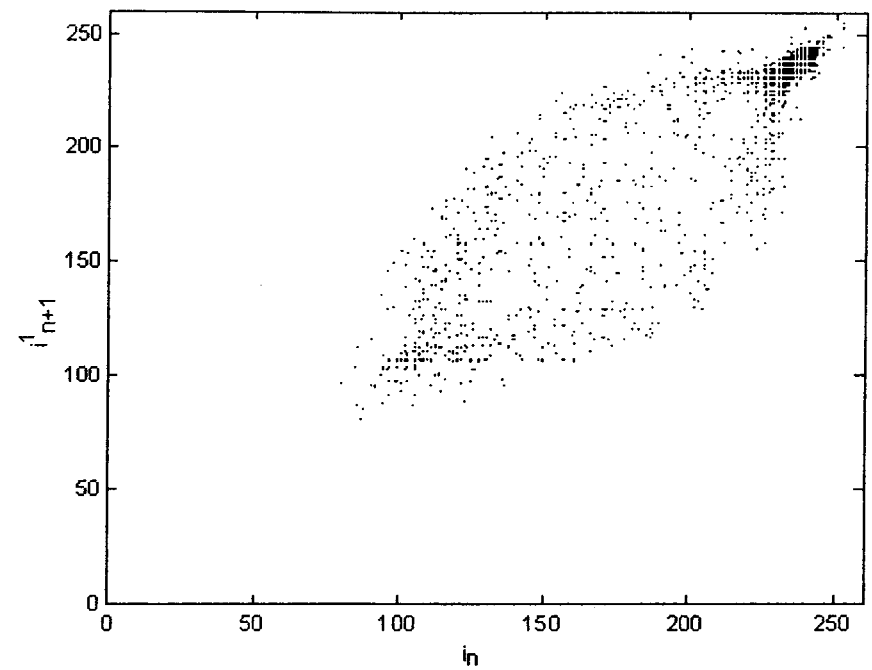


Fig. 5.3(a) SSM of Malayalam handwritten character ഞ [-ah]

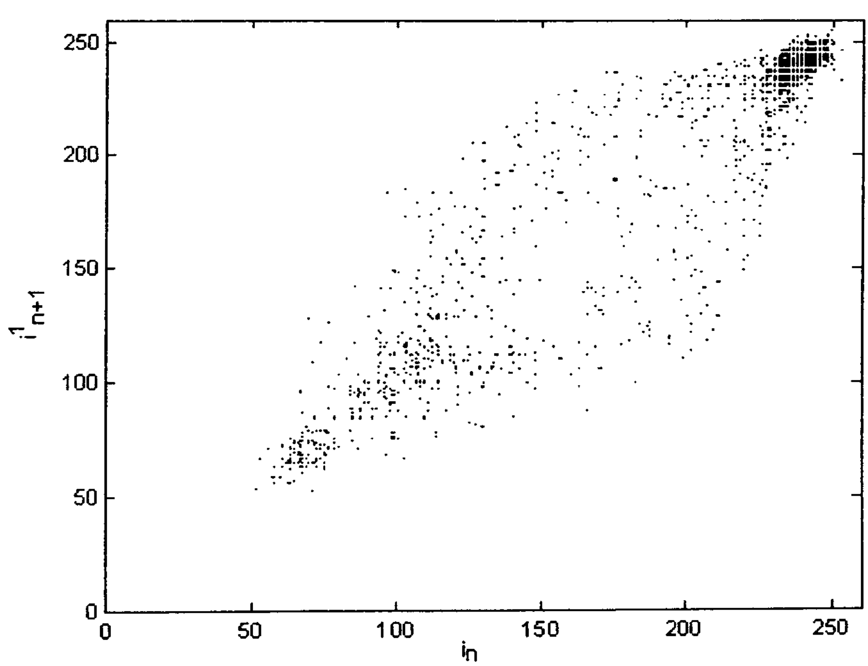


Fig. 5.3(b) SSM of Malayalam handwritten character ഗ[ga]

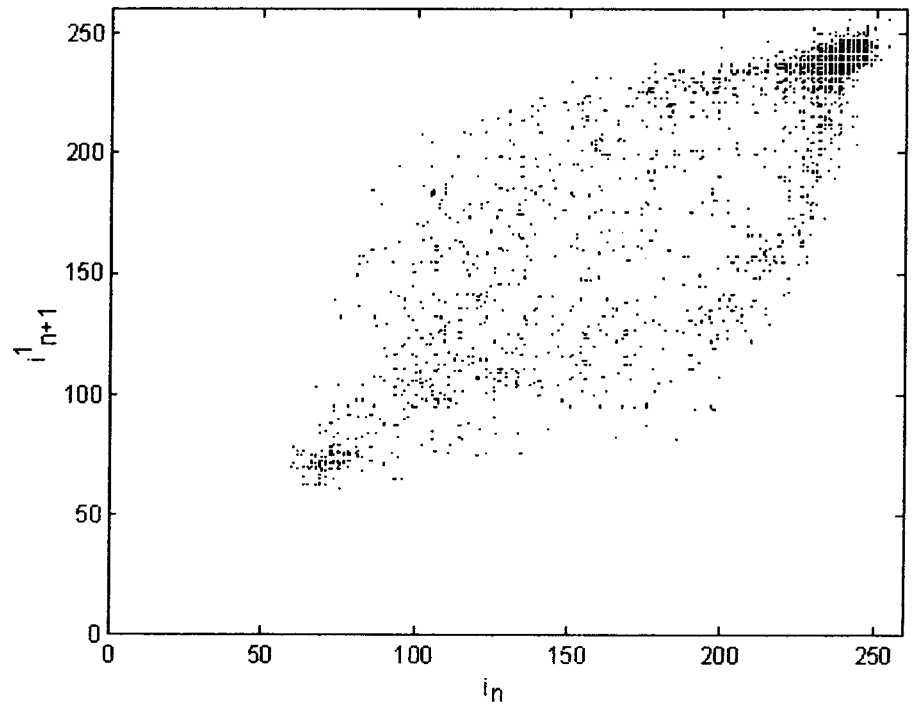


Fig. 5.3(c) SSM of Malayalam handwritten character ട[ta]

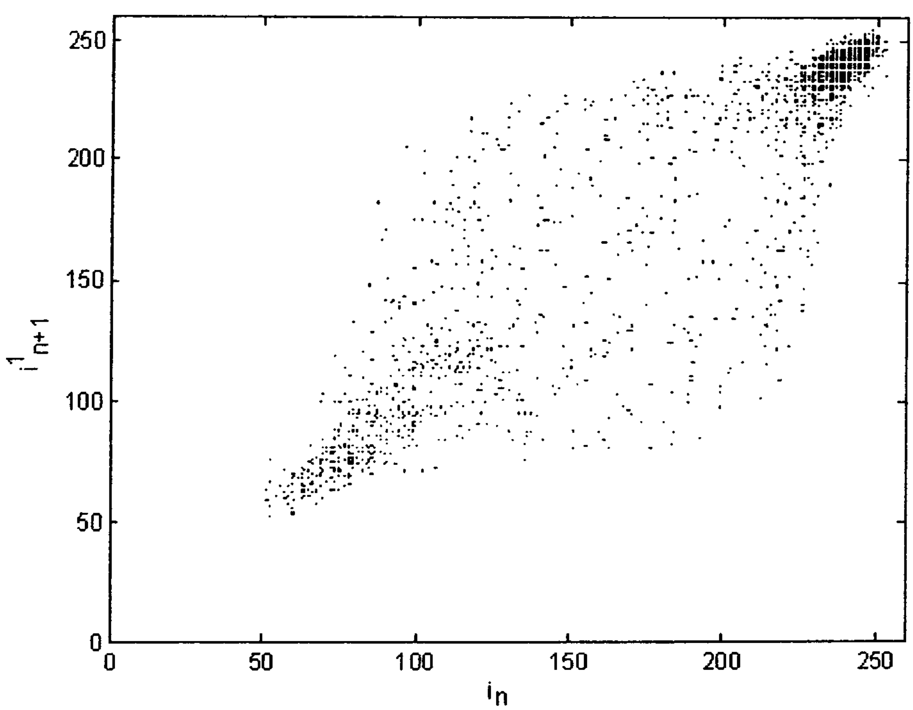


Fig. 5.3(d) SSM of Malayalam handwritten character ഥ[thha]

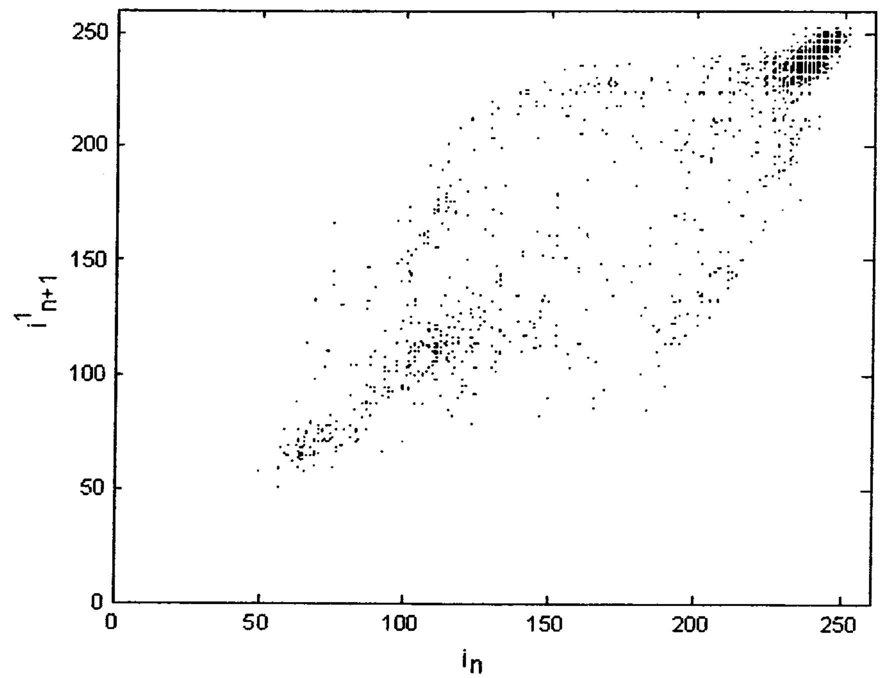


Fig. 5.3(e) SSM of Malayalam handwritten character പ [pa]

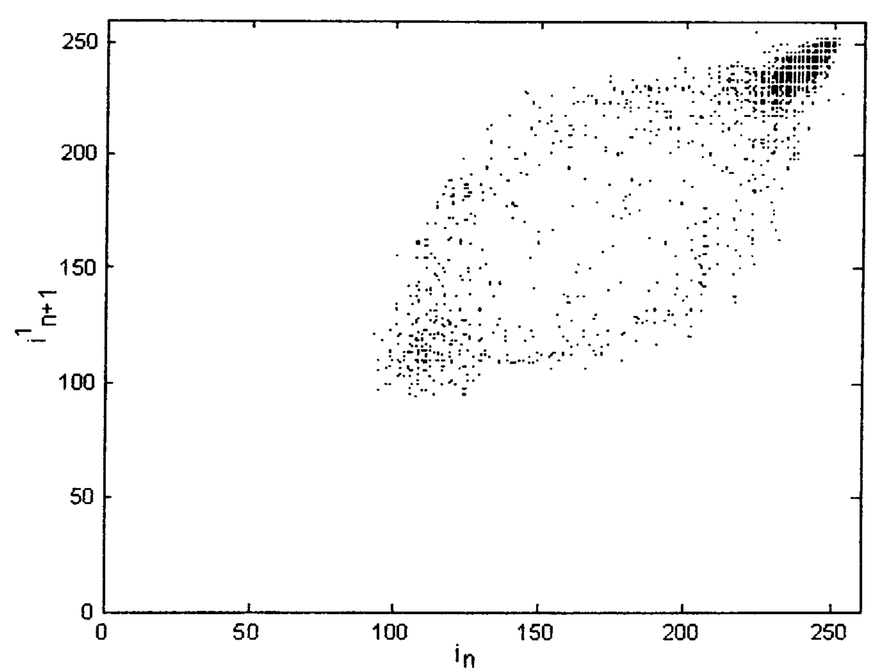


Fig. 5.3(f) SSM of Malayalam handwritten character റ[rha]

Figure 5.3(a-f): State-space maps for Malayalam handwritten characters അഃ [-ah], ഗ[ga], ട[ta], ഥ[thha], പ [pa] and റ[rha] with one directional space variation

Manually inspecting the SSMs of the forty four Malayalam characters, it is observed that more than fifty percent of the characters are distinguishable from each other. Here only one directional space variation is incorporated to construct state-space maps by considering the information regarding only one neighbour of each pixel in the image. In the case of gray-scale images, additional information of its pixel intensity distributions can be achieved by incorporating higher directional space variations. So it is good enough to construct SSMs by incorporating 4-directional and subsequently 8-directional space variations by considering 4-neighbours and 8-neighbours of each pixel in the character image. Following sections describe the incorporation of four and eight directional space variation by considering information about four and eight neighbours of each pixel in the gray-scale images for the construction of SSMs for the handwritten characters.

5.3.2 State-Space Map (SSM) of Handwritten Characters with Four Directional Space Variations

This section deals with the construction of state-space maps for size normalized (66x42 pixel size) handwritten character images with four directional space variations. Here for each pixel $p(x, y)$ in the image the set of its two horizontal and two vertical neighbours, $N4p$, whose co-ordinates are given by $(x-1, y)$, $(x+1, y)$ and $(x, y-1)$, $(x, y+1)$ respectively, are taken into account. Since each of these neighboring pixels are of unit distance from $p(x, y)$, the space variation or space delay τ is one. The intensity value of pixel $p(x, y)$ is taken as i_n and that of its four neighbours are taken as i^1_{n+1}, i^2_{n+1}, i^3_{n+1}, i^4_{n+1}. Here i^1, i^2, i^3 and i^4 indicate four directional space variations. Finally the state-space map with four directional space variation is generated by plotting i_n versus $[i^1_{n+1}, i^2_{n+1}, i^3_{n+1}, i^4_{n+1}]$ i.e., plotting i_n versus i^k_{n+1} where k = 1,2,3,4. This plot is a scatter plot of the trajectory of the system with embedding dimension d = 2 and the space delay τ =1, varies in four directions. Figure 5.4(a-f) shows the state-space maps of the selected Malayalam handwritten characters ഞ [-ah], ഗ [ga], ട[ta], ഥ[thha], പ [pa] and റ[rha] constructed for four directional space variations. From these figures it is evident that the state-space map for different characters are dissimilar in nature. It is also noted that the above obtained SSMs with four directional space variations are more informative than the state-space map with one directional space variation that are obtained in section 5.3.1. Manually inspecting the SSMs with four directional space variations of the forty four Malayalam characters it is observed that more than two third of the characters are distinguishable from each other. The modified version of the state-space map with eight directional space variations by comprising the information about eight neighbours of each pixel in the image is constructed as follows.

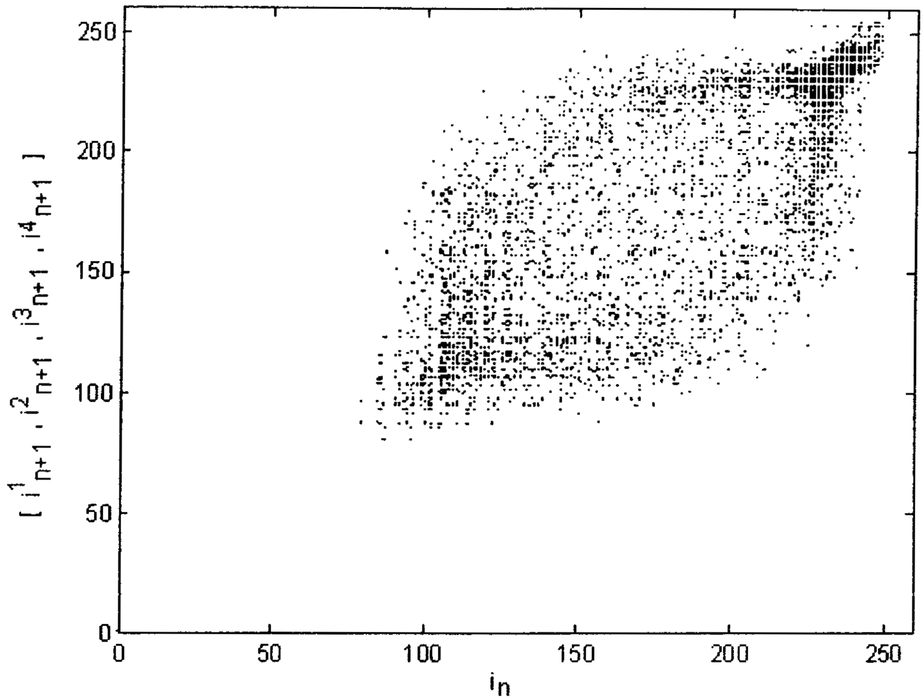


Fig. 5.4(a) SSM of Malayalam handwritten character അ [-ah]

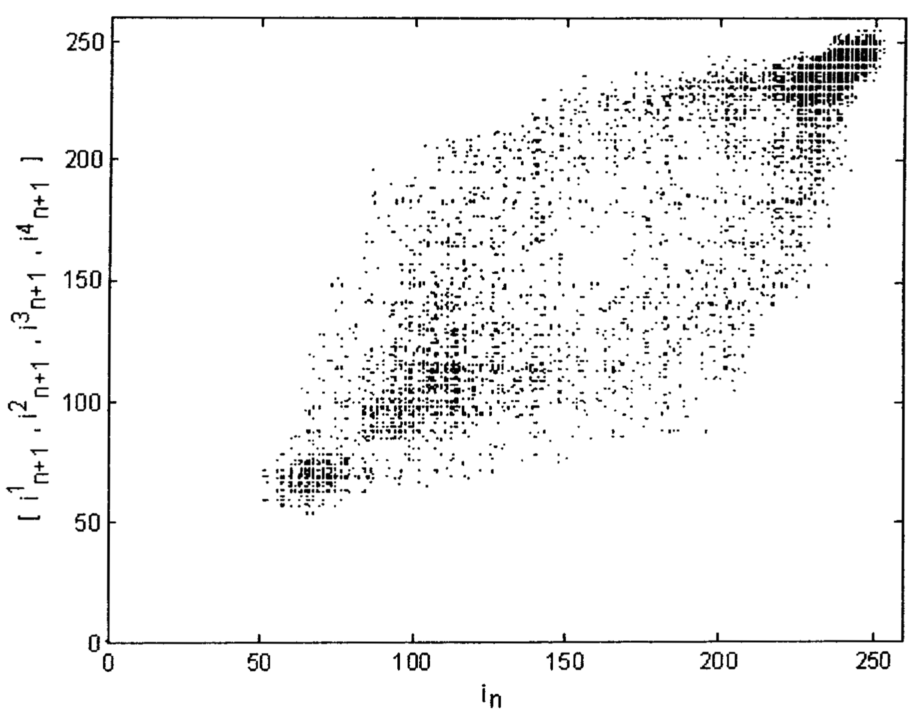


Fig. 5.4(b) SSM of Malayalam handwritten character ഗ[ga]

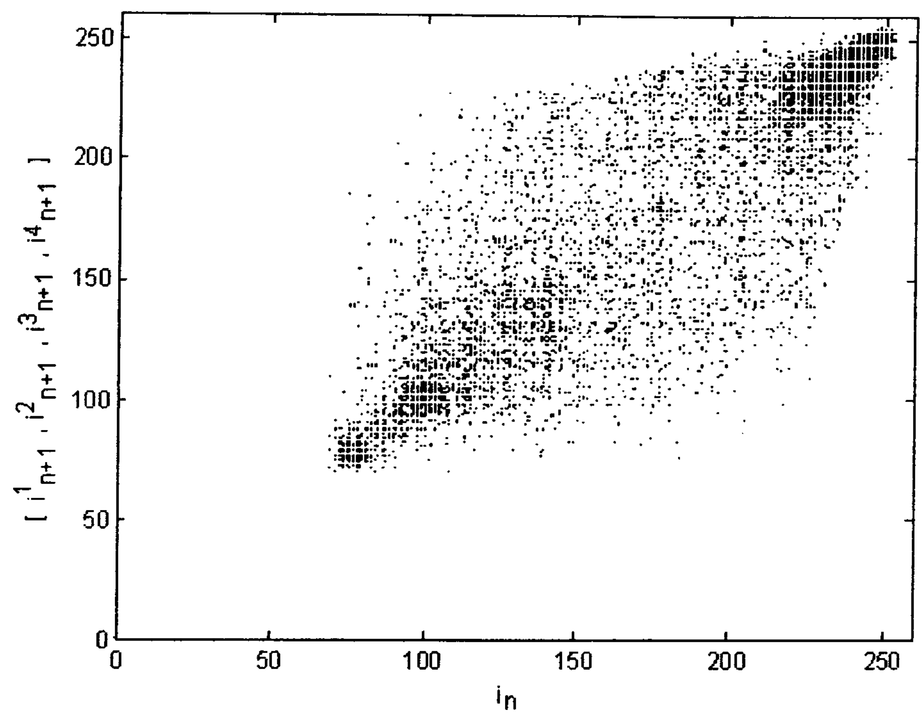


Fig. 5.4(c) SSM of Malayalam handwritten character ട[ta]

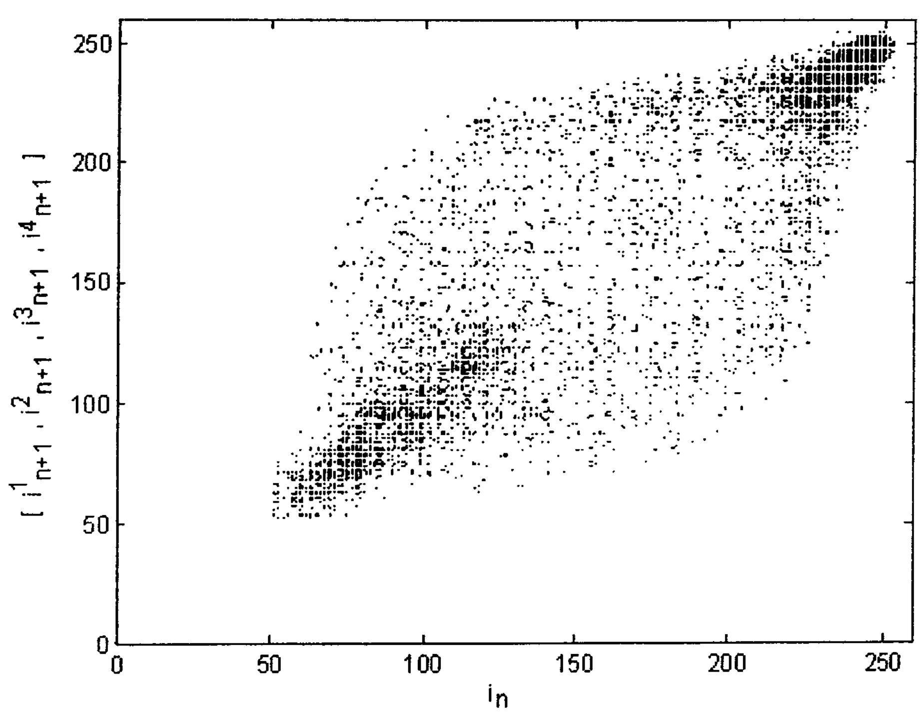


Fig. 5.4(d) SSM of Malayalam handwritten character ഥ[thha]

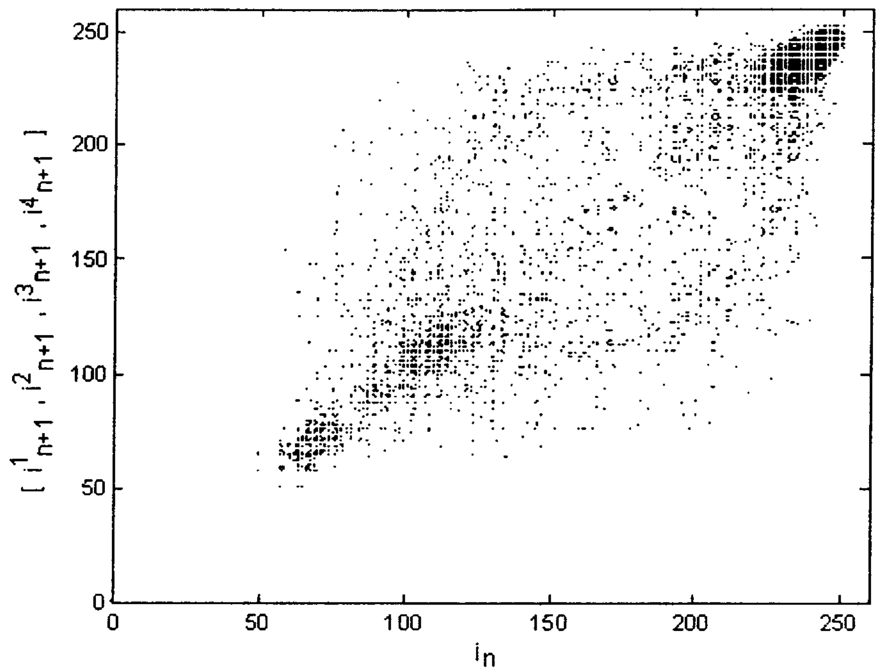


Fig. 5.4(e) SSM of Malayalam handwritten character പ [pa]

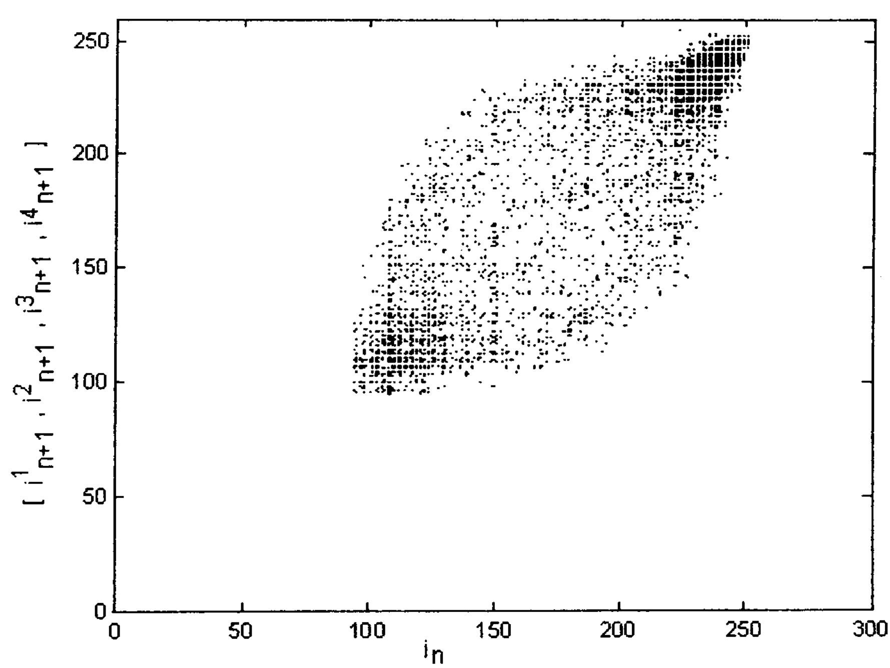


Fig. 5.3(f) SSM of Malayalam handwritten character റ[rha]

Figure 5.4(a-f): State-space maps for Malayalam handwritten characters ങ്ങ [-ah], ഗ [ga], ട[ta], ഥ[thha], പ [pa] and റ[rha] with four directional space variations

5.3.3 State-Space Map (SSM) of Handwritten Characters with Eight Directional Space Variations

This section describes the proposed method to generate the state-space map of the handwritten character images with eight directional space variations by considering the eight neighbours, $N8p$, of each pixel $p(x, y)$ in the character image. In addition to four horizontal and vertical neighbours, $N4p$, mentioned in section 5.3.2, the four diagonal neighbours, $N4D$, of $p(x, y)$ with the co-ordinates $(x\text{-}1, y\text{-}1)$, $(x\text{-}1, y+1)$, $(x+1, y\text{-}1)$ and $(x+1, y+1)$ are also considered here for constructing state-space maps. Since each of these neighbouring pixels are of unit distance from $p(x, y)$, the space variation or space delay τ is one. The intensity value of pixel $p(x, y)$ is taken as i_n and that of its eight neighbours are taken as i^k_{n+1} where k = 1, 2, .., 8. Finally the state- space map with eight directional space variation is generated by plotting i_n versus i^k_{n+1} where k = 1, 2, .., 8. This plot is a scatter plot of the trajectory of the system with embedding dimension d = 2 and the space delay τ =1, varies in eight directions. Figure 5.5 (a-f) shows the state-space maps of the selected Malayalam characters ഞ[-ah], ഗ[ga], ട[ta], ഥ[thha], പ[pa] and റ[rha] constructed for eight directional space variations. From this figure it is evident that the SSM for different characters are dissimilar in nature.

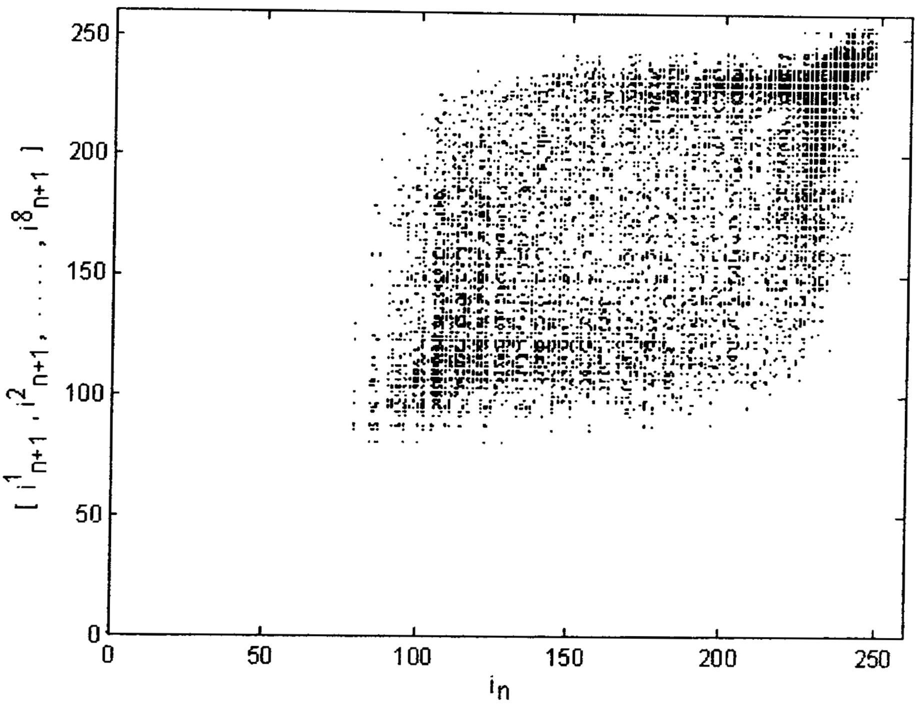


Fig. 5.5(a) SSM of Malayalam handwritten character ഞ [-ah]

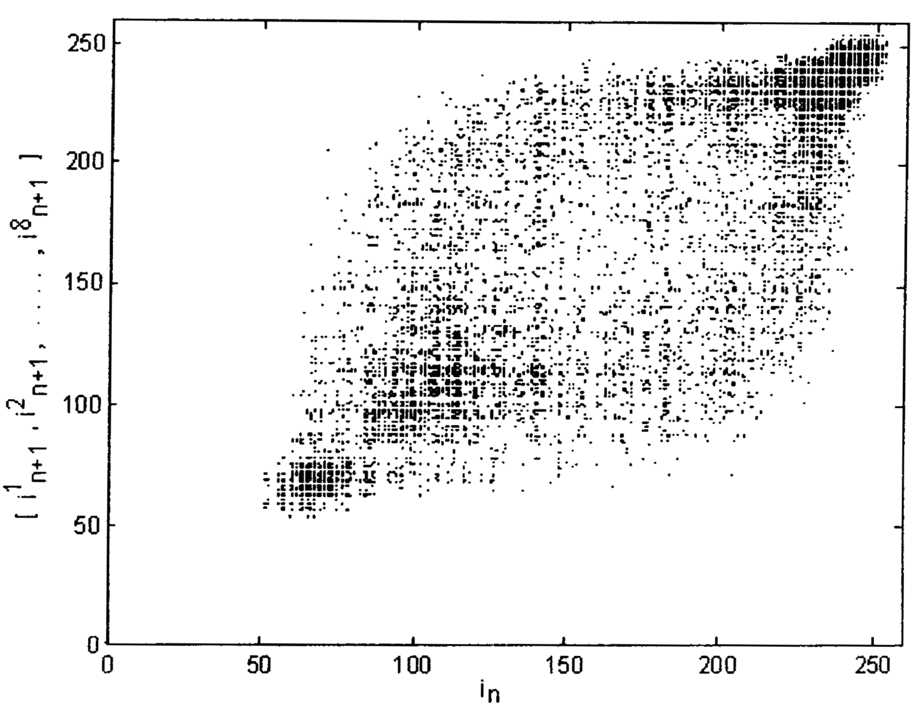


Fig. 5.5(b) SSM of Malayalam handwritten character ഗ [ga]

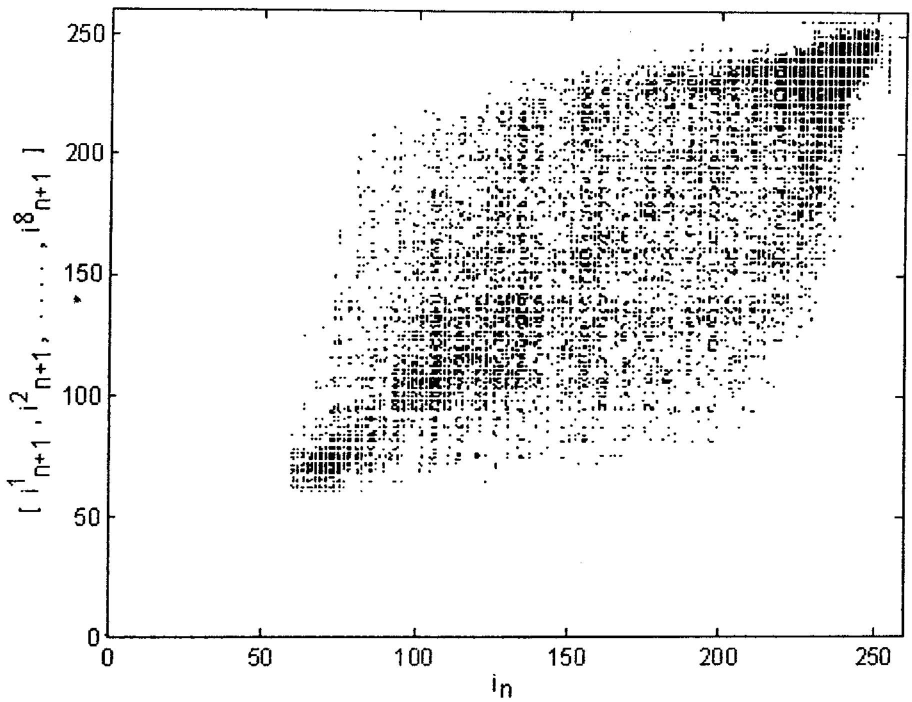


Fig. 5.5(c) SSM of Malayalam handwritten character ട[ta]

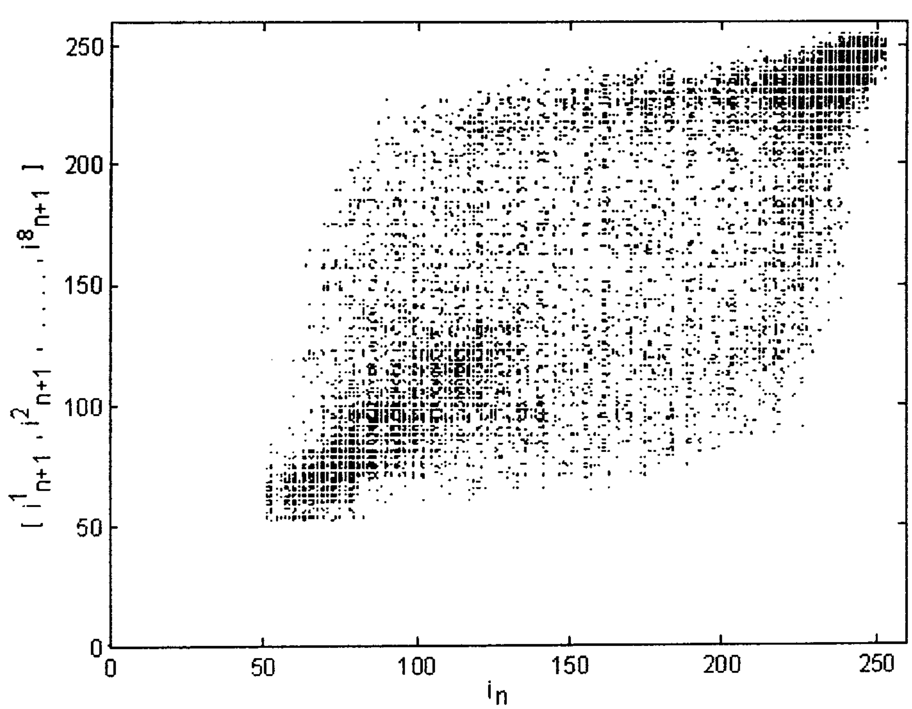


Fig. 5.5(d) SSM of Malayalam handwritten character ഥ[thha]

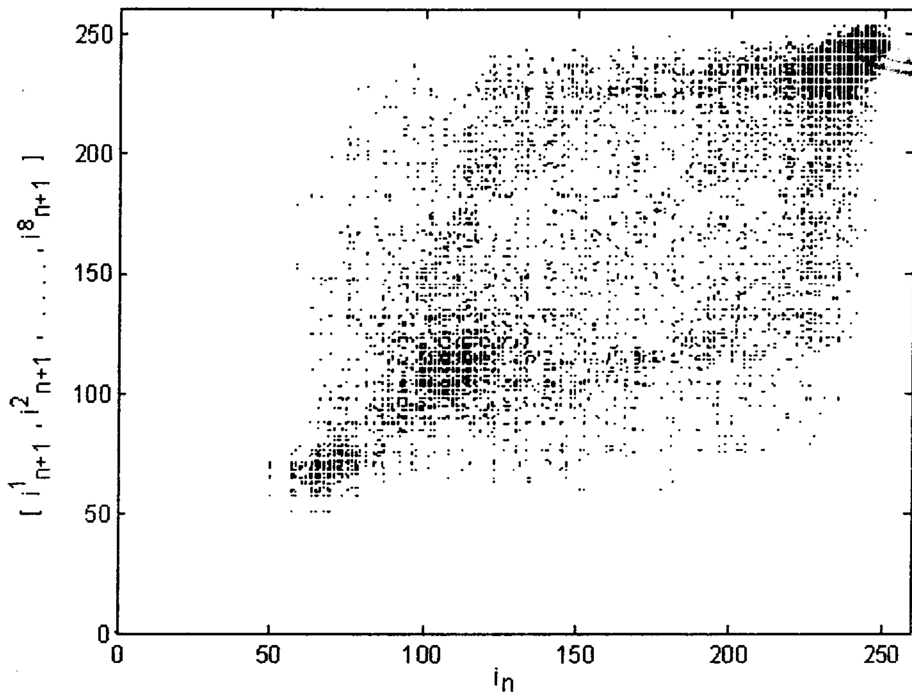


Fig. 5.5(e) SSM of Malayalam handwritten character പ [pa]

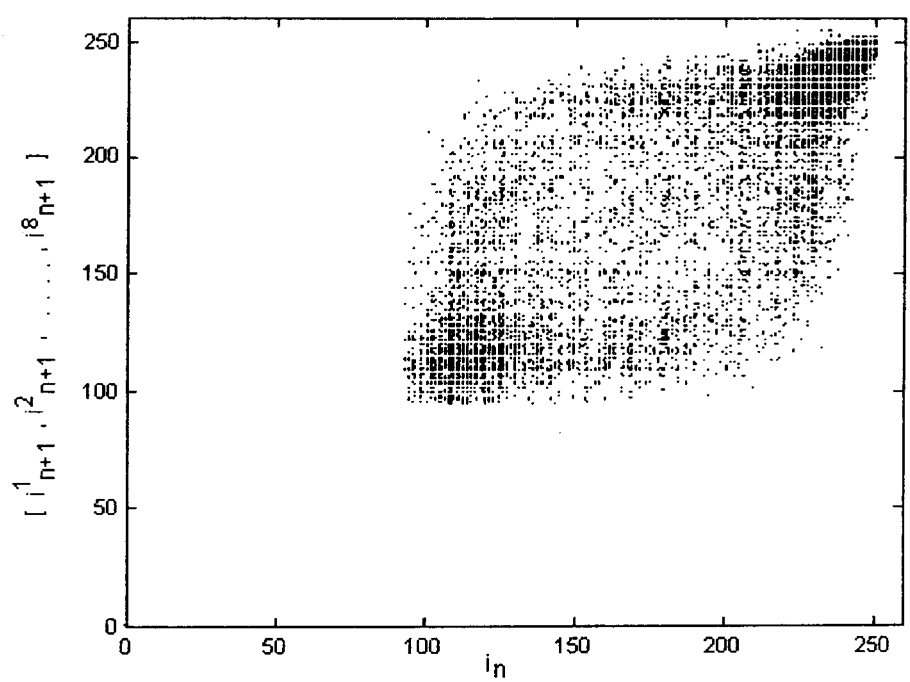
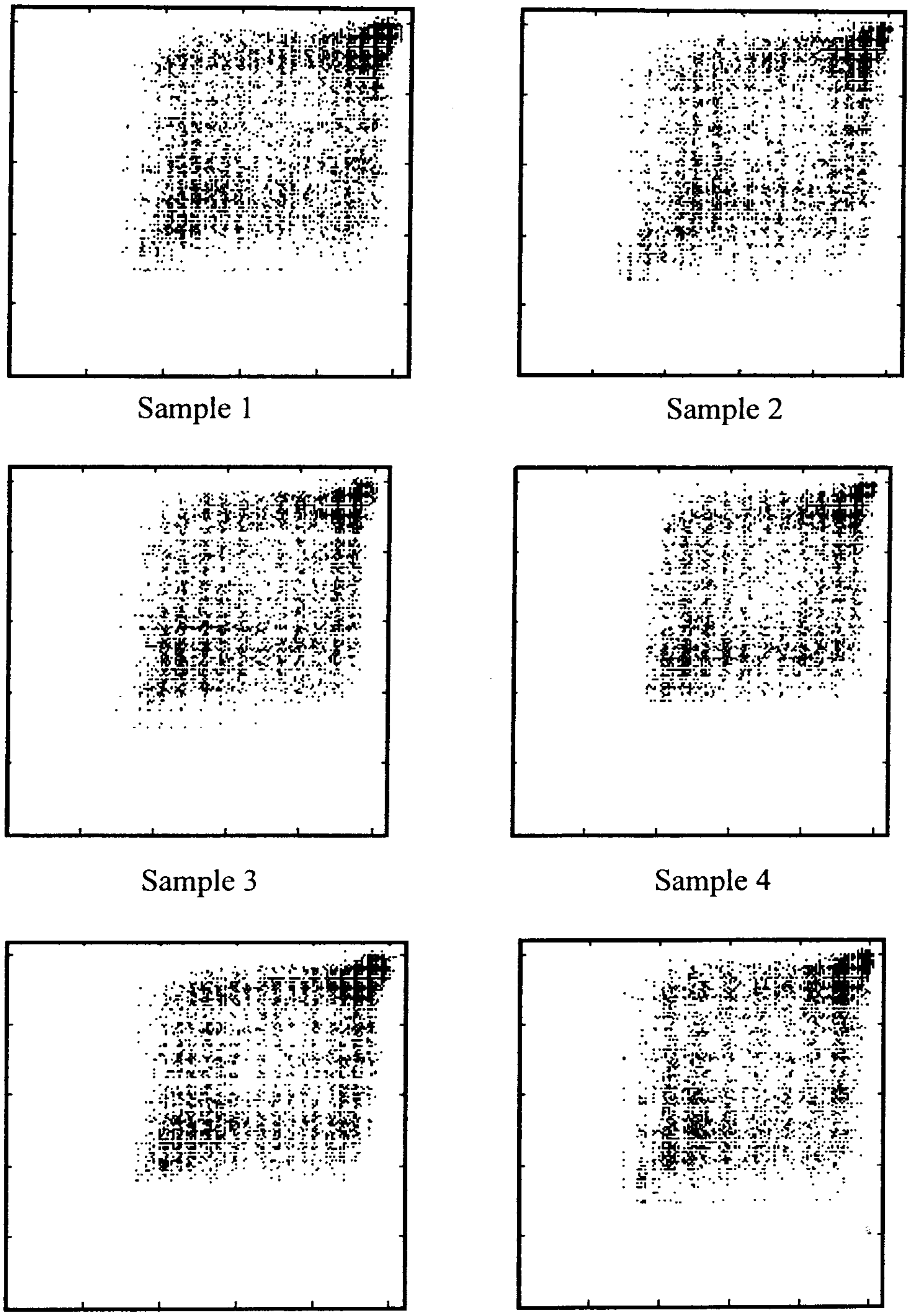


Fig. 5.5(f) SSM of Malayalam handwritten character റ[rha]

Figure 5.5(a-f): State-space maps for Malayalam handwritten characters അ [-ah], ഗ [ga], ട[ta], ഥ[thha], പ [pa] and റ[rha] with eight directional space variations

Manually inspecting this state-space maps with eight directional space variations of the forty four Malayalam characters, it is observed that more than three fourth of the characters are distinguishable from each other. The state-space maps with eight directional space variations generated for different samples of the Malayalam handwritten character ണ [-ah] are shown in figure 5.6. From these figures it is evident that the state-space maps generated for different samples of the same character are similar in nature.

The above results indicate that, state-space map can be effectively utilized for representing each Malayalam handwritten character since it is similar for different samples of the same character and differ from character to character. Moreover the state-space maps with eight directional space variations are found to be well-informed compared to the state-space map with one and four directional space variations and hence it is used further in the feature extraction stage. The following section describes the proposed method of parameter extraction from the state-space maps with eight directional space variations for Malayalam handwritten characters.



Sample 1 Sample 2

Sample 3 Sample 4

Sample 5 Sample 6

Figure 5.6: State-space maps for different samples of Malayalam handwritten characters ണ [-ah] with eight directional space variations

5.4 State-Space Point Distribution (SSPD) Features from the State- space Map

In handwritten character recognition problem, selection of feature extraction method is certainly the most important factor in achieving high recognition performance. This section describes a scheme for extracting features from the gray-scale images of the handwritten characters based on their state-space map with eight directional space variations. For this purpose the state-space map of the character image is divided into grids with 16x16 locations (boxes) as shown in figure 5.7. The box defined by the co-ordinates (0, 0), (1, 1) is taken as box one and box just right side to it is taken as box two and it is extended towards X direction with the last box being (15, 0), (16, 1) is taken as box sixteen. Then the next row is scanned from the beginning and boxes are numbered consecutively. The same process is repeated for all rows and finally 256 boxes are spotted. The State-Space Point Distribution (SSPD) for each pattern is calculated by estimating the number of points in each of these 256 boxes. Using the above information the State-Space Point Distribution graph is plotted by taking the box number in X-axis and the number of points in each box in Y-axis.

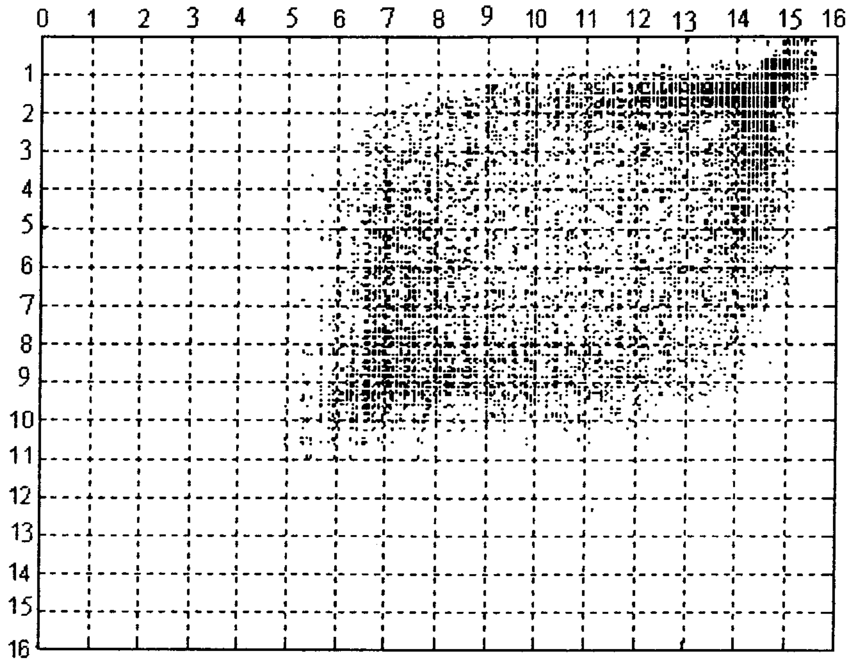


Figure 5.7 State-Space Map with 16 x16 locations

Figure 5.8 (a-f) shows the state-space point distribution graph for the selected Malayalam handwritten characters ണ [-ah], ഗ [ga], ട [ta], ഥ[thha], പ [pa] and റ[rha] obtained from the SSMs with eight directional space variations respectively. The state-space point distribution graph generated for different samples of the above characters are shown in figure 5.9 (a-f). The state- space map and its corresponding state-space point distribution graph obtained for different characters show the identity of that character as regard to the pattern. From the above analysis it is found that an efficient feature vector based on the SSPD can be effectively used to represent Malayalam handwritten characters and hence it can be set as an input to the recognition systems.

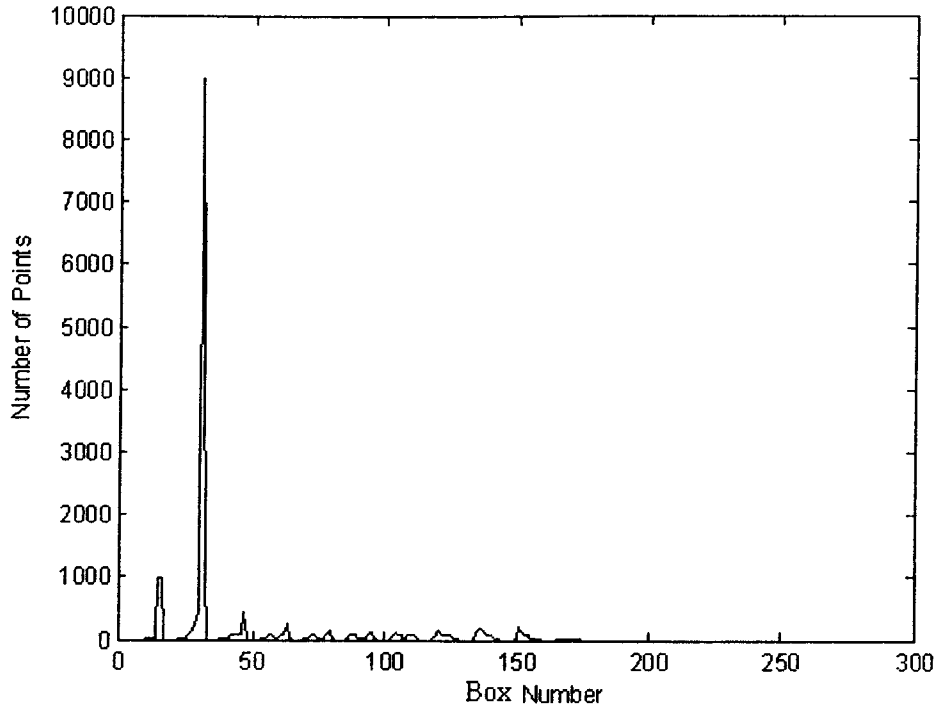


Fig. 5.8 (a) SSPD graph of Malayalam handwritten character ങ [-ah]

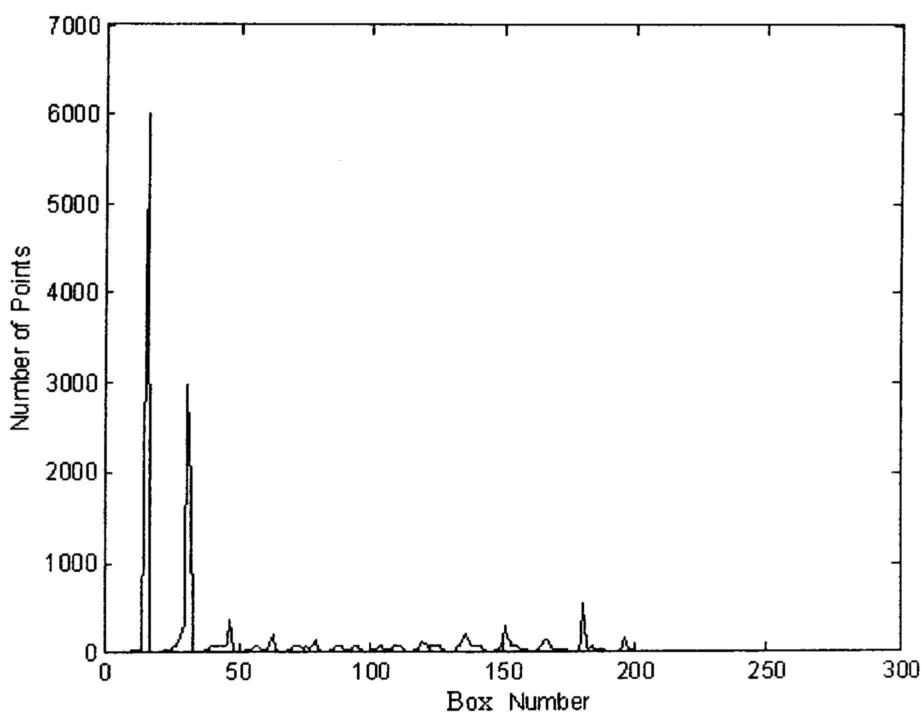


Fig. 5.8 (b) SSPD graph of Malayalam handwritten character ഗ[ga]

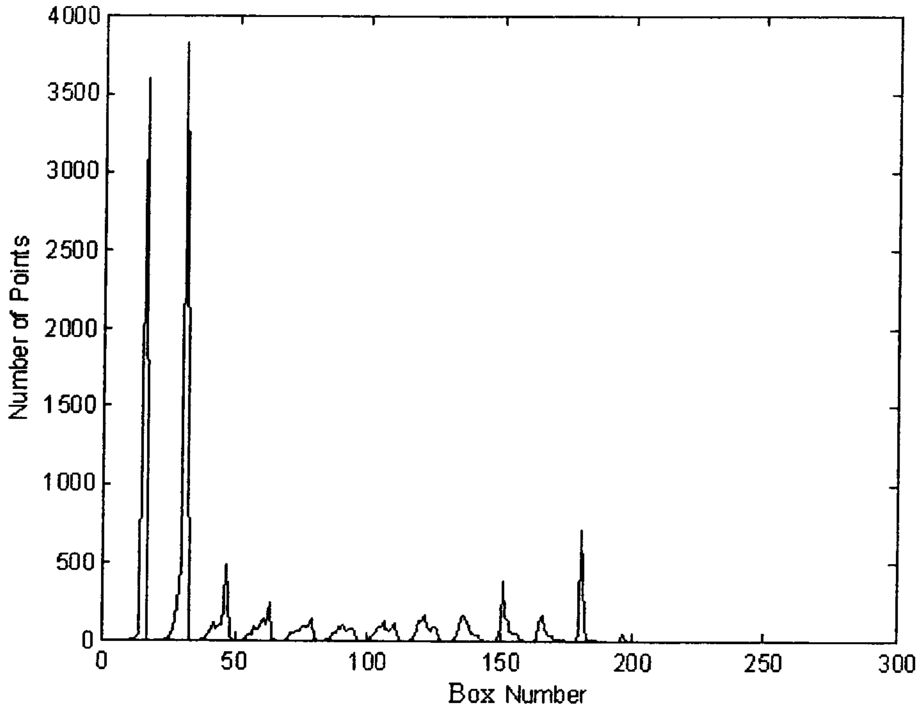


Fig. 5.8 (c) SSPD graph of Malayalam handwritten character ട[ta]

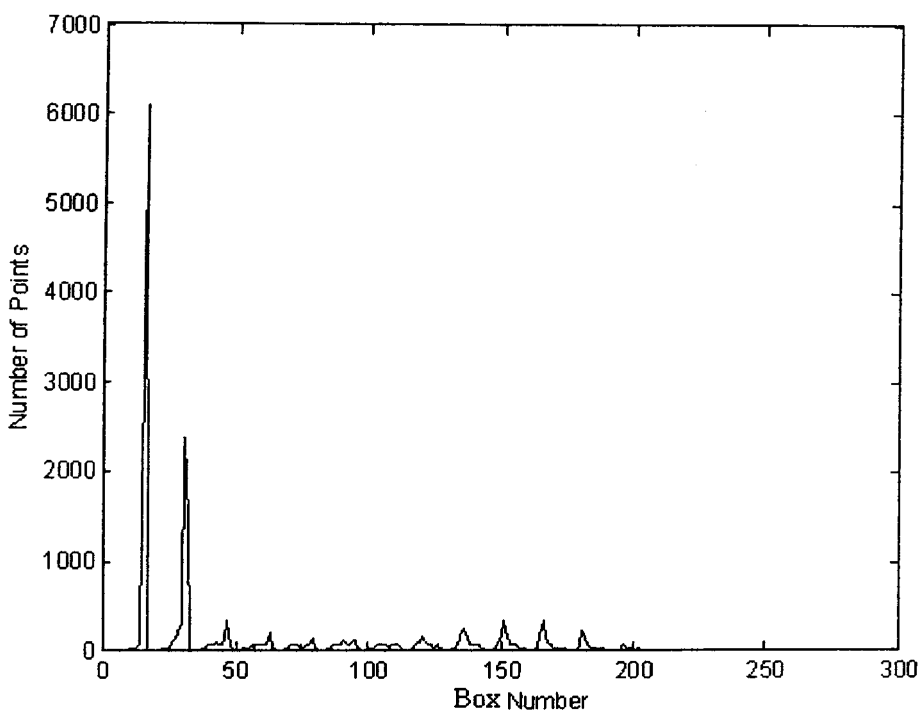


Fig. 5.8 (d) SSPD graph of Malayalam handwritten character ഥ[thha]

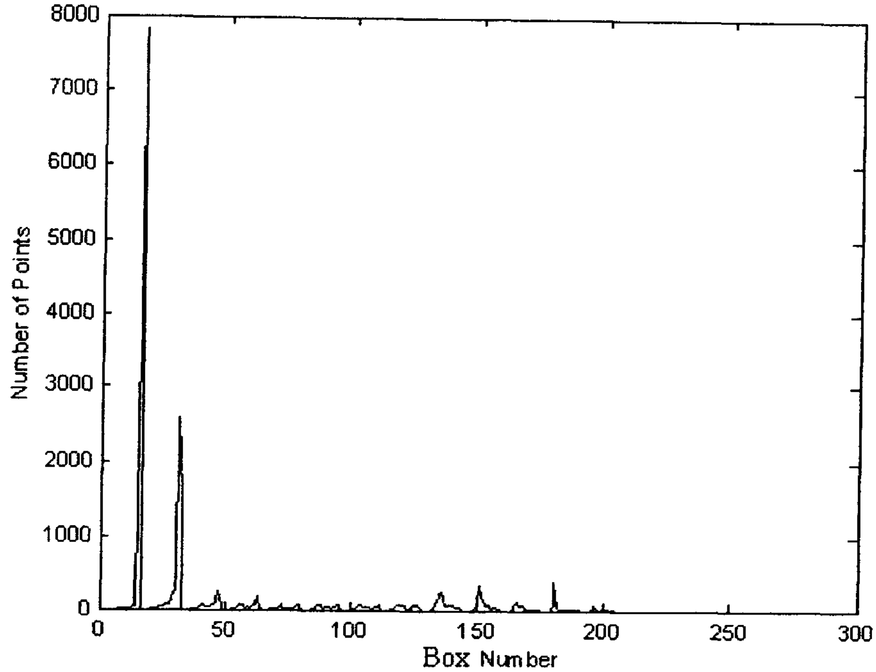


Fig. 5.8 (e) SSPD graph of Malayalam handwritten character പ [pa]

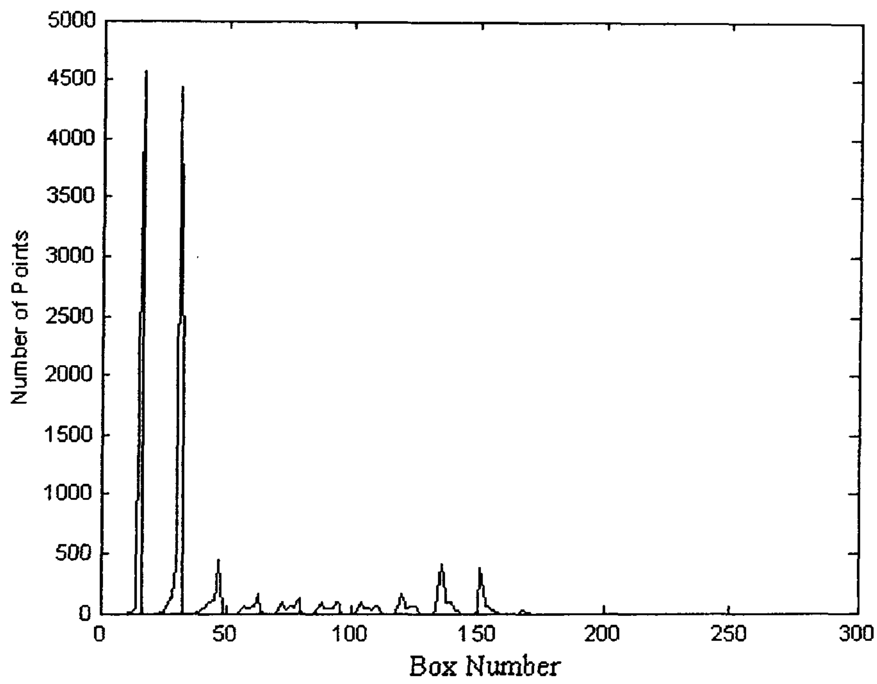


Fig. 5.8 (f) SSPD graph of Malayalam handwritten character റ[rha]

Figure 5.8 (a-f): SSPD graph of Malayalam handwritten characters ഃ [-ah], ഗ [ga], ട [ta], ഥ[thha], പ [pa] and റ[rha] obtained from state-space maps with eight directional space variations

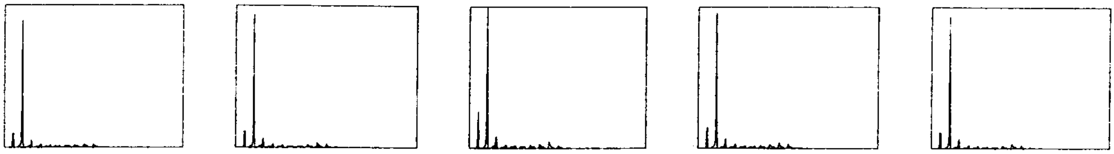


Fig. 5.9 (a) Character ണ [-ah]

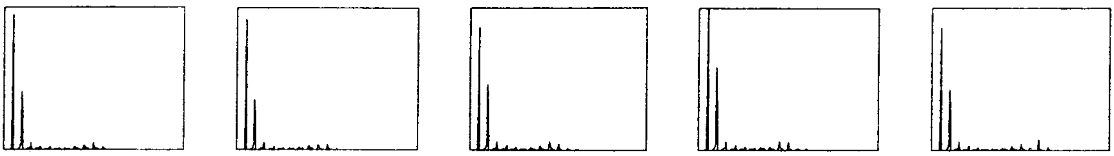


Fig. 5.9 (b) Character ഗ[ga]

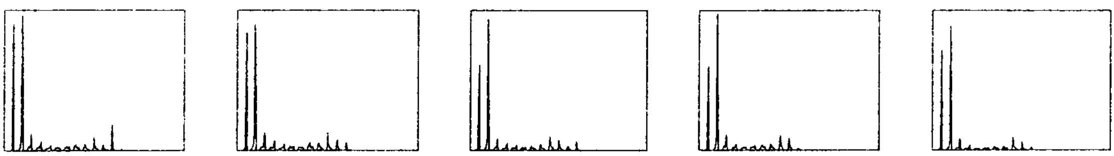


Fig. 5.9 (c) Character ട[ta]

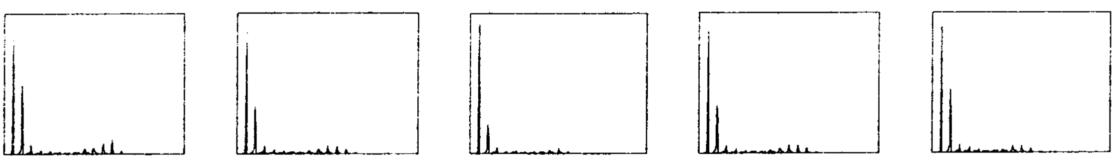


Fig. 5.9 (d) Character ഥ[thha]

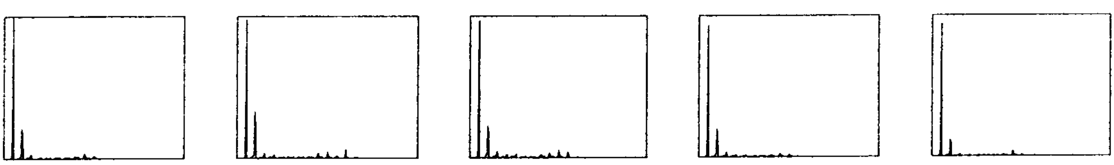


Fig. 5.9 (e) Character പ [pa]

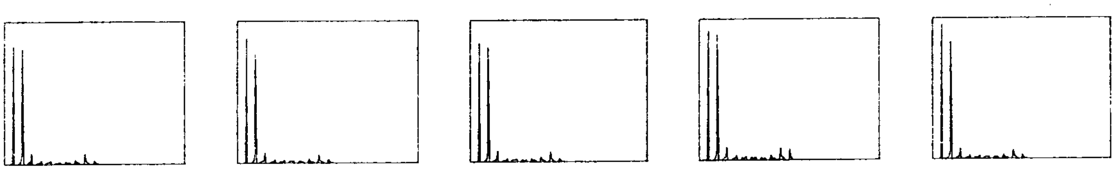


Fig. 5.9 (f) Character റ[rha]

Figure 5.9 (a-f): SSPD graphs of different samples of Malayalam handwritten characters ണ [-ah], ഗ [ga], ട [ta], ഥ[thha], പ [pa] and റ[rha] obtained from the SSMs with eight directional space variations

The feature vector is estimated by computing the average distribution of points in the sixteen consecutive boxes in the SSPD graph. Since there are 256 boxes in the SSPD graph, the feature vector dimension is reduced to sixteen. Figure 5.10 (a-c) shows the feature vector graphs obtained from Malayalam handwritten characters ങ്ങ [-ah], ഗ [ga] and ട [ta] plotted for fifteen samples of each character along with the mean curve. The graphs obtained for different samples of same character are similar where as graphs for different characters are highly distinguishable. Detailed studies on the recognition experiments based on the SSPD features are performed in succeeding chapters.

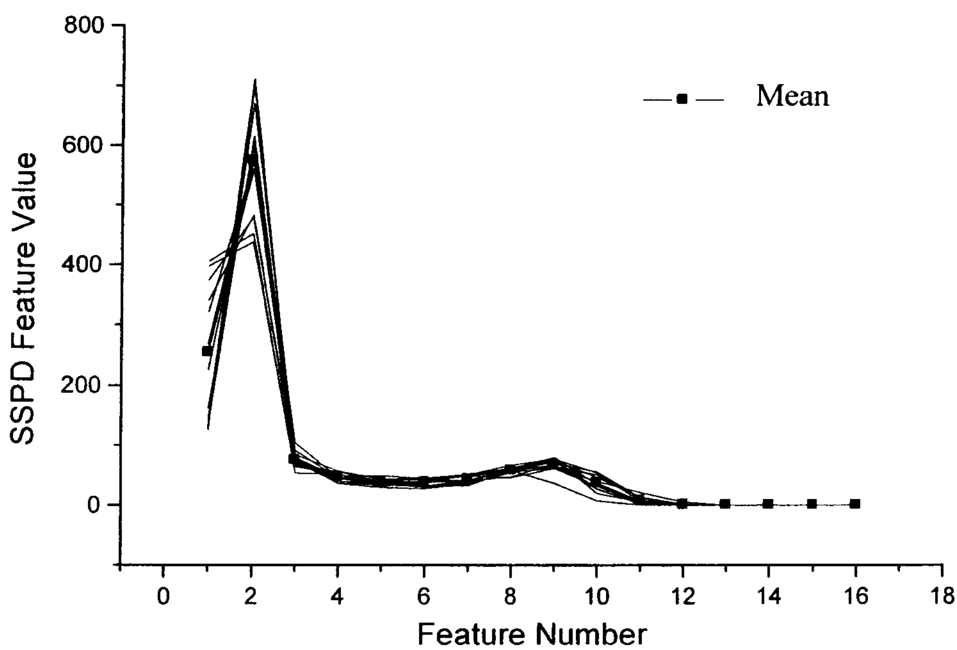


Figure 5.10(a) SSPD feature graph for Malayalam handwritten character ങ്ങ [-ah]

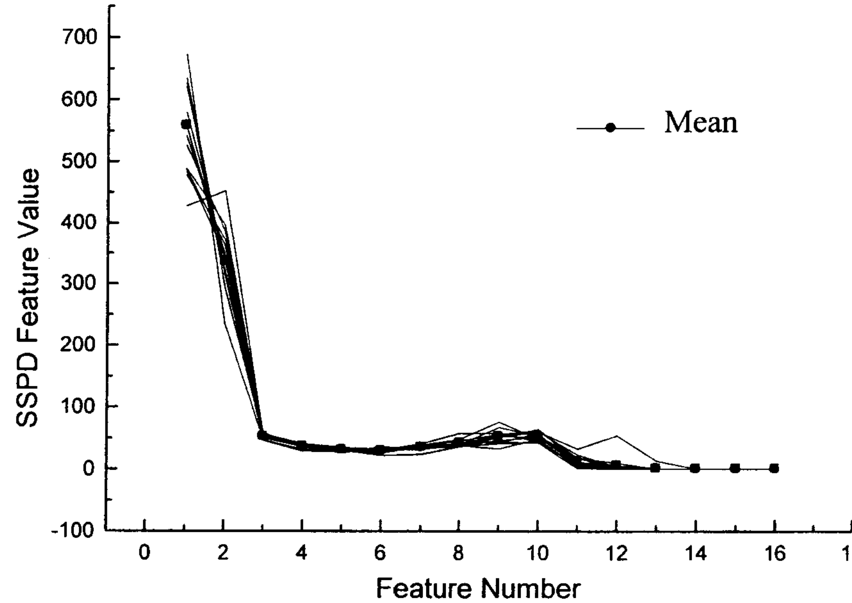


Figure 5.10 (b) SSPD feature graph for Malayalam handwritten character ഗ [ga]

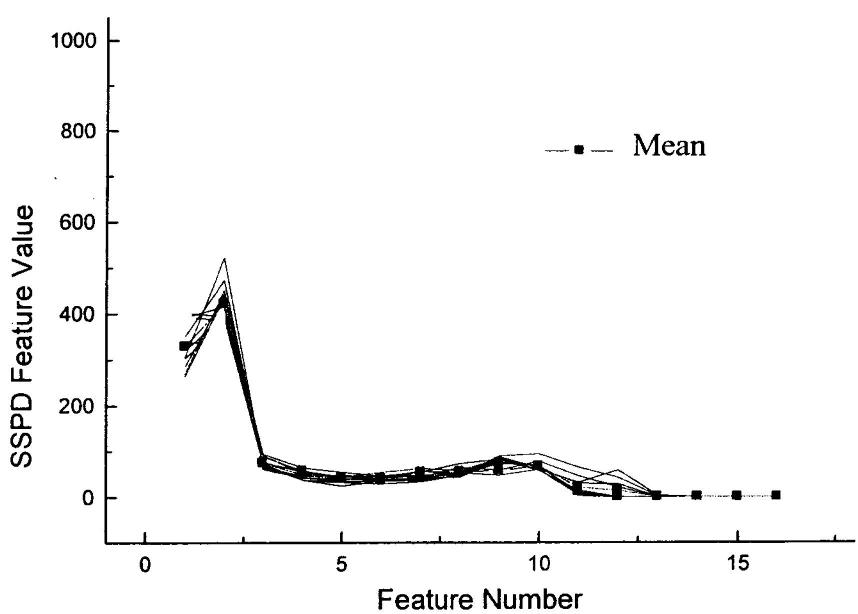


Figure 5.10(c) SSPD feature graph for Malayalam handwritten character ട [ta]

5.5 Conclusions

A novel method for modeling Malayalam handwritten characters based on the gray-scale images is introduced. Reconstructed state-spaces in the form of State-Space Maps (SSMs) for character images are generated for one, four and eight directional space variations. State-space maps with eight directional space variations are found to be well-informed and hence it is used further in the feature extraction process. Sate-Space Point Distribution (SSPD) features are extracted from each of the character samples for recognition purpose. The SSPD feature vector graph plotted for each character shows that the extracted features are similar for different image samples of a given character and at the same time they considerably vary from character to character.

The proposed method is very easy to implement since it does not require binarization. The difficulties in threshold selection and the subsequent loss of information caused by the process of binarization (that may results in broken strokes in binarized character image) are also eliminated in this method. Added advantage is that the feature extraction procedure is computationally less complex and the dimension of the feature vector (sixteen) is comparatively small. So the SSPD parameters derived from the gray-scale based SSMs of handwritten character samples can be effectively utilized for high speed HCR applications.

CHAPTER 6

HANDWRITTEN CHARACTER MODELING USING PERCEPTUAL FUZZY-ZONING

6.1 Introduction

In chapter 5 we introduced a new method for estimating gray-scale image based features for modeling Malayalam handwritten characters. A major problem with the gray-scale based feature extraction is that in some cases it may be difficult to identify the candidate character locations in the image due to the indistinguishable nature of the background noise and the foreground. An alternative method for modeling Malayalam handwritten characters from the pre-processed character image is proposed in this chapter. In order to remove the background noise the size normalized character images are binarized first. These binarized image samples are then thinned into single pixel wide skeletons. The algorithms used for performing the above said pre-processing techniques were already discussed in chapter 4. The pre-processed image samples are then used in the feature extraction process.

Research findings already reported in the area of pattern recognition reveal that the different feature extraction methods fulfill the requirement to a varying degree, depending on the specific recognition problem [Trier.O.D *et al.*, 1996]. Since Malayalam handwritten character patterns are of different nature compared to characters of other languages, it is highly essential to analyze the shape based characteristics of each character pattern before feature extraction process. That is, a study on global properties of character patterns will definitely help in identifying suitable features for recognition. Hence the properties of the forty four Malayalam handwritten characters that are included in the study, based on the aspect ratio, existence ratio and the centroid are identified in this work.

In general, the feature extraction methods for handwritten character recognition are based on two types of features: statistical and structural. The statistical features are derived from statistical distributions of pixels, such as zoning, moments, projection histograms or direction histograms. Structural features are based on topological and geometrical properties of the character, like strokes and their directions, end-points or intersection of segments and loops [Simon.J.C *et al.*, 1980], [Jain.A.K *et al.*, 2000]. The shape analysis of handwritten character images performed using the aspect ratio, existence ratio and the centroid, point up that in the case of Malayalam language, characters of the alphabet are rich in shape and they are subjected to many variations in terms of handwriting styles. So the integration of above stated structural and statistical information is a better solution to accommodate the large variations observed on the handwritten character images and highlight different character properties. Since these two types of features can be considered complementary in nature, the features representing the structural properties of different regions of the character image extracted with the aid of partial decomposition (zoning) should work efficiently for the Malayalam scripts. The first part of this chapter describes the shape analysis of Malayalam characters. In the second section two novel feature extraction methods, which are capable of extracting the structural properties of the character curves based on perceptual zoning, are introduced.

6.2 Shape Analysis of Malayalam Handwritten Characters

The section presents the analysis of shape-based characteristics of Malayalam handwritten characters. Some Malayalam characters often look a lot alike and hence the possibility of confusing them with one another while conducting the recognition experiments is high. Figure 6.1 presents the group of Malayalam characters that are similar in structural features. It is evident that the structural properties of these characters in each group are almost same. So it is extremely difficult to list the structural features that distinguish those characters present in each group.

From the preliminary analysis of the basic structure and spatial allocation of Malayalam characters, it is evident that these different groups of Malayalam characters have many features in common and so they will tend to be confused while considering only the structural features for recognition. The average aspect ratio, existence ratio and centroid of forty four basic Malayalam characters along with their root mean square (r.m.s) deviations are estimated as discussed in the following section.

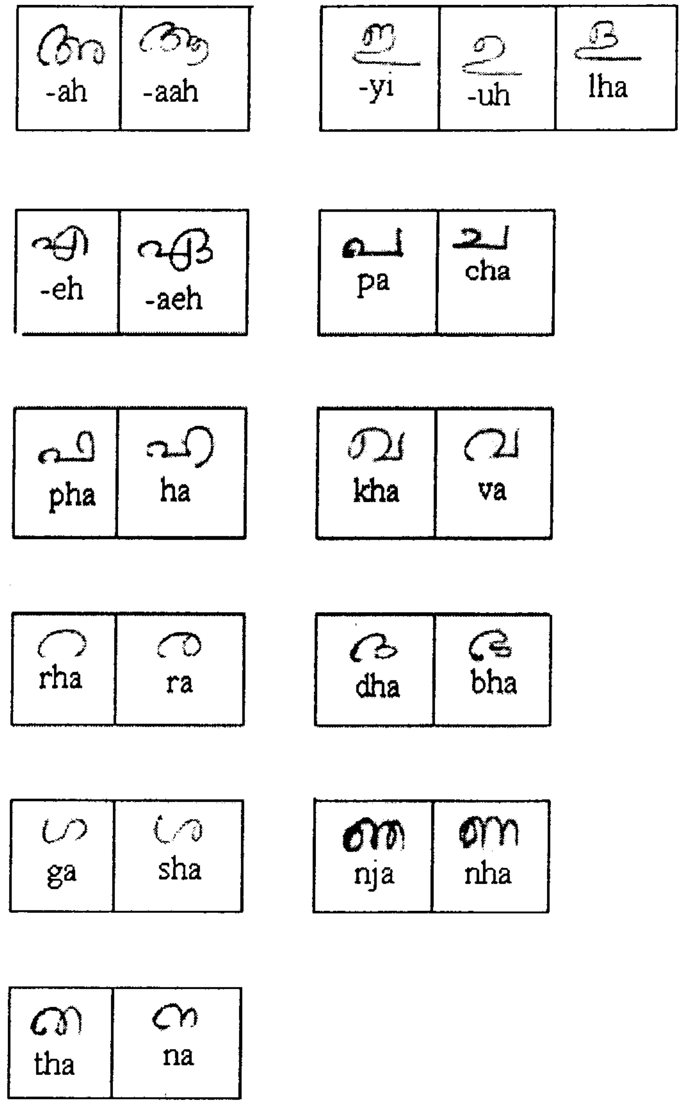


Figure 6.1 Malayalam handwritten characters with similarity in structural features

6.2.1 Aspect Ratio of Malayalam Handwritten Characters

Aspect ratio is defined as the height by width ratio of a character pattern. To compute the aspect ratio the binarised character images are initially cropped into minimum rectangle and the height and width of that rectangle which gives corresponding aspect ratio is found out. The root mean square (r.m.s.) deviations of corresponding aspect ratios for all the characters are also computed. Table 6.1 shows the average aspect ratios of the 716 samples each of forty four Malayalam handwritten characters along with its r.m.s. deviation.

Character Number	Character	Average Aspect Ratio	r.m.s. Deviation	Character Number	Character	Average Aspect Ratio	r.m.s. Deviation
1	അ [-ah]	0.245040	0.024964	23	ണ[nha]	0.238731	0.024126
2	ആ[-aah]	0.333574	0.022236	24	ത [tha]	0.270585	0.024825
3	ഇ [-yi]	0.257479	0.030761	25	ഥ [thha]	0.426270	0.040084
4	ഉ [-uh]	0.353086	0.040186	26	ദ [dha]	0.514277	0.034723
5	ഋ [-eru]	0.495473	0.019608	27	ധ[dhha]	0.410626	0.025870
6	എ [-eh]	0.373648	0.033154	28	ന [na]	0.301681	0.026620
7	ഏ[-aeh]	0.312460	0.021983	29	പ [pa]	0.232511	0.029369
8	ഒ [-oh]	0.514080	0.051725	30	ഫ [pha]	0.272055	0.024126
9	ക [ka]	0.469593	0.029649	31	ബ [ba]	0.210377	0.013746
10	ഖ [kha]	0.342810	0.027655	32	ഭ [bha]	0.397611	0.023169
11	ഗ [ga]	0.371966	0.029466	33	മ [ma]	0.494427	0.034753
12	ഘ [gha]	0.317620	0.025109	34	യ [ya]	0.310232	0.034560
13	ങ [nga]	0.337968	0.026097	35	ര [ra]	0.310964	0.036524
14	ച [cha]	0.285815	0.026877	36	ല [la]	0.365762	0.031663
15	ഛ[chha]	0.207893	0.012593	37	വ [va]	0.350155	0.022131
16	ജ [ja]	0.396542	0.039218	38	ശ [sha]	0.368235	0.028737
17	ഝ[jha]	0.214683	0.022534	39	ഷ[shha]	0.374217	0.021590
18	ഞ [nja]	0.250719	0.021421	40	സ [sa]	0.349704	0.042320
19	ട [ta]	0.650080	0.065690	41	ഹ [ha]	0.272440	0.016852
20	ഠ[tta]	0.611888	0.054973	42	ള [lha]	0.317387	0.058082
21	ഡ [da]	0.307560	0.039422	43	ഴ [zha]	0.686336	0.045801
22	ഢ [dda]	0.300096	0.048678	44	റ [rha]	0.365557	0.040562

Table 6.1 Average aspect ratios of forty four Malayalam handwritten characters with root mean square (r.m.s.) deviation

From the table it can be seen that the aspect ratios of the Malayalam handwritten characters vary within the interval 0.20789 to 0.68634 for the collected samples. The smallest average aspect ratio is for the character ഛ[chha] and the highest value is obtained for the character ഴ [zha]. The root mean square (r.m.s) deviations vary in the range 0.01259 to 0.06569. The smallest deviation is acquired for the character ഛ[chha] and the highest deviation is shown by the character ട[ta]. Figure 6.2 shows the graph plotted for the average aspect ratios of the forty four Malayalam characters.

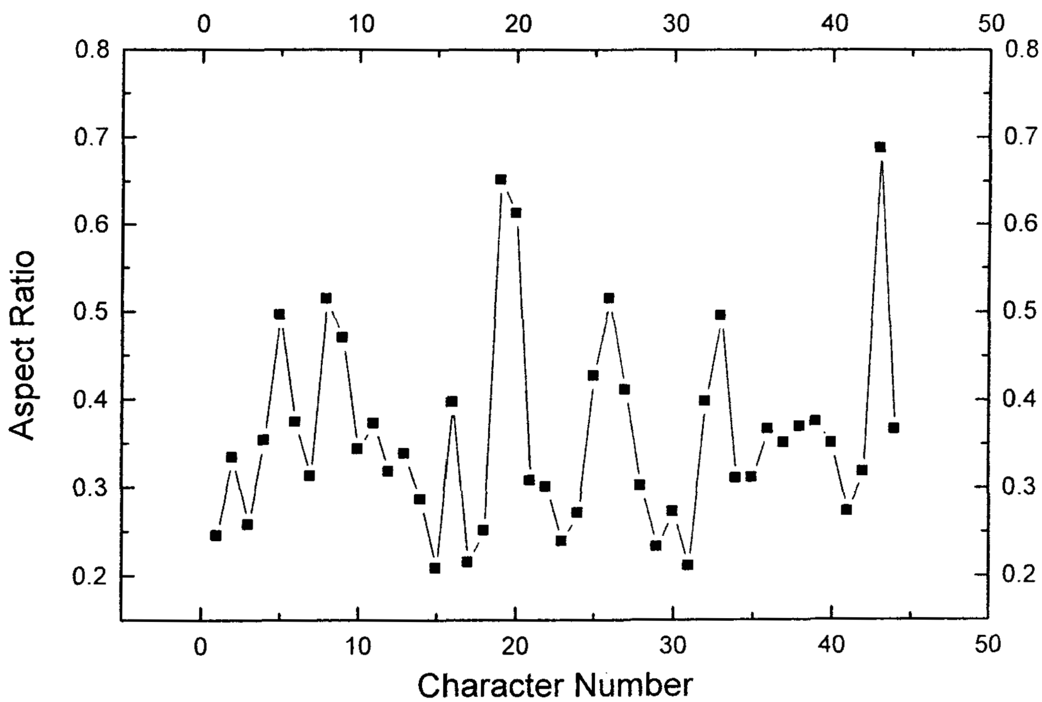


Figure 6.2 Average aspect ratios of forty four Malayalam handwritten characters

The graph indicates that it is hard to classify the characters based on the aspect ratios of the characters. Even though the aspect ratios alone can't be considered as a feature for classification, it gives us a better understanding about the shape of Malayalam Characters.

6.2.2 Existence Ratio of Malayalam Handwritten Characters

Existence ratio is defined as the ratio of the number of black and white pixels in a character image pattern. The average existence ratio and its root mean square deviation computed for the forty four basic Malayalam handwritten characters from the 716 samples of each character are tabulated in Table 6.2. From the table it can be seen that the existence ratios vary in the range 0.04373 to 0.08721 for the collected samples of Malayalam characters. The smallest average existence ratio is for the character പ[pa] and the highest value is obtained for the character ജ [ja].

The root mean square deviations vary in the range 0.01583 to 0.006507. The smallest deviation is acquired for the character ഹ[ha] and the highest deviation is shown by the character ട[ta]. It is also noted that the root mean square (r.m.s) deviations obtained for existence ratios are much less than that obtained in the case of aspect ratios. Figure 6.3 shows the graph plotted for the average existence ratio of the forty- four basic characters.

Character Number	Character	Average Existence Ratio	r.m.s. Deviation	Character Number	Character	Average Existence Ratio	r.m.s. Deviation
1	അ [-ah]	0.078453	0.004809	23	ണ[nha]	0.073230	0.001759
2	ആ[-aah]	0.073229	0.004959	24	ത [tha]	0.065432	0.001823
3	ഇ [-yi]	0.068182	0.003223	25	ഥ [thha]	0.055671	0.002396
4	ഉ [-uh]	0.059123	0.003120	26	ദ [dha]	0.053269	0.003572
5	ഋ [-eru]	0.083207	0.002022	27	ധ[dhha]	0.050006	0.001751
6	എ [-eh]	0.069448	0.002537	28	ന [na]	0.053161	0.002077
7	ഏ[-aeh]	0.071321	0.001966	29	പ [pa]	0.043728	0.001956
8	ഒ [-oh]	0.065459	0.001806	30	ഫ [pha]	0.055701	0.002908
9	ക [ka]	0.070042	0.002134	31	ബ [ba]	0.073010	0.003310
10	ഖ [kha]	0.064593	0.002392	32	ഭ [bha]	0.060288	0.003372
11	ഗ [ga]	0.045707	0.002208	33	മ [ma]	0.069854	0.003367
12	ഘ [gha]	0.061538	0.004600	34	യ [ya]	0.072180	0.004268
13	ങ [nga]	0.067753	0.002437	35	ര [ra]	0.056859	0.001839
14	ച [cha]	0.048259	0.002296	36	ല [la]	0.060623	0.006049
15	ഛ[chha]	0.062784	0.003999	37	വ [va]	0.054164	0.003099
16	ജ [ja]	0.087213	0.002533	38	ശ [sha]	0.054903	0.002814
17	ഝ[jha]	0.080017	0.004897	39	ഷ[shha]	0.071584	0.002978
18	ഞ [nja]	0.076160	0.003167	40	സ [sa]	0.053165	0.002951
19	ട [ta]	0.054076	0.006507	41	ഹ [ha]	0.049949	0.001583
20	ഠ[tta]	0.050790	0.003233	42	ള [lha]	0.067015	0.004515
21	ഡ [da]	0.056884	0.001605	43	ഴ [zha]	0.063913	0.001838
22	ഢ [dda]	0.066768	0.003248	44	റ [rha]	0.043865	0.002032

Table 6.2 Existence ratio of the Malayalam handwritten characters with root mean square (r.m.s) deviation

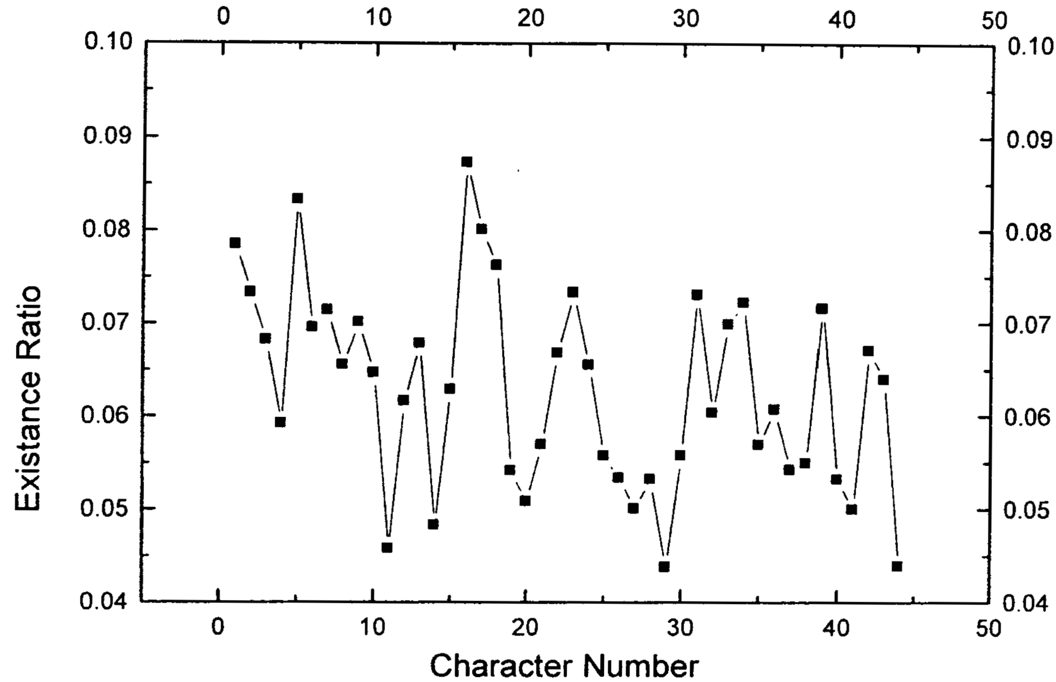


Figure 6. 3. Average existence ratios of forty four Malayalam handwritten characters

The graph indicates that it is hard to group the characters based on the existence ratios of the characters. Even though the existence ratios alone can't be considered as a feature for classification, it can be used as a parameter to identify the statistical distribution of each character curve and hence provide better information about its shape.

6.2.3 Centroid of Malayalam Handwritten Characters

The centroid of a character pattern is a single point (C_x, C_y) defined by the average X and average Y co-ordinates of each black pixel that constitute that pattern. The X and Y co-ordinate values of the centroid C_x and C_y of a character curve can be estimated as,

$$C_x = \frac{1}{N}\sum_{i=1}^{N} x_i \quad \text{and} \quad C_y = \frac{1}{N}\sum_{i=1}^{N} y_i$$

where N is the total number of black pixels that constitute the character curve and x_i and y_i are the X and Y co-ordinate values of those black pixels.

Table 6.3 shows the average centroid values computed for the forty four Malayalam handwritten characters by considering 716 samples of each character. From the table it can be seen that the X values of the centroid vary in the range 17.886 to 27.249 for the collected samples of Malayalam characters. The smallest average X value for the centroid is acquired for the character ൌ[-aah] and the highest value is obtained for the character പ[pa]. The average Y values of the centroid vary in the range 23.989 to 38.623 for the collected samples. The smallest average Y value for the centroid is acquired for the character ൈ[-yi] and the highest value is obtained for the character ഛ[chha].

The average r.m.s. deviation of centroid is tabulated in Table 6.4. From the table it can be seen that the r.m.s deviation for X values of the centroid vary in the range 0.432 to 2.283 for the collected samples of Malayalam characters. The smallest r.m.s deviation for X value of the centroid is acquired for the character ഡ[da] and the highest value is obtained for the character ട[ta]. The average r.m.s deviations for Y values of the centroid vary in the range 0.401 to 2.921 for the collected samples. The smallest r.m.s deviation for *Y* value of the centroid is acquired for the character ണ[nha] and the highest value is obtained for the character ട [ta].

Figure 6.4 shows the scatter plot drawn for the average centroids of the forty four Malayalam characters. From the graph it is evident that the grouping or classification of character patterns based on centroid is not feasible. But the scatter plot seems to be an aid to understand the complexity of the character patterns to be recognized.

The above shape analysis performed on Malayalam handwritten characters based on aspect ratio, existence ratio and centroid indicates that it is extremely difficult to classify the character patterns based on the global structural or statistical features. It is also evident that these global features considerably deviate from the norm or average. Therefore the studies in the direction of designing parameters that are capable of representing the information content regions in a character pattern are essential. The following section describes new feature extraction methods proposed for Malayalam handwritten characters based on region decomposition or perceptual zoning.

Character Number	Character	Centroid (Cx,Cy)	Character Number	Character	Centroid (Cx,Cy)
1	അ [-ah]	(22.072, 31.859)	23	ണ [nha]	(20.105, 30.397)
2	ആ[-aah]	(17.886, 33.464)	24	ത [tha]	(20.203, 34.095)
3	ഇ [-yi]	(23.918, 23.989)	25	ഥ [thha]	(24.523, 32.012)
4	ഉ [-uh]	(25.405, 27.265)	26	ദ [dha]	(22.120, 32.010)
5	ഋ [-eru]	(20.535, 32.169)	27	ധ[dhha]	(24.782, 31.429)
6	എ [-eh]	(21.417, 37.401)	28	ന [na]	(19.433, 32.936)
7	ഏ [-aeh]	(21.724, 37.241)	29	പ [pa]	(27.249, 35.441)
8	ഒ [-oh]	(22.319, 31.659)	30	ഫ [pha]	(25.933, 38.262)
9	ക [ka]	(25.115, 30.779)	31	ബ [ba]	(21.981, 32.654)
10	ഖ [kha]	(21.893, 32.521)	32	ഭ [bha]	(21.291, 35.754)
11	ഗ [ga]	(22.262, 31.329)	33	മ [ma]	(23.414, 35.145)
12	ഘ [gha]	(20.581, 32.869)	34	യ [ya]	(21.880, 34.169)
13	ങ [nga]	(19.669, 30.122)	35	ര [ra]	(20.457, 33.791)
14	ച [cha]	(26.269, 36.135)	36	ല [la]	(23.587, 29.563)
15	ഛ [chha]	(22.958, 38.623)	37	വ [va]	(21.286, 34.282)
16	ജ [ja]	(23.541, 29.235)	38	ശ [sha]	(22.625, 32.207)
17	ഝ [jha]	(21.093, 32.955)	39	ഷ [shha]	(24.828, 36.538)
18	ഞ [nja]	(19.942, 31.426)	40	സ [sa]	(23.642, 35.247)
19	ട [ta]	(24.259, 33.351)	41	ഹ [ha]	(22.879, 36.650)
20	ഠ[tta]	(21.902, 30.739)	42	ള [lha]	(25.152, 27.131)
21	ഡ [da]	(21.308, 34.764)	43	ഴ [zha]	(21.161, 35.511)
22	ഢ [dda]	(21.607, 36.748)	44	റ [rha]	(18.467, 33.266)

Table 6.3: Centroid of the forty four Malayalam handwritten characters

Character Number	Character	r.m.s Deviation from Centroid	Character Number	Character	r.m.s Deviation from Centroid
1	അ [-ah]	(0.969, 1.139)	23	ണ [nha]	(0.504, 0.401)
2	ആ[-aah]	(0.747, 0.989)	24	ത [tha]	(0.694, 0.833)
3	ഇ [-yi]	(0.475, 1.505)	25	ഥ [thha]	(1.072, 0.609)
4	ഉ [-uh]	(0.891, 1.559)	26	ദ [dha]	(0.938, 2.066)
5	ഋ [-eru]	(0.608, 0.856)	27	ധ[dhha]	(0.498, 0.703)
6	എ [-eh]	(0.867, 1.158)	28	ന [na]	(0.507, 0.969)
7	ഏ [-aeh]	(0.613, 0.669)	29	പ [pa]	(0.788, 0.818)
8	ഒ [-oh]	(0.799, 0.994)	30	ഫ [pha]	(0.595, 0.868)
9	ക [ka]	(0.795, 0.904)	31	ബ [ba]	(1.015, 0.841)
10	ഖ [kha]	(0.844, 1.190)	32	ഭ [bha]	(0.689, 0.741)
11	ഗ [ga]	(0.564, 1.219)	33	മ [ma]	(0.530, 0.924)
12	ഘ [gha]	(1.676, 1.577)	34	യ [ya]	(1.433, 1.287)
13	ങ [nga]	(0.794, 0.757)	35	ര [ra]	(0.664, 0.461)
14	ച [cha]	(1.090, 1.022)	36	ല [la]	(1.484, 1.873)
15	ഛ [chha]	(2.258, 1.390)	37	വ [va]	(0.887, 1.932)
16	ജ [ja]	(0.715, 1.283)	38	ശ [sha]	(1.008, 0.509)
17	ഝ [jha]	(1.100, 1.089)	39	ഷ [shha]	(0.679, 1.195)
18	ഞ [nja]	(0.537, 0.775)	40	സ [sa]	(1.189, 1.173)
19	ട [ta]	(2.283, 2.921)	41	ഹ [ha]	(0.861, 0.597)
20	ഠ[tta]	(0.769, 1.863)	42	ള [lha]	(0.823, 2.571)
21	ഡ [da]	(0.432, 0.827)	43	ഴ [zha]	(1.044, 0.620)
22	ഢ [dda]	(0.703, 1.069)	44	റ [rha]	(0.504, 0.591)

Table 6.4: Root mean square (r.m.s) deviation in centroid for the forty four Malayalam handwritten characters

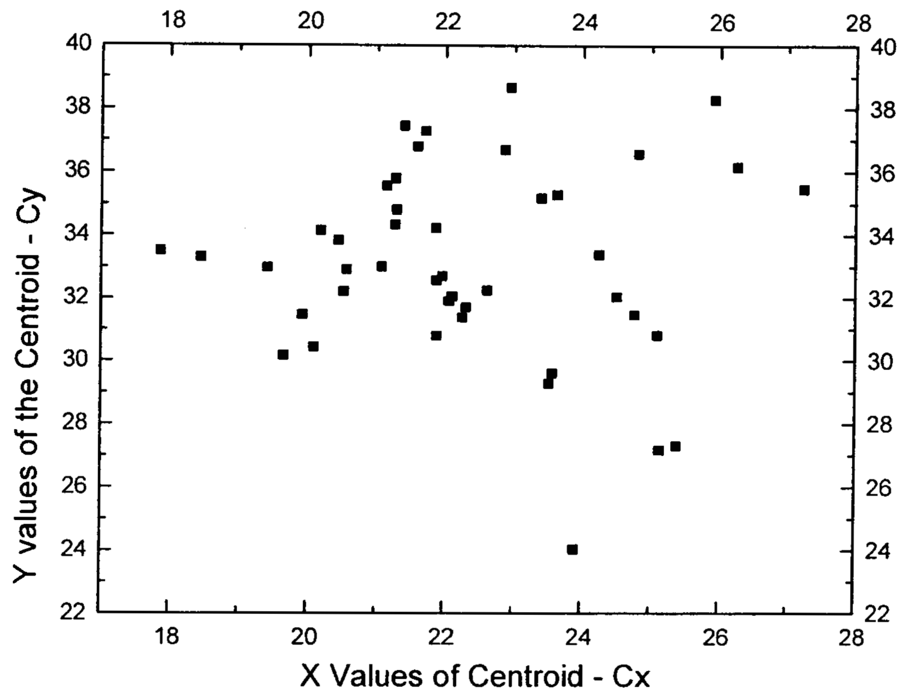


Figure 6.4 Centroids of forty four Malayalam handwritten characters

6.3 Perceptual Zoning for Handwritten Character Modeling

Perceptual zoning or region decomposition can be viewed as a mechanism for local information analysis on partitions of a given pattern and resembles that of human perception. This method investigates the recognition of patterns based on their parts, by identifying potential parameters when an uncertainty or confusion occurs at a given point. In this perspective it is necessary to locate the meaningful region of the character curve for effective and efficient recognition [Sekuler.R and Blake.R, 1994], [Schomaker.L and Segers.E, 1998]. But the challenging task is to identify the most meaningful part or information content region, which helps to identify each character. In most of the character recognition applications zoning is mainly applied to compute the percentage of black pixels in each zone. Remarkable results are reported in the literature on handwritten character recognition in various languages based on this method [Kimura.F and Sridhar.M, 1991], [Takahashi.H, 1991], [Bokser.M, 1992]. But the non-uniform sized zones used in most of these handwritten character recognition studies are not much helpful for Malayalam characters. It is mainly because of the diverse structural characteristics of Malayalam characters. It's encompassing round diffusion differentiates it from other scripts and numerals (where the non-uniform sized zoning techniques are mainly used), which have vertical or horizontal dispersion.

In the present study we propose two feature extraction techniques specially designed for the Malayalam characters by dividing the character patterns into nine uniform zones. Firstly, zoning method is implemented with the vector distance features and later it's modified version is proposed using fuzzy-zoning method with normalized vector distance feature as discussed in the following sections.

6.3.1 Zoned Vector Distance (Z-VD) Features for Handwritten Character Modeling

In this section a new feature extraction scheme for Malayalam handwritten characters based on zoned vector distance measure is introduced. From the analysis conducted on the aspect ratios of Malayalam handwritten characters, in section 6.2.1, it is observed that the ratios vary in the range 0.20789 to 0.68634. So it is difficult to enclose every Malayalam character in a standard rectangular grid as done in much of the character recognition works in other languages. Even if we try to enclose Malayalam characters within a standard rectangular structure, the rectangle should be large enough to accommodate the largest character. Splitting this into equal zones, results in each zone being abnormally large with respect to small characters and some zones will remain without any character curve information at all. This is the serious anomaly possessed while enclosing the non-uniform sized Malayalam handwritten characters in a standard rectangular structure. In order to overcome this problem, before applying zoning, we have normalized all the character samples into 66x42 pixel size window using Affine Transformation and bilinear interpolation technique. The output window size 66x42 pixels is arrived by trial and error experiment.

The size normalized character images are then binarised using Otsu's global thresholding algorithm. The rotation invariant thinning algorithm is applied on the binarised character images to get single pixel wide character skeletons. The detailed descriptions of the above stated pre-processing algorithms were discussed in chapter 4.

On enclosing the size normalized Malayalam handwritten characters in a standard minimum rectangle, in almost all the cases, the character curve is found to be equally distributed in all the zones of enclosing rectangle. In this study we propose a zoning method in which the standard windows, constructed for enclosing the size normalized character images, are partitioned into nine (3x3) uniform zones. Figure 6.5 shows the zoning applied on the pre-processed Malayalam handwritten character ഞ [-ah].

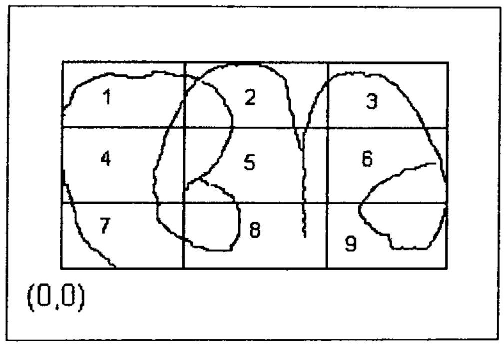


Figure 6.5 Zoning applied on Malayalam character ഞ [-ah]

The structural property of the character curve present in each zone is obtained by estimating the sum of the vector distances of each pixel on the curve from the absolute origin. For this purpose the bottom right corner of the minimum rectangle enclosing the character curve is considered as the absolute origin (0,0). The vector distance of k^{th} pixel $p_k(x, y)$ on the character curve in the r^{th} region out of nine regions at location (x, y) is computed by the expression,

$$dr_k = (x^2 + y^2)^{1/2}$$

The zoned vector distances Z-VD_r for each region is computed by adding the vector distances dr_k computed for all the black pixels of the character curve in that particular zone with the expression

$$\text{Z-VD}_r = \sum_{k=1}^{N_r} dr_k \quad \text{for } r = 1, 2, \ldots, 9.$$

where N_r is the number of pixels in r^{th} region

The proposed zoning mechanism with vector distance feature extraction method to model handwritten Malayalam characters is summarized in the following algorithm.

Algorithm: Feature extraction based on zoned vector distance (Z-VD)

Step 1: Crop the handwritten character image sample taken from the database in to minimum rectangle such that the character curve fits exactly to the rectangle.

Step 2: Apply size normalization and map the character image to a standard rectangle of size 66x42 pixels and perform thresholding and thinning on each character image.

Step 3: Divide the resultant image into 3x3 equal zones.

Step 4: Compute the vector distance (dr_k) for character curve included in each zone by considering the bottom left corner as the absolute origin (0, 0). The vector distance of k^{th} pixel in r^{th} zone at location (x, y) is given by the expression

$$dr_k = (x^2 + y^2)^{1/2}$$

Step 5: The zoned vector distances Z-VD_r is estimated by computing the sum of vector distances of all black pixels in a particular zone using the expression,

$$Z\text{-}VD_r = \sum_{k=1}^{N_r} dr_k \quad \text{for } r = 1, 2, \ldots, 9.$$

where N_r is the number of pixels in r^{th} region

Step 6: The nine feature elements computed in step 5 constitute the final feature vector Z-VD.

Figure 6.6(a-f) shows the zone number r versus zoned vector distance Z-VD_r feature graph plotted for the different samples of selected Malayalam characters. In the graph, the slim lines represent the feature graph obtained using the repeated samples of the same character written by different writers. The bold line represents the average of the above said multiple curves. Analysis of these feature graphs drawn for forty four Malayalam handwritten character shows that for different samples of the same character, the distribution of Z-VD features are similar in nature and at the same time it is distinguishable for different characters. The recognition experiments based on the Z-VD features using different pattern classification techniques are presented in the two succeeding chapters.

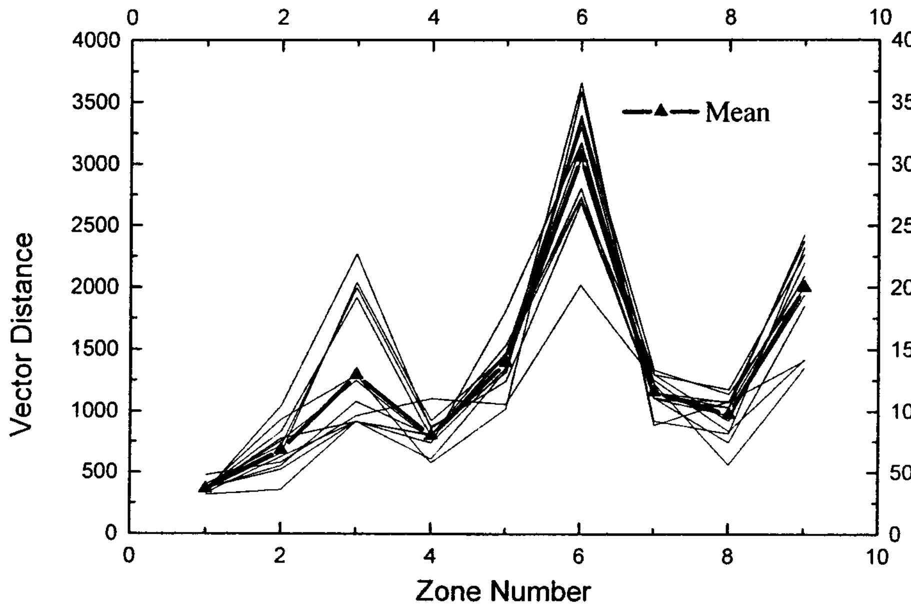


Figure 6.6 (a) Zone number vs. zoned vector distance (Z-VD$_r$) feature graph for Malayalam handwritten character ഞ [-ah]

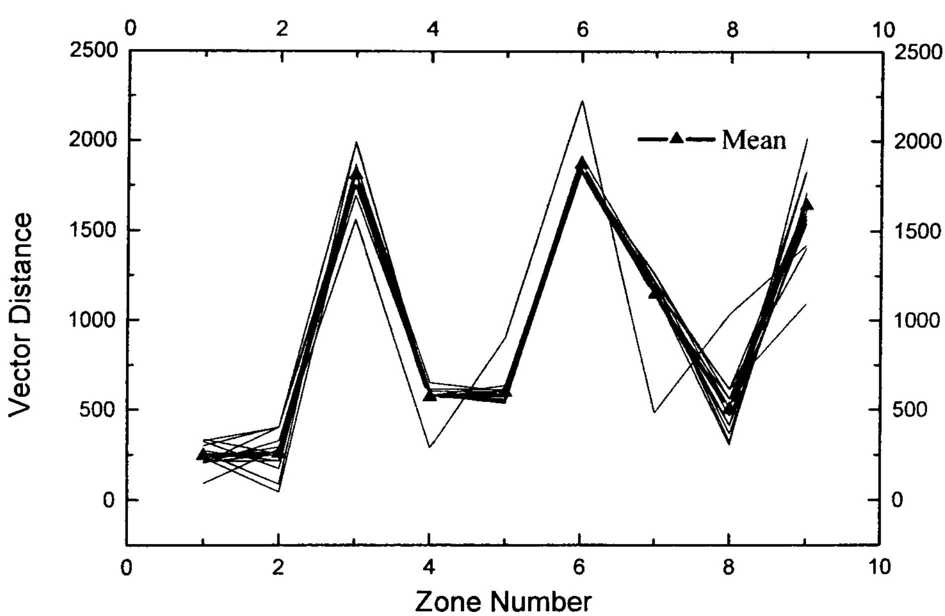


Figure 6.6 (b) Zone number vs. zoned vector distance (Z-VD$_r$) feature graph for Malayalam handwritten character ഗ[ga]

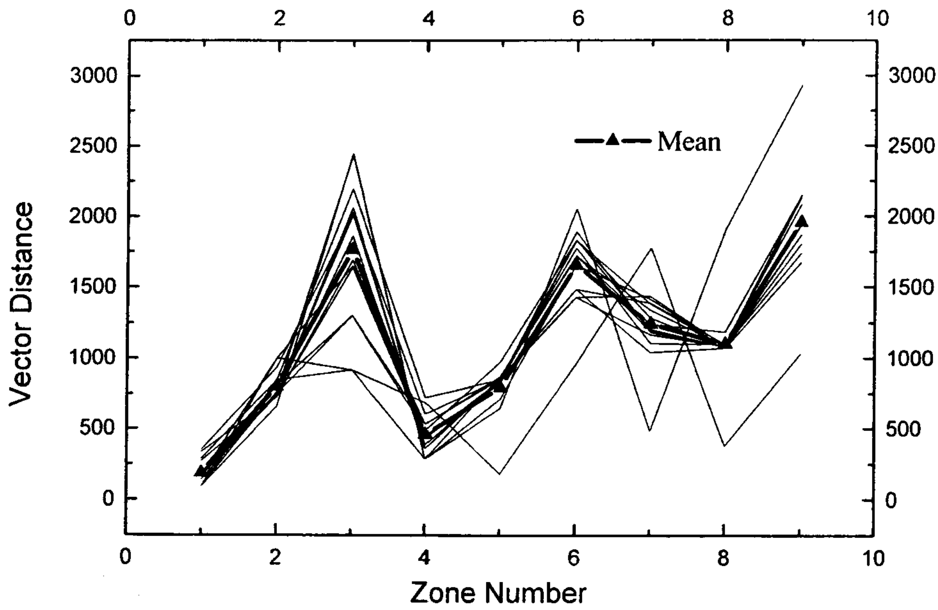


Figure 6.6 (c) Zone number vs. zoned vector distance (Z-VD$_r$) feature graph for Malayalam handwritten character ട[ta]

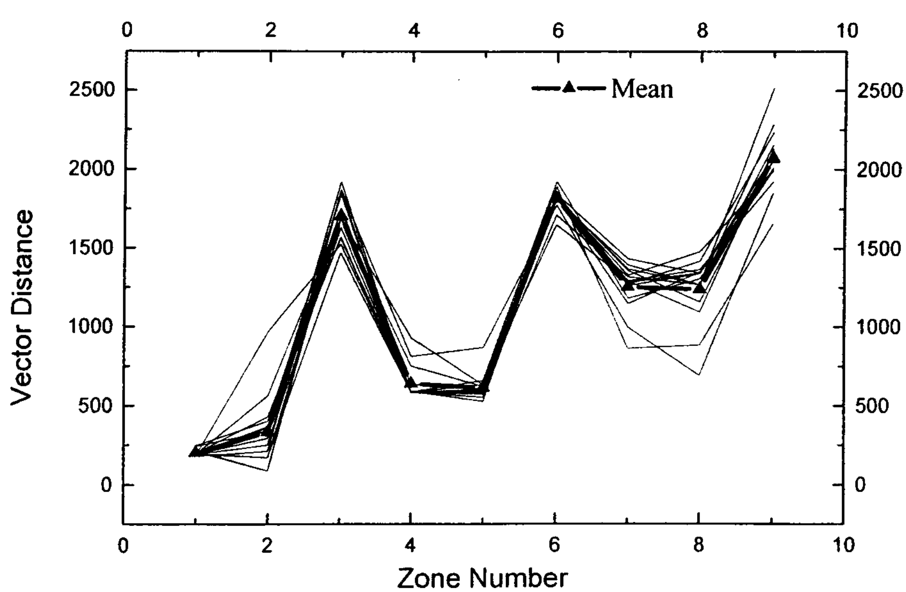


Figure 6.6 (d) Zone number vs. zoned vector distance (Z-VD$_r$) feature graph for Malayalam handwritten character ഥ[thha]

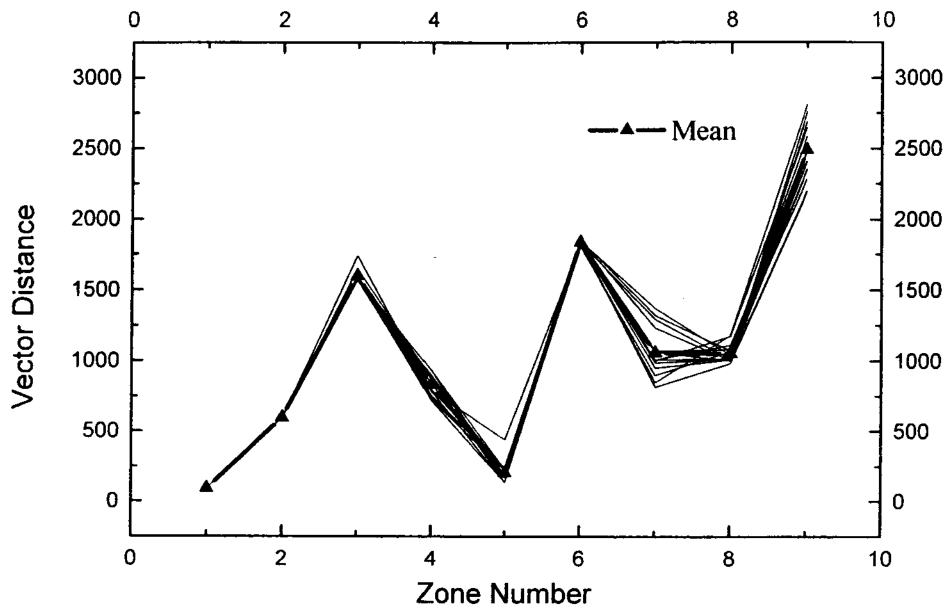


Figure 6.6 (e) Zone number vs. zoned vector distance (Z-VD_r) feature graph for Malayalam handwritten character പ [pa]

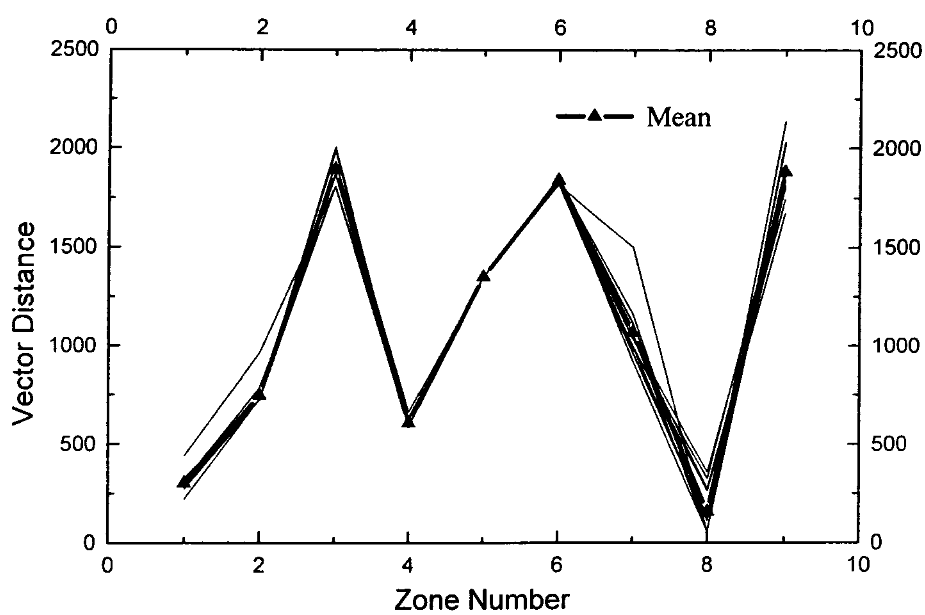


Figure 6.6 (f) Zone number vs. zoned vector distance (Z-VD_r) feature graph for Malayalam handwritten character റ[rha]

Two main limitations can be identified with the above extracted Z-VD features. The first one is due to the varying length of character curves obtained for different samples of the same character located in a particular zone as illustrated in figure 6.7. If the character curve lengths vary considerably for different samples of the same character inside a particular zone then the sum of the vector distances estimated for all the pixels in that region also vary accordingly. This may lead to the irregularity in feature vector extracted from different samples of the same character.

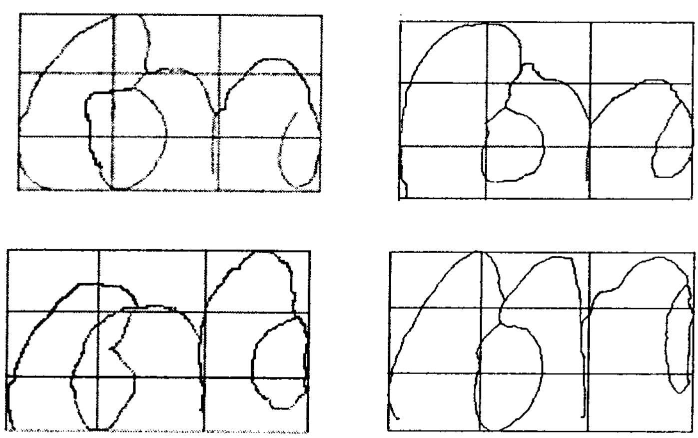


Figure 6.7 Samples of the Malayalam handwritten character ഒ [-ah] with varying length character curves in zones

The second problem is that when character curve is close to the zone borders, small variations in the curve could lead to large variations in the extracted features. Some selected samples of the handwritten Malayalam character ഒ[-ah] with deviations on the character curve close to the zone borders are shown in figure 6.8. The following section deals with the fine-tuning of the above proposed Z-VD parameters in order to surmount these two limitations.

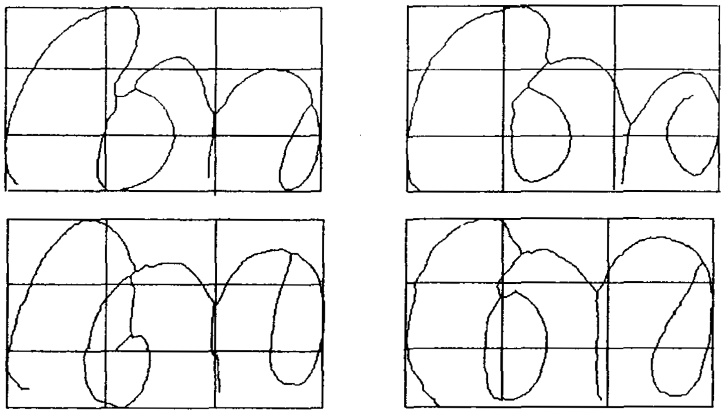


Figure 6.8 Samples of the handwritten Malayalam character ഞ [-ah] with deviations on the character curve close to the zone borders

6.3.2 Fuzzy-Zoned Normalized Vector Distance (FZ-NVD) Features for Handwritten Character Modeling

The difficulties occurred due to the variable length of the character curve for the different samples of the same character in each zone are surmounted by computing the normalized vector distances for each zone. This modification is achieved by dividing the total vector distance computed for each zone with the total number of black pixels constituting the character curve. The second problem identified with the Z-VD features, emerged due to the formation of crisp zone boundaries used for partitioning the character images, is tackled by introducing fuzzy borders for each zone. Here each region boundaries except the outer boundaries are converted into fuzzy borders. This is achieved by dividing each zone borders into 3 fuzzy regions. The nine fuzzy zones (1 to 9) thus obtained are classified as corner regions (1, 3, 7 and 9), peripheral regions (2, 6, 8 and 4) and the central region (5).

Nine fuzzy zones with their fuzzy region boundaries obtained by this region decomposition process are illustrated in figure 6.9 (a-d).

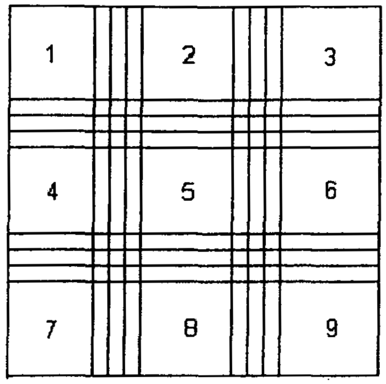


Fig. 6.9 (a) Zoning with fuzzy borders

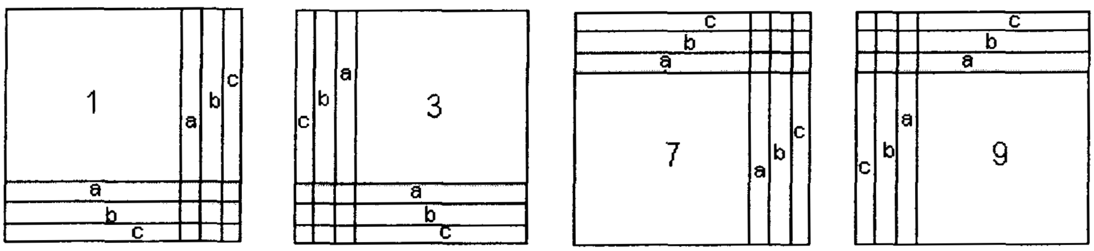


Fig. 6.9 (b) Corner zones

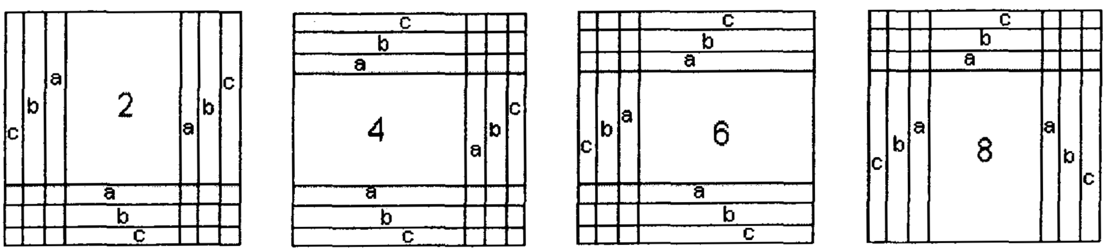


Fig. 6.9 (c) Peripheral zones

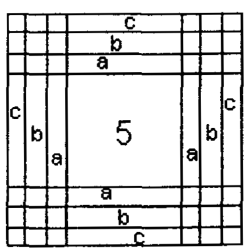


Fig. 6.9(d) Central zone

Figure 6.9(a-d) Nine fuzzy-zones with the region boundaries

The fuzzy regions fixed for each of the zone boundaries are named as region-a, region-b and region-c. Membership values for each pixel as part of the character curve within these fuzzy boundaries are set according to their degree of inclusion (membership grades) in that region. A pixel that is not a part of the fuzzy boundary is assigned a membership value 1. All pixels in the fuzzy region-a are assigned a membership value 0.75, pixels in the fuzzy region-b are assigned a membership value 0.5 and fuzzy region-c gets a membership value 0.25. This fuzzy zone design finds a single pixel as a member of different fuzzy regions with varying membership values.

In the next stage each of the fuzzy-zones are treated separately for the feature extraction process. By considering the bottom left corner of the minimum rectangle enclosing the character curve as the absolute origin, the vector distance for k^{th} pixel in the r^{th} region at location (x, y) is computed as

$$dr_k = (x^2 + y^2)^{1/2}$$

The sum of the vector distances of all the pixels from the origin for each zone is computed separately. After that the normalized vector distance Z-NVD$_r$ is computed, for each zone, by dividing the sum of the vector distances of all the black pixels constituting the character curve that resides in a particular zone with the total number of black pixels in that zone using the expression,

$$\text{Z-NVD}_r = \frac{1}{N_r} \sum_{k=1}^{N_r} \mathrm{d}r_k \quad \text{for } r = 1, 2, \ldots, 9.$$

where N_r is the number of pixels representing the character curve that reside in r^{th} region.

The fuzzification of the normalized vector distance Z-NVD$_r$ is performed on the basis of fuzzy zoning concept. Here the vector distance calculated for each pixel dr_k is multiplied with their membership value in that region. The fuzzified normalized vector distance feature value for each zone is computed by the expression,

$$\text{FZ-NVD}_r = \frac{1}{N_r} \sum_{k=1}^{N_r} \mu r_k \ \mathrm{d}r_k \quad \text{for } r = 1, 2, \ldots, 9.$$

where μr_k is the membership value of the pixel k in the region r and dr_k is the vector distance from the origin for the pixel k in the region r. These nine normalized fuzzy vector distances computed for each of the fuzzy regions constitute the final feature vector FZ-NVD. The proposed fuzzy-zoning based feature extraction method used for modeling Malayalam handwritten characters is summarized in the following algorithm.

Algorithm: Feature extraction based on Fuzzy-Zoned Normalized Vector Distance (FZ-NVD)

Step 1: Crop the handwritten character image sample taken from the database in to minimum rectangle such that the character curve fits exactly to the rectangle.

Step 2: Apply size normalization by mapping the character image to a standard rectangle of size 66x42 pixels and perform binarization and thinning on each character image sample.

Step 3: Divide the resultant image into 3x3 uniform sized zones (1 to 9). Region boundaries except the outer boundary are converted into fuzzy borders.

Step4: Fuzzy membership values 0.25, 0.5 or 0.75 are assigned to each and every pixel present in the three regions of the fuzzy borders according to their degree of inclusion.

Step 5: Compute the vector distance $\mathrm{d}r_k$ for each of the pixels in the character curve included in each zone by considering the bottom left corner as the absolute origin (0, 0). The vector distance for k^{th} pixel with r^{th} zone at location (x, y) is given by the expression $\mathrm{d}r_k = (x^2 + y^2)^{1/2}$

Step 6: The normalized vector distance Z-NVD$_r$ for each zone is computed by dividing the sum of the vector distances of all the black pixels (points on the character curve) with the total number of pixels in that zone using the expression,

$$\text{Z-NVD}_\text{r} = \frac{1}{N_r} \sum_{k=1}^{N_r} \mathrm{d}r_k \quad \text{for } r = 1, 2, \ldots, 9.$$

where N_r is the number of black pixels in r^{th} region

Step 7: Perform fuzzification of the above obtained normalized vector distance FZ-NVD$_r$ by multiplying the vector distance computed for each pixel dr_k by their membership values using the expression,

$$\text{FZ-NVD}_r = \frac{1}{N_r} \sum_{k=1}^{N_r} \mu r_k \, dr_k \quad \text{for } r = 1, 2, \ldots, 9.$$

where μr_k is the membership value of the pixel k in the region r and dr_k is the vector distance from the origin for the pixel k in the region r.

Step 8: The nine normalized fuzzy vector distances computed in step 7 constitute the final feature vector FZ-NVD.

Figure 6.10(a-f) shows the feature vector graph plotted for the FZ-NVD features obtained for the different samples of selected Malayalam characters. The slim lines in the graph represent the feature graph obtained from the repeated samples of the same character written by different writers. The bold line indicates the mean feature curve. From these graphs we see that the distribution of feature values is almost similar for different samples of a particular character and varies from character to character. It is also important that the dimension of the feature vector is nine and it is comparatively very small. So this extracted information can be effectively utilized as a parameter for high speed Malayalam handwriting recognition applications. The recognition experiments based on FZ-NVD features using different pattern classifiers have been conducted in the succeeding chapters.

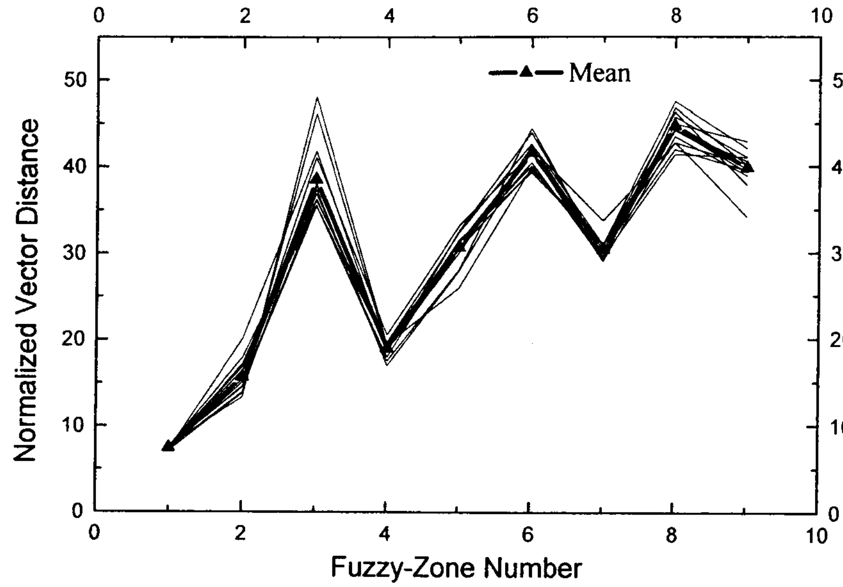


Figure 6.10 (a) Fuzzy-zone number vs. fuzzy-zoned normalized vector distance (FZ-NVD$_r$) feature graph for Malayalam handwritten character ഞ [-ah]

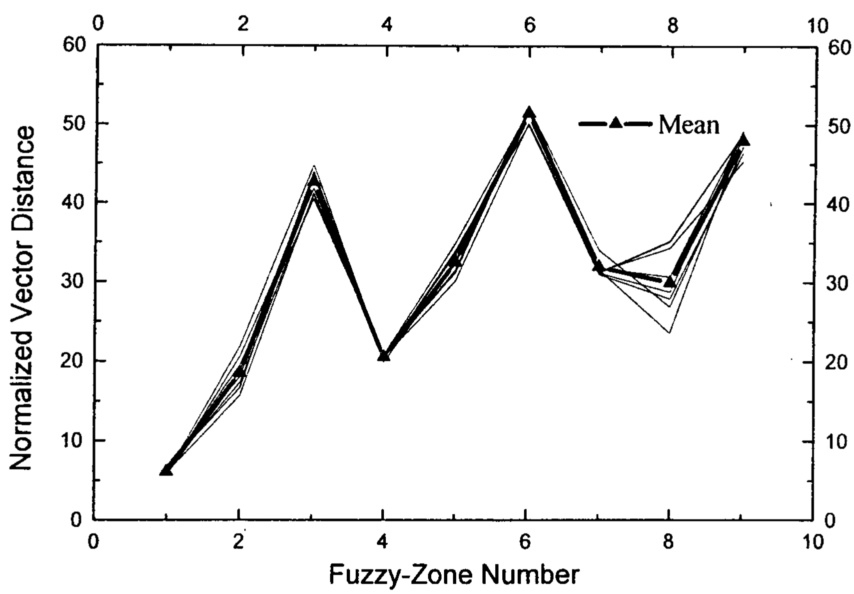


Figure 6.10 (b) Fuzzy-zone number vs. fuzzy-zoned normalized vector distance (FZ-NVD$_r$) feature graph for Malayalam handwritten character ഗ[ga]

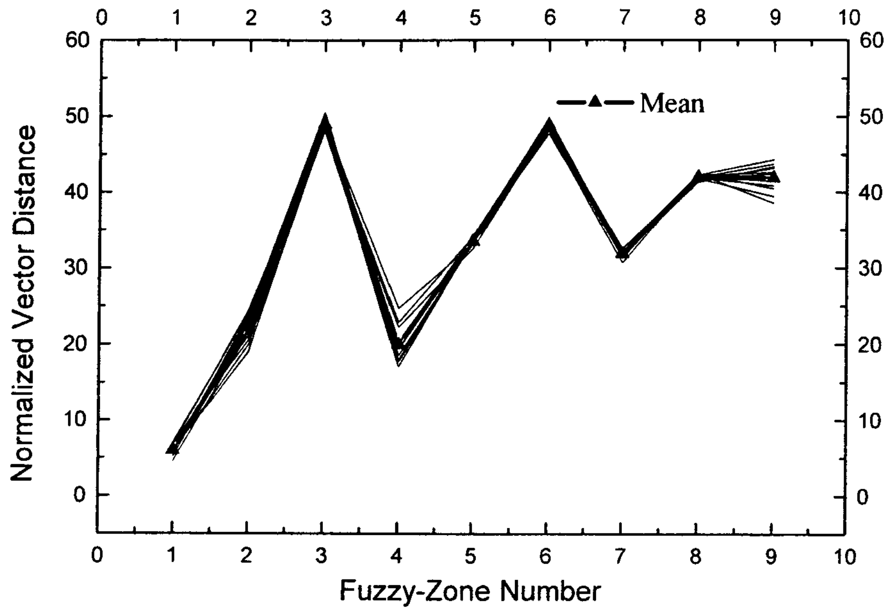


Figure 6.10 (c) Fuzzy-zone number vs. fuzzy-zoned normalized vector distance (FZ-NVD$_r$) feature graph for Malayalam handwritten character ട[ta]

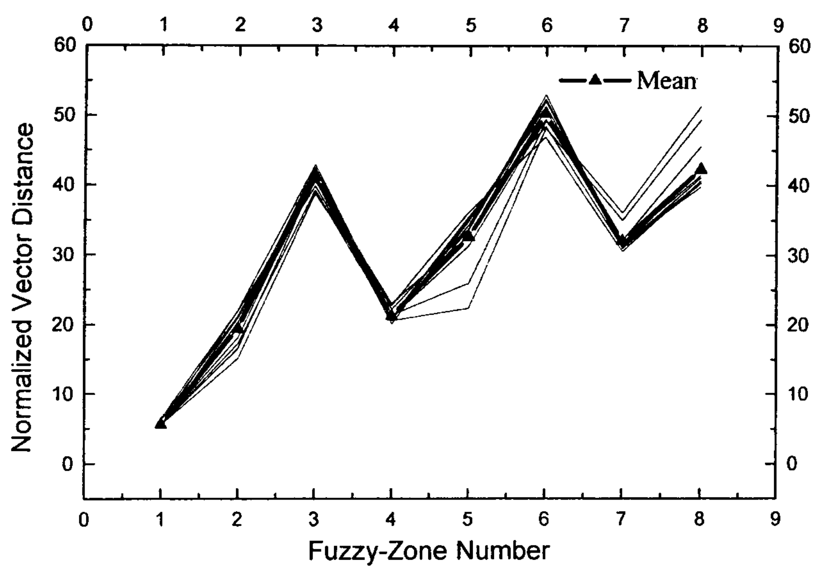


Figure 6.10 (d) Fuzzy-zone number vs. fuzzy-zoned normalized vector distance (FZ-NVD$_r$) feature graph for Malayalam handwritten character ഥ[thha]

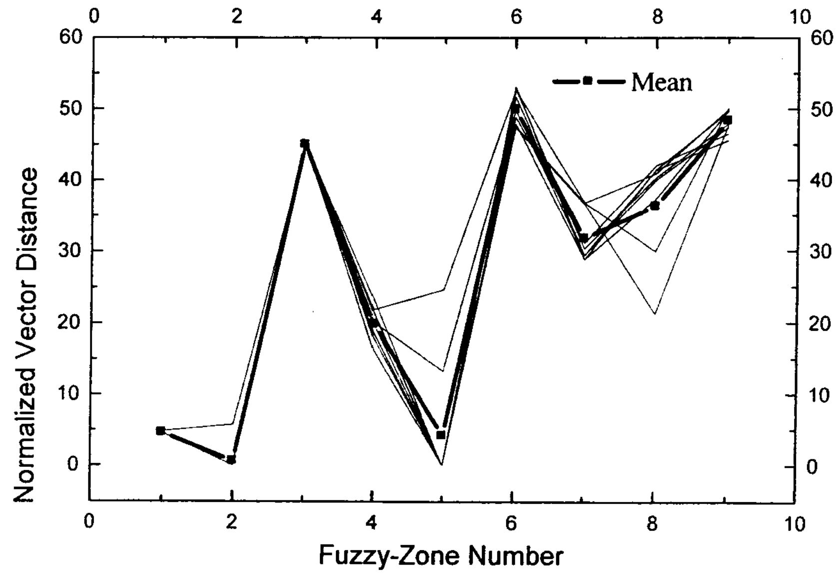


Figure 6.10 (e) Fuzzy-zone number vs. fuzzy-zoned normalized vector distance (FZ-NVD$_r$) feature graph for Malayalam handwritten character പ [pa]

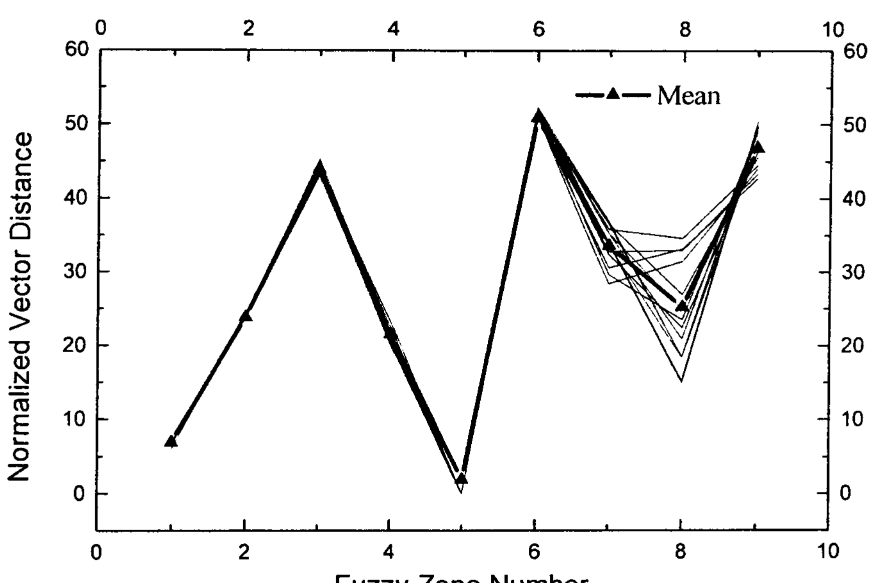


Figure 6.10 (f) Fuzzy-zone number vs. fuzzy-zoned normalized vector distance feature (FZ-NVD$_r$) graph for Malayalam handwritten character റ[rha]

6.4 Conclusions

The global characteristics of Malayalam handwritten character patterns are analyzed based on the aspect ratio, existence ratio and the centroid of the character curve. The experimental results indicate that it is extremely difficult to classify the character patterns based on the global features. As a solution, an efficient algorithm for identifying the characteristics of character patterns based on their parts is introduced. Consequently a novel method for handwritten character modeling using perceptual zoning of the pre-processed character images using the vector distance as the parameter measure is proposed and evaluated. These features are then fine tuned with the fuzzy-zoning and normalized vector distance measures. It is found that the distribution patterns obtained from these features are almost similar for repeated samples of the same character written by different writers and varies from character to character. So these parameters can be effectively employed to model handwritten Malayalam characters and hence can be further utilized for recognition.

This feature extraction technique is computationally simple and easy to implement. The added advantage is that since the feature vector dimension is very small it can be used for high-speed recognition applications.

CHAPTER 7

HANDWRITTEN PATTERN RECOGNITION USING CLUSTER ANALYSIS, *k*-NN CLASSIFIER AND CLASS-MODULAR NEURAL NETWORK

7.1 Introduction

The best pattern recognizers in most instances are human beings. Yet we do not completely understand how they recognize patterns. Pattern recognition is the study of how machines can observe the environment, learn to distinguish pattern of interest from their background, and make sound and reasonable decisions about the categories of the patterns. Automatic (machine) recognition, description, classification and grouping of patterns are important problems in a variety of engineering and scientific disciplines. Pattern recognition can be viewed as the categorization of input data into identifiable classes via the extraction of significant features or attributes of the data from the background of irrelevant details. Duda and Hart [Duda.R.O and Hart.P.E, 1973] define it as a field concerned with machine recognition of meaningful regularities in noisy or complex environment. It encompasses a wide range of information processing problems of great practical significance from handwritten character recognition and speech recognition, to fault detection in machinery and medical diagnosis. Today, pattern recognition is an integral part of most intelligent systems built for decision making.

Normally the pattern recognition processes make use of one of the following two classification strategies.

i. Supervised classification (e.g., discriminant analysis) in which the input pattern is identified as a member of a predefined class.

ii. Unsupervised classification (e.g., clustering) in which the pattern is assigned to a hitherto unknown class.

In the present study the well-known approaches that are widely used to solve pattern recognition problems including clustering technique (*c*-Means algorithm), Statistical pattern classifier (*k*-Nearest Neighbour classifier), and connectionist approach (Class-Modular Artificial Neural Networks) are used for recognizing Malayalam handwritten characters. Here *c*-Means clustering technique is based on unsupervised learning approach. The *k*-NN classifier and class-modular neural network work on the basis of supervised learning strategy.

The state-space point distribution (SSPD) features extracted from the gray-scale images of the handwritten characters as explained in chapter 5, zoned vector distance (Z-VD) features and perceptual fuzzy-zoned normalized vector distance (FZ-NVD) features estimated from the pre-processed character images as discussed in chapter 6 are used as parameters for recognition study. The recognition experiments are conducted using the different pattern recognition algorithms in order to identify the credibility of the proposed parameters. This chapter is organized in three sections. First section presents the character recognition experiments conducted using cluster analysis. The second section deals with recognition experiments conducted using *k*-NN statistical classifier. The third section describes the class-modular neural network architecture and the simulation experiments conducted for the recognition of Malayalam handwritten characters along with the performance comparisons of various classifies and proposed parameters.

7.2 Cluster Analysis for Pattern Recognition

In real life applications, we handle a huge amount of information that are perceived. Here processing every piece of information as a single entity would be impossible. Thus we tend to categorize entities into clusters, which are characterized by common attributes of the entities it contains and hence the huge amount of information contained in any relevant process is reduced. Some of the common definitions proposed for a cluster are given below.

1. A cluster is a set of entities that are alike, and entities from different clusters are not alike.
2. A cluster is an aggregation of points in the test space such that the distance between any two points in the cluster is less than the distance between any point in the cluster and any point not in it.
3. Clusters may be described as connected regions of a *p*-dimensional space containing a relatively high density of points, separated from other such regions by a region containing a relatively low density of points.

Clustering is a major tool used in pattern recognition processes generally for data reduction, hypothesis generation, hypothesis testing and prediction based on grouping. In several cases, the amount of data available in a problem can be very large and, as a consequence, its effective processing becomes very demanding. In this context data reduction by the help of cluster analysis can be used in order to group data into a number of reduced representative clusters. Then each cluster can be processed as a single entity. In some other applications cluster analysis can be used to infer some hypothesis concerning the nature of the data. These hypotheses must then be verified using other data sets and in this context, cluster analysis is used for the verification of the validity of a specific hypothesis.

Another important application of cluster analysis is the prediction based on grouping. In this case, cluster analysis is applied to the available data set, and the resulting clusters are characterized based on the characteristics of the patterns by which they are formed. Consequently, if we are given an unknown pattern, we can determine the cluster to which it is more likely to belong and we characterize it based on the characteristics of the respective cluster. In the present study, we are interested in applying cluster analysis for prediction based on the grouping using *c*-Means clustering technique for the recognition of handwritten character patterns. The implementation details and experimental results using these techniques are explained in the following section.

7.2.1 *c*-Means Clustering for Handwritten Character Recognition

The *c*-Means algorithm is one of the most simple and well-known clustering techniques. This has been applied to variety of pattern recognition problems. It is based on the minimization of an objective function, which is defined as the sum of the squared distances from all points in a cluster domain to the cluster centre. Determining the prototypes or cluster centers is a major task in designing a classifier based on clustering. This is normally achieved on the basis of minimum distance approach. Prior to designing pattern-clustering algorithms we must define a similarity measure by which we decide whether or not two patterns $\boldsymbol{x}$ and $\mathbf{y}$ are members of the same cluster. A similarity measure $\delta(\boldsymbol{x}, \boldsymbol{y})$ is usually defined, so that the principle $\lim \delta(\boldsymbol{x}, \boldsymbol{y}) = 0$ as $\boldsymbol{x} \rightarrow \boldsymbol{y}$ hold. This is the case for example, if the patterns are in R^n and we define

$$\delta(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{x} - \boldsymbol{y}\|^2$$

The detailed procedure and implementation details of the *c*-Means clustering technique used for classifying Malayalam handwritten characters are explained below.

The *c*-Means algorithm partitions a collection of n vectors $\boldsymbol{x}_j$, where $j = 1,\ldots\ldots, n$ into m groups G_i, $i = 1,\ldots\ldots, m$ and finds cluster centers $\boldsymbol{c}_i$, $i = 1,\ldots, m$ corresponding to each group such that a cost function of dissimilarity (or distance) measure is minimized.

A generic distance function $d(\boldsymbol{x}_k, \boldsymbol{c}_i)$ can be applied for vector $\boldsymbol{x}_k$ in group i; the corresponding cost function is thus expressed as

$$J = \sum_{i=1}^{m} J_i = \sum_{i=1}^{m} \left(\sum_{k, x_k \in G_i} d(x_k - c_i) \right) \quad \text{.......................... 7.1}$$

In this work the Euclidean distance is chosen as the dissimilarity measure between a vector $\boldsymbol{x}_k$ in group G_i and the corresponding cluster centre $\boldsymbol{c}_i$. Here the cost function is defined by

$$J = \sum_{i=1}^{m} J_i = \sum_{i=1}^{m} \left(\sum_{k, x_k \in G_i} \| x_k - c_i \|^2 \right) \quad \text{........................7.2}$$

where $J_i = \sum_{k, x_k \in G_i} \| x_k - c_i \|^2$ is the cost function within group i. The value of J_i depends on the geometrical properties of G_i and the location of $\boldsymbol{c}_i$.

The collection of partitioned groups can be defined by a m x n binary membership matrix $\boldsymbol{U}$, where the element u_{ij} is 1 if the j^{th} data point $\boldsymbol{x}_j$ belongs to group i, and 0 otherwise. Once the cluster centers $\boldsymbol{c}_i$ are fixed, the value of u_{ij} can be computed using the expression,

$$u_{ij} = \begin{cases} 1 \text{ if } \|\boldsymbol{x}_j - \boldsymbol{c}_i\|^2 \leq \|\boldsymbol{x}_j - \boldsymbol{c}_k\|^2, \text{ for each } k \neq i, \\ 0 \text{ otherwise} \end{cases} \quad \text{........7.3}$$

where $i = 1, \ldots\ldots, m$

$j = 1, \ldots\ldots, n$ and $k \leq m$

That is, x_j belongs to group i if c_i is the closest centre among all centers. Since a given data point can only be in a group, the membership matrix U has the following properties:

$$\sum_{i=1}^{m} u_{ij} = 1, \quad \forall j = 1, \ldots, n \quad \text{and}$$

$$\sum_{i=1}^{m} \sum_{j=1}^{n} u_{ij} = n$$

If u_{ij} is fixed, then the optimal cluster centre c_i that minimize the cost function in equation (7.1) is computed by finding the mean of all vectors in group i given by the expression,

$$c_i = \frac{1}{|G_i|} \sum_{k, x_k \in G_i} x_k \quad \ldots\ldots\ldots\ldots\ldots\ldots 7.4$$

where $|G_i|$ is the size of G_i, or

$$|G_i| = \sum_{j=1}^{n} u_{ij}$$

For a batch–mode operation, the c-Means algorithm is presented with a data set x_i, $i = 1, \ldots, n$; the algorithm determines the cluster centers c_i, and the membership matrix U iteratively using the following steps.

Step 1: Initialize the cluster centers c_i, $i = 1, \ldots, m$. This is typically achieved by randomly selecting m points from among all of the data points.

Step 2: Determine the membership matrix U using the equation (7.3).

Step 3: Compute the cost function according to equation (7.2). Stop if either it is below a certain tolerance value or its improvement over previous iteration is below a certain threshold.

Step 4: Update the cluster centers according to equation (7.4). Go to step 2.

Finally in the recognition stage an unknown pattern $\boldsymbol{x}$ is compared with each final cluster centers obtained by applying the above procedure. The cluster l with minimum distance from the unknown pattern $\boldsymbol{x}$ is found out by the given expression,

$$\boldsymbol{x} \in l \text{ if } \| \boldsymbol{x} - \boldsymbol{c}_l \|^2 < \| \boldsymbol{x} - \boldsymbol{c}_i \|^2$$

$$\text{for all } i = 1,2\ldots, m\ , i \neq l$$

The following section describes the simulation of the above algorithm along with the recognition results obtained for handwritten character patterns.

7.2.2 Simulation Experiments and Results

The recognition experiment is conducted by simulating the above algorithm using MATLAB. The State-Space Point Distribution (SSPD) parameters extracted from the gray-scale character images as discussed in chapter 5, Z-VD features and FZ-NVD features extracted as explained in chapter 6 are used for recognition purpose. Here we used a set of 15,752 samples of the forty four Malayalam handwritten characters collected from 358 writers for iteratively computing the final cluster centers and a disjoint set of character patterns of same size from the database for recognition purpose.

The recognition accuracies of forty four Malayalam handwritten characters based on the above said three features using *c*-Means clustering technique are given in Table 7.1. The graphical representation of these recognition results based on different features using *c*-Means clustering technique is shown in figure 7.1(a-d).

The recognition results indicate the credibility of the extracted features on the basis of clusters that can be formed with the help of an unsupervised learning process. The forty four cluster centers formed from the training set show that the extracted features are good enough to distinguish the character patterns from one another. The overall recognition accuracies obtained for the forty four Malayalam characters using *c*-Means clustering technique based on SSPD, Z-VD and FZ-NVD features are 64.148%, 67.426% and 69.001% respectively.

The alternative classifier used in this study is the well-known non-parametric *k*-Nearest Neighbour statistical classifier. The following section describes the recognition experiments performed using the above said features and *k*-NN classifier.

Character Number	Character	Recognition Accuracy (%)		
		SSPD Feature	Z-VD Feature	FZ-NVD Feature
1	അ [-ah]	76.53	86.87	89.38
2	ആ[-aah]	59.49	60.05	64.25
3	ഇ [-yi]	84.35	77.09	79.89
4	ഉ [-uh]	68.99	68.44	71.51
5	ഋ [-eru]	61.45	64.80	65.64
6	എ [-eh]	82.12	84.63	84.07
7	ഏ[-aeh]	79.32	65.64	70.95
8	ഒ [-oh]	64.53	79.33	79.89
9	ക [ka]	80.16	74.58	75.41
10	ഖ [kha]	63.69	67.04	72.62
11	ഗ [ga]	70.11	84.92	85.47
12	ഘ [gha]	65.36	67.59	71.78
13	ങ [nga]	69.83	74.46	79.61
14	ച [cha]	58.66	65.36	67.79
15	ഛ[chha]	69.55	69.83	69.83
16	ജ [ja]	61.45	59.78	62.57
17	ഝ[jha]	76.25	76.53	79.33
18	ഞ [nja]	75.69	80.17	79.61
19	ട [ta]	58.49	68.16	59.49
20	ഠ[tta]	59.78	59.49	62.45
21	ഡ [da]	78.49	79.89	80.45
22	ഢ [dda]	64.80	56.70	59.78

(Continued)

Table 7.1 Recognition accuracies of forty four basic Malayalam handwritten characters based on SSPD, Z-VD, FZ-NVD features using *c*-Means clustering technique

Character Number	Character	Recognition Accuracy (%)		
		SSPD Feature	Z-VD Feature	FZ-NVD Feature
23	ണ[nha]	55.02	67.32	72.91
24	ത [tha]	52.51	55.86	58.37
25	ഥ [thha]	72.62	83.24	84.63
26	ദ [dha]	65.36	66.48	70.95
27	ധ[dhha]	81.56	75.41	78.49
28	ന [na]	62.01	64.25	69.55
29	പ [pa]	54.47	58.66	69.49
30	ഫ [pha]	79.05	54.75	57.82
31	ബ [ba]	54.45	59.78	65.08
32	ഭ [bha]	51.13	82.12	83.24
33	മ [ma]	50.28	54.47	55.59
34	യ [ya]	43.57	48.60	51.67
35	ര [ra]	78.49	65.92	65.92
36	ല [la]	41.89	68.60	53.35
37	വ [va]	63.12	82.68	64.34
38	ശ [sha]	62.84	51.39	55.86
39	ഷ[shha]	53.35	54.47	58.10
40	സ [sa]	45.58	67.88	67.60
41	ഹ [ha]	51.11	69.83	71.79
42	ള [lha]	50.28	53.35	55.59
43	ഴ [zha]	55.03	57.82	59.22
44	റ [rha]	69.71	52.51	54.75
Average Recognition accuracy (%)		64.14818	67.42591	69.00182

Table 7.1 Recognition accuracies of forty four basic Malayalam handwritten characters based on SSPD, Z-VD, FZ-NVD features using *c*-Means clustering technique

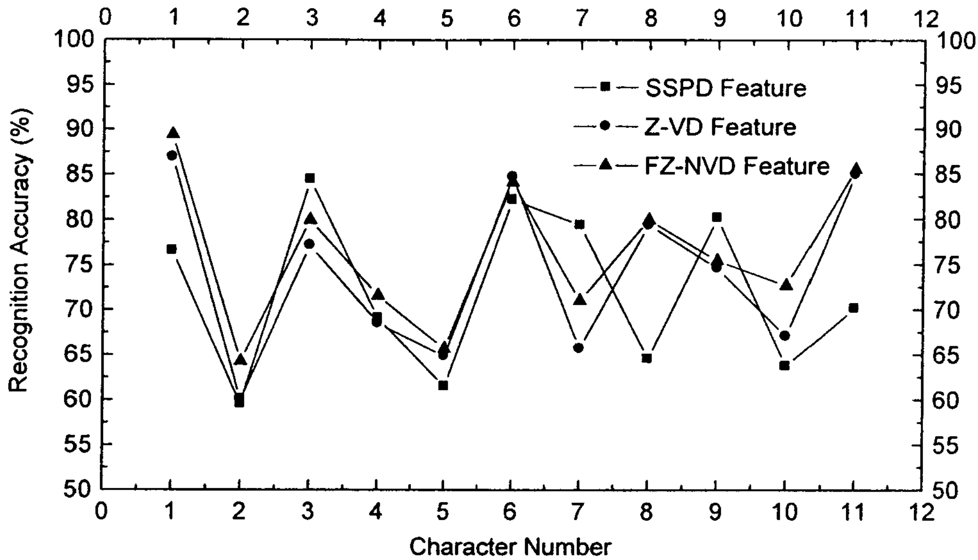


Figure 7.1 (a)

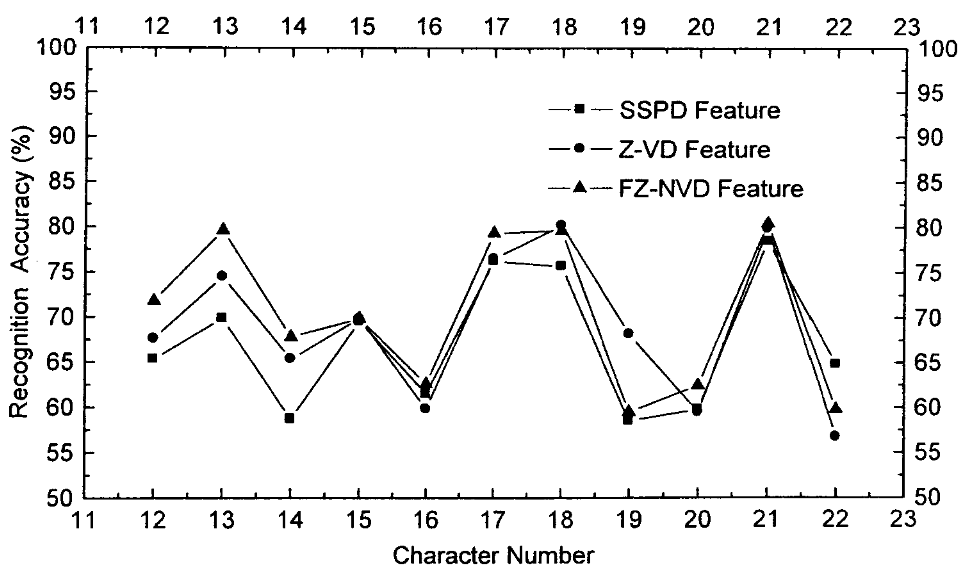


Figure 7.1(b) (Continued)

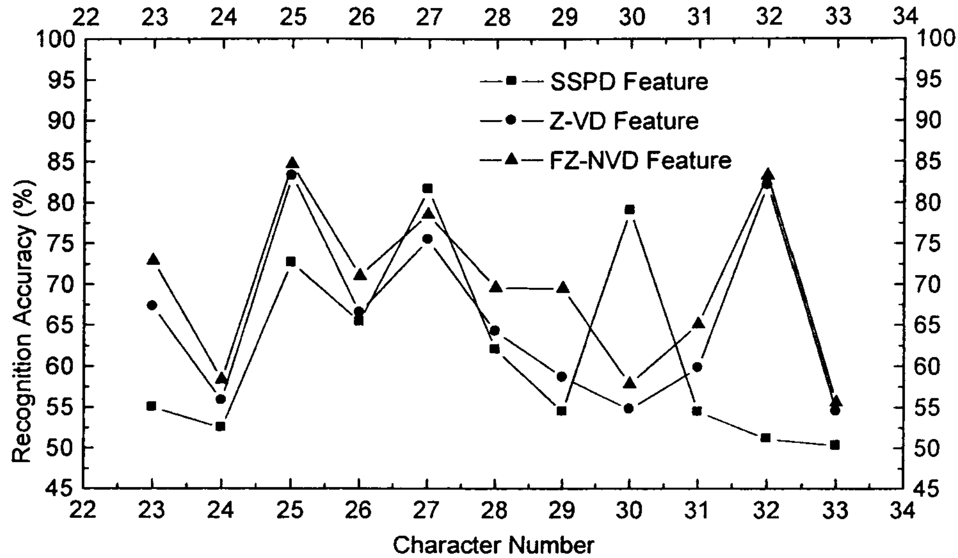


Figure 7.1(c)

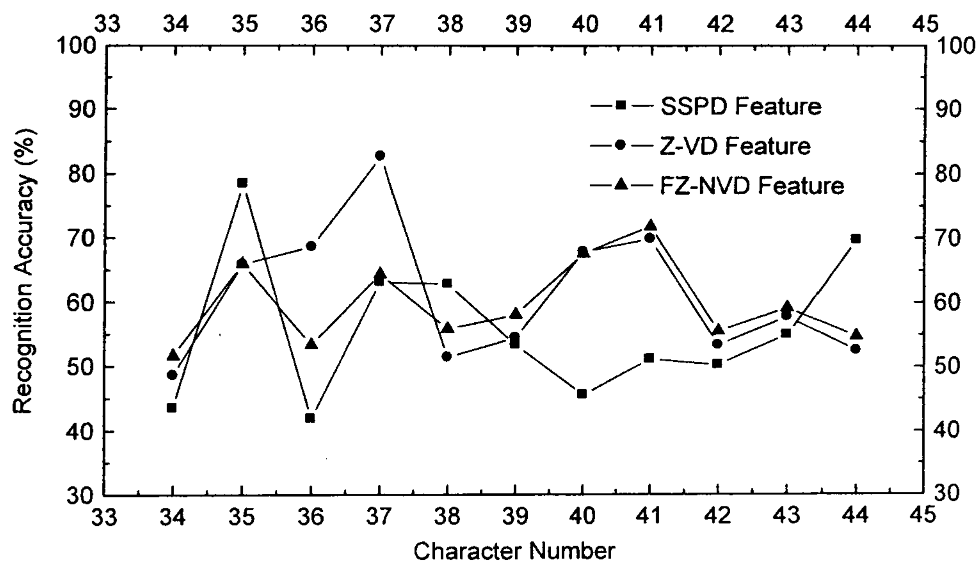


Figure 7.1(d)

Figure 7.1(a-d) Recognition accuracies of forty four Malayalam handwritten characters using SSPD, Z-VD and FZ-NVD features using *c*-Means clustering

7.3 Statistical Pattern Classification

In the statistical pattern classification process, each pattern is represented by a d dimensional feature vector and it is viewed as a point in the d-dimensional space. Given a set of training patterns from each class, the objective is to establish decision boundaries in the feature space which separate patterns belonging to different classes. The recognition system is operated in two phases, training (learning) and classification (testing). The following section describes the pattern recognition experiment conducted for the recognition of forty four basic Malayalam handwritten characters using k-NN classifier.

7.3.1 *k*-Nearest Neighbour Classifier for Handwritten Pattern Recognition

Pattern classification by distance functions is one of the earliest concepts in pattern recognition [Tou.J.T and Gonzalez.R.C, 1974], [Friedman.M. and Kandel.A, 1999]. Here the proximity of an unknown pattern to a class serves as a measure of its classification. A class can be characterized by single or multiple prototype pattern(s). The k-Nearest Neighbour method is a well-known non-parametric classifier, where a posteriori probability is estimated from the frequency of nearest neighbours of the unknown pattern. It considers multiple prototypes while making a decision and uses a piecewise linear discriminant function. Various pattern recognition studies with first-rate performance accuracy are also reported based on this classification technique [Ray A.K. and Chatterjee B, 1984], [Zhang.B and Srihari.S.N, 2004], [Pernkopf.F, 2005].

Consider the case of m classes c_i, i =1,....., m and a set of N samples patterns y_i, i =1,..., N whose classification is *a priory* known. Let $\boldsymbol{x}$ denote an arbitrary incoming pattern. The nearest neighbour classification approach classifies $\boldsymbol{x}$ in the pattern class of its nearest neighbour in the set y_i, i = 1,....., N i.e.,

$$\text{If } \|\boldsymbol{x}-\boldsymbol{y}_j\|^2 = \min \|\boldsymbol{x}-\boldsymbol{y}_i\|^2 \quad \text{where } 1 \le i \le N$$

$$\text{then } \boldsymbol{x} \in c_j.$$

This scheme can be termed as 1-NN rule since it employs only one nearest neighbour to $\boldsymbol{x}$ for classification. This can be extended by considering the k nearest neighbours to $\boldsymbol{x}$ and using a majority-rule type classifier. The following algorithm summarizes the classification process.

Algorithm: Minimum distance *k*-Nearest Neighbour classifier

Input: N – number of pre-classified patterns

m – number of pattern classes.

(y_i, c_i), $1 \le i \le N$ - N ordered pairs, where y_i is the i^{th} pre-classified pattern and c_i it's class number ($1 \le c_i \le m$ for all i).

k - order of NN classifier (i.e. the k closest neighbours to the incoming patterns are considered).

$\boldsymbol{x}$ - an incoming pattern.

Output: L – class number into which $\boldsymbol{x}$ is classified.

Step 1: Set $S = \{ (y_i, c_i) \}$, where $i = 1,\ldots, N$

Step 2: Find $(y_j, c_j) \in S$ which satisfies

$$\| \boldsymbol{x} - \boldsymbol{y}_j \|^2 = \min \| \boldsymbol{x} - \boldsymbol{y}_i \|^2 \quad \text{where } 1 \le i \le m$$

Step 3: If $k = 1$ set $L = c_j$ and stop; else initialize an

m -dimensional vector I

$I(i') = 0,\ i' \neq c_j;\ I(c_j) = 1$ where $1 \le i' \le m$ and

set $S = S - \{ (y_j, c_j) \}$

Step 4: For $i_0 = 1,\ldots, k\text{-}1$ do steps 5-6

Step 5: Find $(y_j, c_j) \in S$ such that

$$\| \boldsymbol{x} - \boldsymbol{y}_j \|^2 = \min \| \boldsymbol{x} - \boldsymbol{y}_i \|^2 \quad \text{where } 1 \le i \le N$$

Step 6: Set $I(c_j) = I(c_j) + 1$ and $S = S - \{ (y_j, c_j) \}$.

Step 7: Set $L = \max \{ I(i') \}$, $1 \le i' \le m$ and stop.

In the case of k-Nearest Neighbour classifier, we compute the distance of similarity between the features of a test sample and the features of every training sample. The class of the majority among the k - nearest training samples is deemed as the class of the test sample.

7.3.2 Simulation Experiments and Results

The recognition experiment is conducted by simulating the above algorithm using MATLAB. The State-Space Point Distribution (SSPD) of the gray-scale character images extracted as discussed in Chapter 5, Z-VD features and FZ-NVD feature extracted as explained in chapter 6 are used in the recognition study. Here also we used the same set of 15,752 samples of the forty four Malayalam handwritten characters collected from 358 writers which are employed in the previous experiment for training and a disjoint set of character patterns of same size from the database for recognition purpose. The recognition accuracies obtained for the forty four basic Malayalam handwritten characters using the above said features using *k*-NN classifier are tabulated in Table 7.2. The graphical representation of these recognition results based on different features using k-NN classifier is shown in figure 7.2 (a-d)

The overall recognition accuracies obtained for the forty four Malayalam characters using k-NN classifier and SSPD, ZVD_r and F -$ZNVD_r$ features are 71.606%, 73.102% and 75.819% respectively. The recognition results are found better than the previous experiment conducted using *c*-Means clustering technique. These two algorithms do not fully accommodate the small variations in the extracted features. These results specify the need of improving the classification algorithm for large class pattern classification problems. In the next section we present a recognition study conducted using class modular neural network that is capable of adaptively accommodating the minor variations in the extracted features.

Character Number	Character	Recognition Accuracy (%)		
		SSPD Feature	Z-VD Feature	FZ-NVD Feature
1	അ [-ah]	86.59	96.92	97.77
2	ആ[-aah]	60.61	64.25	65.92
3	ഇ [-yi]	96.64	85.47	87.16
4	ഉ [-uh]	73.46	76.26	77.37
5	ഋ [-eru]	65.36	66.20	69.83
6	എ [-eh]	86.31	87.43	89.94
7	ഏ[-aeh]	80.45	72.90	84.64
8	ഒ [-oh]	66.20	82.40	85.75
9	ക [ka]	93.02	75.41	79.61
10	ഖ [kha]	70.95	73.74	78.49
11	ഗ [ga]	72.62	94.97	94.41
12	ഘ [gha]	71.23	71.50	75.41
13	ങ [nga]	77.09	77.09	77.65
14	ച [cha]	64.25	79.33	82.40
15	ഛ[chha]	71.23	75.41	77.93
16	ജ [ja]	70.67	74.02	75.97
17	ഝ[jha]	79.32	81.00	80.16
18	ഞ [nja]	80.45	82.12	80.44
19	ട [ta]	61.45	66.76	61.45
20	ഠ[tta]	59.50	67.32	66.76
21	ഡ [da]	91.06	92.17	93.78
22	ഢ [dda]	73.12	67.88	67.88

(Continued)

Table 7.2 Recognition accuracies of forty four basic Malayalam handwritten characters based on SSPD, Z-VD, FZ-NVD features using *k*-NN classifier

Character Number	Character	Recognition Accuracy (%)		
		SSPD Feature	Z-VD Feature	FZ-NVD Feature
23	ണ[nha]	65.64	73.00	84.63
24	ത [tha]	64.80	64.53	70.39
25	ഥ [thha]	89.38	85.47	86.59
26	ദ [dha]	76.54	76.54	79.05
27	ധ[dhha]	85.20	89.38	89.38
28	ന [na]	74.30	74.30	81.84
29	പ [pa]	67.04	67.04	68.16
30	ഫ [pha]	85.47	65.36	67.04
31	ബ [ba]	64.25	64.53	70.67
32	ഭ [bha]	66.20	85.75	85.75
33	മ [ma]	63.40	63.41	64.53
34	യ [ya]	58.10	58.66	67.04
35	ര [ra]	86.87	67.04	71.23
36	ല [la]	55.31	54.47	60.01
37	വ [va]	67.40	86.03	89.66
38	ശ [sha]	65.08	58.94	63.41
39	ഷ[shha]	59.36	59.50	62.20
40	സ [sa]	58.38	65.36	67.32
41	ഹ [ha]	57.82	74.30	76.54
42	ള [lha]	54.47	55.03	58.66
43	ഴ [zha]	59.22	59.78	62.29
44	റ [rha]	74.86	57.51	58.94
Average Recognition accuracy (%)		71.60614	73.10182	75.81932

Table 7.2 Recognition accuracies of forty four basic Malayalam handwritten characters based on SSPD, Z-VD, FZ-NVD features using *k*-NN classifier

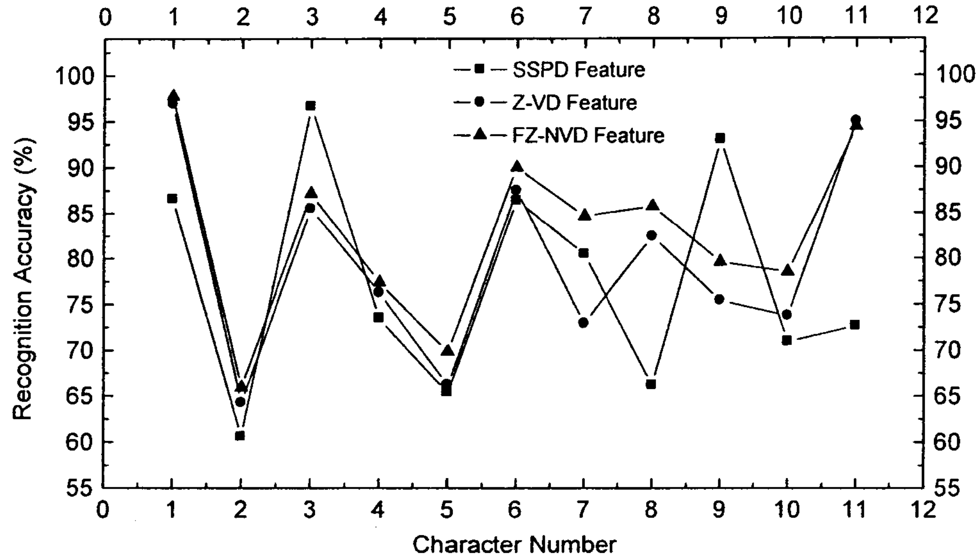


Figure 7.2(a)

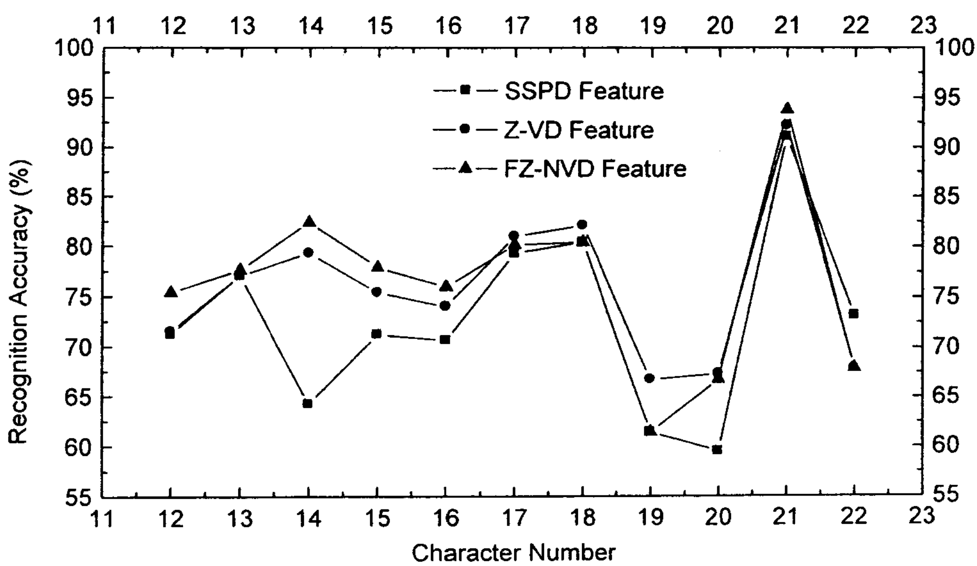


Figure 7.2(b) (Continued)

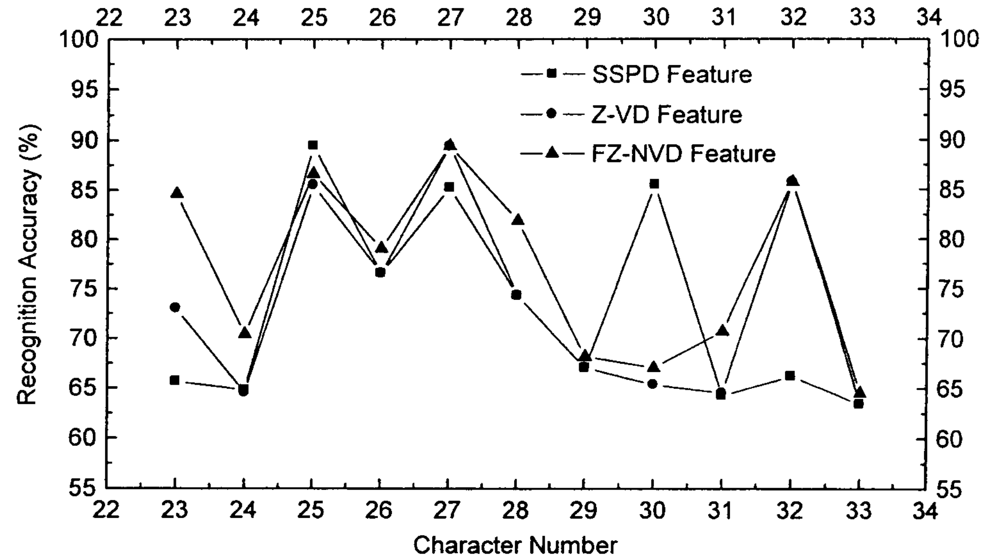


Figure 7.2 (c)

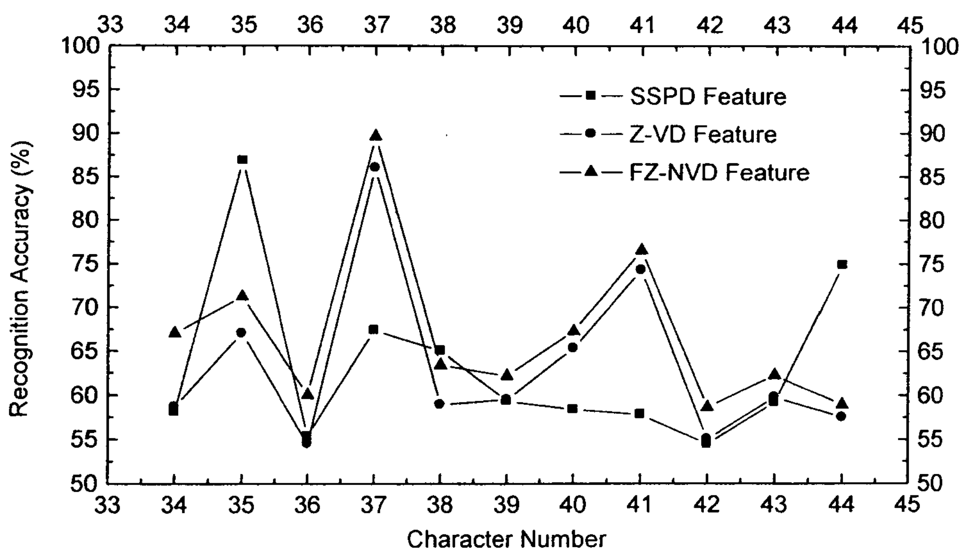


Figure 7.2(d)

Figure 7.2(a-d) Recognition accuracies of forty four Malayalam handwritten characters using SSPD, Z-VD and FZ-NVD features using *k*-NN classifier

7.4 Class-modular Neural Network for Handwritten Character Recognition

In recent years, neural networks have been successfully applied in many of the pattern recognition and machine learning systems [Ripley.B.D, 1996], [Haykin.S, 2004], [Simpson.P.K, 1990]. These models are composed of a highly interconnected mesh of nonlinear computing elements, whose structure is drawn from analogies with biological neural systems. Since the advent of Feed Forward Multi Layer Perception (FFMLP) and error-backpropagation training algorithm great improvements in terms of recognition performance and automatic training have been achieved in the area of character recognition [Looney.C.G, 1997]. Specifically they have been used effectively in recognition of handwritten numerals and letters which comprise a large variation in writing styles [Suen.C.Y *et al.*, 1992], [Srikantan.G *et al.*, 1996], [Cho.S.B,1997], [Oh.I.-S. and Suen.C.Y, 1998]. However, there is an impractical side in directly using these architecture for recognition purpose due to the large number of classes, especially in the recognition of unconstrained handwritten characters of languages with large character sets like Chinese, Korean and most of the Indian languages including Malayalam [Song.H.-H and Lee.S.-W., 1998], [Mui.L *et al.*, 1994]. To overcome this limitation, in the present study, we used a class-modular architecture suitable for the classification module using FFMLP for the recognition of forty four Malayalam handwritten characters [Oh.I.-S. and

Suen.C.Y, 2002]. The following sections deal with the recognition experiments conducted based on the class-modular feed-forward neural network for handwritten Malayalam characters. A brief description about the diverse use of neural networks in pattern recognition followed by the general ANN architecture is presented first. In the next section the error-backpropagation algorithm used for training class-modular FFMLP is illustrated. The Final section deals with the class-modular neural network architecture used for the handwritten pattern classification studies followed by the description of simulation experiments and recognition results.

7.4.1 Neural Networks for Pattern Recognition

Artificial Neural Networks (ANN) can be most adequately characterized as computational models with particular properties such as the ability to adapt or learn, to generalize, to cluster or organize data, based on a massively parallel architecture. The history of ANNs starts with the introduction of simplified neurons in the work of McCulloch and Pitts [McCulloch.W.S and Pitts.W, 1943]. These neurons were presented as models of biological neurons and as conceptual mathematical neurons like threshold-logic devices that could perform computational task. The work of Hebb further developed the understanding of this neural model [Hebb.D.O, 1949]. Hebb proposed a qualitative mechanism describing the process by which synaptic connections are modified in order to reflect the learning process undertaken by interconnected neurons, when they are influenced by some environmental stimuli. Rosenblatt with his perceptron model, further enhanced our understanding of artificial learning devices [Rosenblatt.F., 1959]. However, the analysis by Minsky and Papert in their work on perceptrons, in which they showed the deficiencies and restrictions existing in these simplified models, caused a major set back in this research area [Minsky .M.L and Papert.S.A., 1988]. ANNs attempt to replicate the computational power (low level arithmetic processing ability) of biological neural networks and, there by, hopefully endow machines with some of the (higher-level) cognitive abilities that biological organisms possess. These networks are reputed to possess the following basic characteristics:

- Adaptiveness: the ability to adjust the connection strengths to new data or information
- Speed : due to massive parallelism
- Robustness: to missing, confusing, and/ or noisy data
- Optimality: regarding the error rates in performance

Several neural network learning algorithms have been developed in the past years. In these algorithms, a set of rules defines the evolution process undertaken by the synaptic connections of the networks, thus allowing them to learn how to perform specified tasks. The following sections provide an overview of neural network models and discuss in more detail about the learning algorithm used in classifying the handwritten characters, namely the Back-propagation (BP) learning algorithm.

7.4.2 General ANN Architecture

A neural network consists of a set of massively interconnected processing elements called neurons. These neurons are interconnected through a set of connection weights, or synaptic weights. Every neuron i has N_i inputs, and one output Y_i. The inputs labeled s_{i1}, s_{i2}, …, s_{iNi} represent signals coming either from other neurons in the network, or from external world. Neuron i has N_i synaptic weights, each one associated with each of the neuron inputs. These synaptic weights are labeled w_{i1}, $w_{i2}, \ldots, w_{iNi}$, and represent real valued quantities that multiply the corresponding input signal. Also every neuron i has an extra input, which is set to a fixed value θ, and is referred to as the threshold of the neuron that must be exceeded for there to be any activation in the neuron. Every neuron computes its own internal state or total activation, according to the following expression,

$$x_j = \sum_{i=1}^{N_i} w_{ij}\, s_{ij} + \vartheta_i \qquad j = 1, 2, \ldots\ldots, M$$

where M is the total number of Neurons and N_i is the number of inputs to each neuron. Figure 7.3 shows a schematic description of the neuron. The total activation is simply the inner product of the input vector $S_i = [s_{i0}, s_{i1}, \ldots, s_{iNi}]^T$ by the weight vector $W_i = [w_{i0}, w_{i1}, \ldots w_{iNi}]^T$. Every neuron computes its output according to a function $Y_i = f(x_i)$, also known as threshold or activation function. The exact nature of f will depend on the neural network model under study.

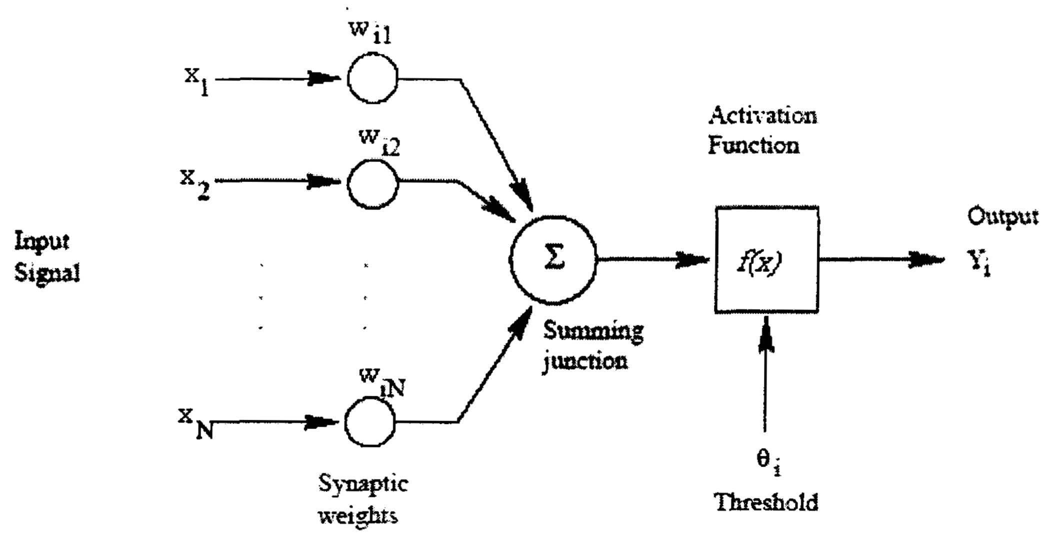


Figure 7.3 Simple neuron representation

In the present study, we use a mostly applied sigmoid function in the thresholding unit defined by the expression,

$$S(x) = \frac{1}{1+e^{-ax}}$$

This function is also called S-shaped function. It is a bounded, monotonic, non-decreasing function that provides a graded non-linear response as shown in figure 7.4.

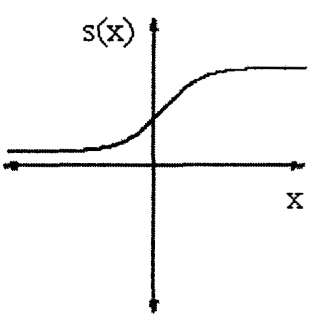


Figure 7.4. Sigmoid threshold function

The network topology used in this study is the feed forward network. In this architecture the data flow from input to output units strictly feed-forward, the data processing can extend over multiple layers of units but no feed back connections are present.

This type of structure incorporates one or more hidden layers, whose computation nodes are correspondingly called hidden neurons or hidden nodes. The function of the hidden nodes is to intervene between the external input and the network output. By adding one or more layers, the network is able to extract higher-order statistics. The ability of hidden neurons to extract higher-order statistics is particularly valuable when the size of the input layer is large. The structural architecture of the neural network is intimately linked to the learning algorithm used to train the network. In this study we used Error Back-propagation learning algorithm to train the input patterns in the multilayer feed forward neural network. The detailed description of the learning algorithm is given in the following section.

7.4.3 Back-propagation Algorithm for Training Feed-Forward Multi Layer Perceptron (FFMLP)

The back propagation algorithm (BP) is the most popular method for neural network training and it has been use to solve numerous real life problems. In a multi layer feed forward neural network BP performs iterative minimization of a cost function by making weight connection adjustments according to the error between the computed and desired output values. Figure 7.5 shows a general three layer network.

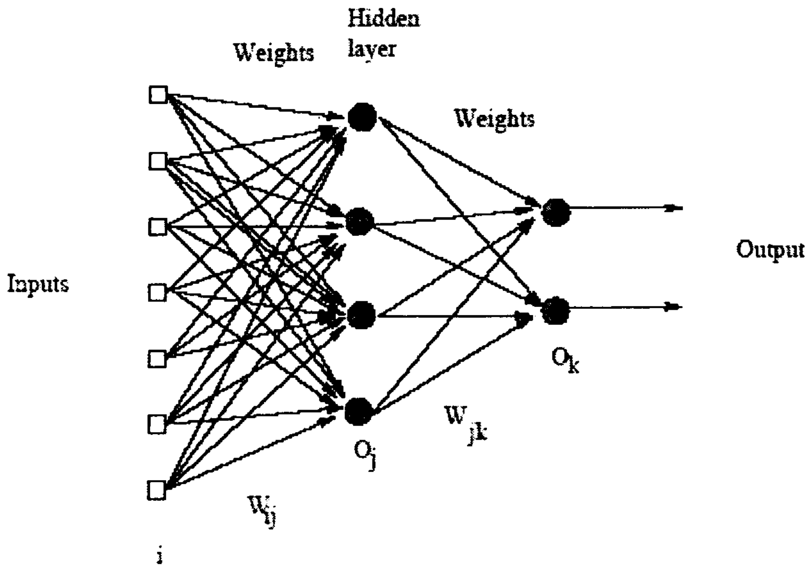


Figure 7.5. A general three layer network

The following relationships for the derivation of the back-propagation hold:

$$o_k = \frac{1}{1 + e^{-net_k}}$$

$$net_k = \sum_j w_{ik} o_j$$

$$o_j = \frac{1}{1+e^{-net_j}}$$

$$net_j = \sum_j w_{ij} o_i$$

The cost function (error function) is defined as the mean square sum of differences between the output values of the network and the desired target values. The following formula is used for this error computation,

$$E = \frac{1}{2}\sum_{p}(\sum_{k} t_{pk} - o_{pk})^2)$$

where p is the subscript representing the pattern and k represents the output units. In this way, t_{pk} is the target value of output unit k for pattern p and o_{pk} is the actual output value of layer unit k for pattern p. During the training process a set of feature vectors corresponding to each pattern class is used. Each training pattern consists of a pair with the input and corresponding target output. The patterns are presented to the network sequentially, in an iterative manner. The appropriate weight corrections are performed during the process to adapt the network to the desired behavior. The iterative procedure continues until the connection weight values allow the network to perform the required mapping. Each presentation of whole pattern set is named an *epoch*.

The minimization of the error function is carried out using the gradient-descent technique. The necessary corrections to the weights of the network for each iteration n are obtained by calculating the partial derivative of the error function in relation to each weight w_{jk}, which gives a direction of steepest descent. A gradient vector representing the steepest increasing direction in the weight space is thus obtained. Due to the fact that a minimization is required, the weight update value Δw_{jk} uses the negative of the corresponding gradient vector component for that weight. The delta rule determines the amount of weight update based on this gradient direction along with a step size,

$$\Delta w_{jk} = -\eta \frac{\partial E}{\partial w_{jk}}$$

The parameter η represents the step size and is called the learning rate. The partial derivative is equal to,

$$\frac{\partial E}{\partial w_{jk}} = \frac{\partial E}{\partial o_k} \frac{\partial o_k}{\partial net_k} \frac{\partial net_k}{\partial w_{jk}} = -(t_k - o_k) o_k (1 - o_k) o_j$$

The error signal δk is defined as,

$$\delta_k = (t_k - o_k) o_k (1 - o_k)$$

so that the delta rule formula becomes:

$$\Delta w_{jk} = \eta \delta_k o_j$$

For the hidden neuron, the weight change of w_{ij} is obtained in a similar way. A change to the weight, w_{ij}, changes o_j and this changes the inputs into each unit k, in the output layer. The change in E with a change in w_{ij} is therefore the sum of the changes to each of the output units. The change rules produces:

$$\frac{\partial E}{\partial w_{ij}} = \sum_k \frac{\partial E}{\partial o_k} \frac{\partial o_k}{\partial net_k} \frac{\partial net_k}{\partial o_j} \frac{\partial o_j}{\partial net_j} \frac{\partial net_j}{\partial w_{ij}}$$

$$= \sum_k -(t_k - o_k) o_k (1 - o_k) w_{jk} o_j (1 - o_j) o_i$$

$$= -o_i\, o_j (1 - o_j) \sum_k \delta_k w_{jk}$$

So that defining the error δj as:

$$\delta_j = o_j (1 - o_j) \sum_k \delta_k w_{jk}$$

we have the weight change in the hidden layer is equal to:

$$\Delta w_{ij} = \eta \delta_j o_i$$

The δ_k for the output units can be calculated using directly available values, since the error measure is based on the difference between the desired output t_k and the actual output o_k. However, that measure is not available for the hidden neurons. The solution is to back-propagate the δk values, layer by layer through the network, so that finally the weights are updated.

A momentum term was introduced in the back-propagation algorithm by Rumelhart [Rumelhart.D.E. *et al.*, 1986]. Here the present weight is modified by incorporating the influence of the passed iterations. Then the delta rule becomes

$$\Delta w_{ij}(n) = -\eta \frac{\partial E}{\partial w_{jk}} + \alpha \Delta w_{ij}(n-1)$$

where α is the momentum parameter and determines the amount of influence from the previous iteration on the present one. The momentum introduces a *damping* effect on the search procedure, thus avoiding oscillations in irregular areas of the error surface by averaging gradient components with opposite sign and accelerating the convergence in long flat areas. In some situations it possibly avoids the search procedure from being stopped in a local minimum, helping it to skip over those regions without performing any minimization there. Momentum may be considered as an approximation to a second–order method, as it uses information from the previous iterations. In some applications, it has been shown to improve the convergence of the back-propagation algorithm.

In the present study since the output class number is considerably large, a class-modular concept is incorporated with the above given FFMLP architecture with error Backpropagation algorithm. The following section describes the architecture and implementation details of the proposed system designed for the recognition of forty four Malayalam handwritten characters.

7.4.4 Class-modular Neural Network Architecture and Training Strategies

Modularity is an essential concept, which is supposed to be incorporated appropriately in the design of systems for diverse application areas of science and engineering. In the traditional non-modular neural network architecture, one large network is used to implement a system which discriminates the input samples into one of k class regions in a high-dimensional feature space. Determining the optimal decision boundaries in the k-classification module for 44-class Malayalam handwritten character recognition in a high-dimensional feature space is very complex. So it inevitably poses the very complex problem from both mathematical and perceptual viewpoints of determining the optimal decision boundaries, for all the classes involved in a common high-dimensional feature space. This seriously limits the recognition performance of the MLP system.

In order to overcome the above problem a class modularity concept suitable for the classification module using FFMLP is used in the present study. In this model, the original k-classification problem is decomposed into k 2-classification sub problems, each of which discriminates a class from the other k-1 classes. A 2-classification problem is then solved by the 2-classifier specifically designed for the corresponding class. Here the k 2-classifiers solve the original k-classification problem co-operatively and the class decision module integrates the outputs from the k 2-classifiers. Figure 7.6 shows the FFMLP architecture used for this k 2-classifier. The modular FFMLP classifier consists of k sub networks, $M_{\omega i}$ for $0 \leq i < k$, each responsible for one of the k classes. In the present experiment $k = 44$ where each k represents a character to be recognized. The specific task of $M_{\omega i}$ is to classify the patterns into two groups, Ω_0 and Ω_1 where $\Omega_0 = \{\omega_i\}$ and

$\Omega_1 = \{ \omega_j \mid 0 \leq j < k \text{ and } j \neq i \}$. Here we can design the architecture of $M_{\omega i}$ in the same way as the conventional non-modular FFMLP.

Each subnet $M_{\omega i}$ is designed with one input layer, one hidden layer and one output layer. These three layers are fully connected. The input layer has d nodes to accept the d-dimensional feature vector. The output layer has two nodes, denoted by O_0 and O_1 corresponding to the classes Ω_0 and Ω_1 respectively. The number of hidden layers and number of nodes in each hidden layer are fixed by the trial and error experiments. The architecture for the entire network constructed by k sub-networks is shown in figure 7.7

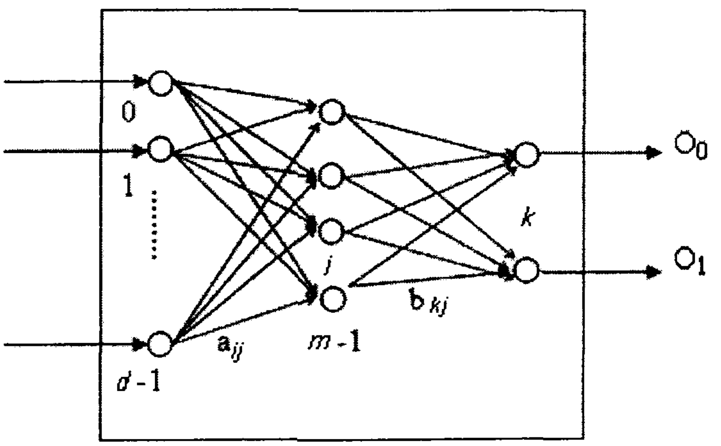


Fig. 7.6 The sub-network, $M_{\omega i}$, architecture used in class-modular FFMLP

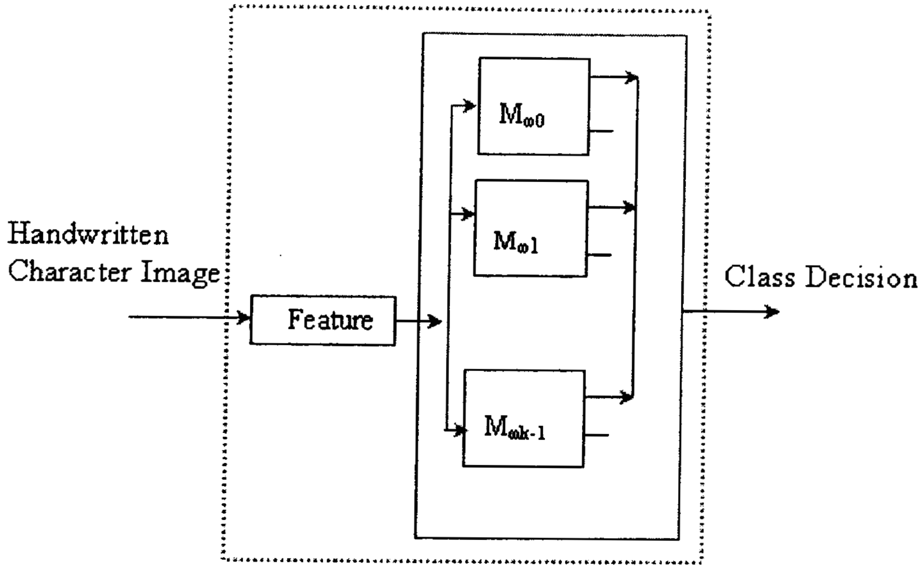


Figure 7.7 The whole network architectures for the class-modular FFMLP

The feature vectors extracted for each character image is applied to the input layer of every sub network corresponding to each class. Then each sub-network $M_{\omega i}$ calculates the forward process using its own weight set to produce an output vector $D = (O_0, O_1)$ and the final decision vector is constituted using O_0 only.

In the training phase each of k 2-classifeirs is trained independently of the other classes. The error-backpropagation algorithm is applied to each of 2-classifiers in the same way as the conventional FFMLP. The training set is prepared separately for each of k 2-classifieres. To train 2-classifier for class ω_i , we organize the samples in the training set into two groups, $Z_{\Omega 0}$ and $Z_{\Omega 1}$, such that $Z_{\Omega 0}$ has the samples from classes in Ω_0, and $Z_{\Omega 1}$ has the samples from classes in Ω_1, i.e.,

$$Z_{\Omega 0} = \bigcup_{\omega k \in \Omega 0} Z_{\omega k} \quad \text{and} \quad Z_{\Omega 1} = \bigcup_{\omega k \in \Omega 1} Z_{\omega k}$$

After preparing the training set for class ω_i, forward and backward computation processes are applied to train $M_{\omega i}$. Note that each class uses the same mathematical expression to estimate the values of its weight sets, $\boldsymbol{A}^{\omega c}$ and $\boldsymbol{B}^{\omega c}$. Forward pass for a 2-classifier of the class c where $0 \leq c < k$ is given below.

$$y_j = f\left(\sum_{i=0}^{d-1} a_{ji}^{c}\, x_i\right) \text{ for } 0 \leq j \leq m,$$

$$o_k = f\left(\sum_{j=0}^{m-1} b_{kj}^{c}\, y_j\right) \text{ for } 0 \leq k \leq 2,$$

In the backward pass the following computation processes are executed.

$$\Delta b_{kj}^{c} = \left[E\,(t_k - o_k)\, o_k\ (1 - o_k)\right] y_j$$

$$\text{for } 0 \leq j < m \text{ and } 0 \leq k < 2$$

$$\Delta a_{ji}^{c} = \left[E y_j\ (1 - y_j)\ \left(\sum_{k=0}^{1} (t_k - o_k)\, o_k (1 - o_k)\, b_{kj}^{c}\right)\right] x_i$$

$$\text{for } 0 \leq i < d \text{ and } 0 \leq j < m$$

$$b_{kj}^{c} = b_{kj}^{c} + \Delta b_{kj}^{c}$$

$$a_{ji}^{c} = a_{ji}^{c} + \Delta a_{ji}^{c}$$

When training cycles are completed for all the classes, the training process terminates and then the k final weight sets $\boldsymbol{A}^{\omega c}$ and $\boldsymbol{B}^{\omega c}$ for $0 \leq c < k$ are saved. These final weight sets are used in the recognition stage. In order to recognize an unknown input character pattern, the class decision module takes only the values of O_0 and uses the *winner-takes-all* scheme to determine the final class.

7.4.5 Simulation Experiments and Results

Present study investigates the recognition capabilities of the above explained class modular FFMLP-based Malayalam handwritten character recognition system. For this purpose the multilayer feed forward neural network with class-modular architecture is simulated with the Backpropagation learning algorithm. A constant learning rate, 0.1, is used. The initial weights are obtained by generating random numbers ranging from 0.1 to 1. The number of nodes in the input layer is fixed according to the feature vector size. Since each sub-network produces two output vectors the number of nodes in the output layer is fixed as 2 in each case. The recognition experiment is repeated by changing the number of hidden layers and number of nodes in each hidden layer. After this trial and error experiment, the number of hidden layers is fixed as one and the number of nodes in the hidden layer is set to eighteen for obtaining the successful architecture in the present study.

After initiating the training, the Mean-Square-Errors (MSE) at the output layers of k 2-classifiers is monitored separately in order to determine the termination of training. The training process is terminated when the MSE is less than $\varepsilon1$, or the most recent n epochs produce the MSE less than $\varepsilon2$ on the average or the number of epochs exceeds T. The error tolerances $\varepsilon1$ is fixed as 0.01, $\varepsilon2$ as 0.1 and number of *epochs* T as 10,000 as it was found optimum by trial and error experiment.

The network is trained using the SSPD feature extracted from the gray-scale images, Z-VD and FZ-NVD features modeled from the pre-processed images of the Malayalam handwritten characters separately. Here we used a set of 15,752 samples of the forty four Malayalam handwritten characters collected from 358 writers for iteratively computing the final weight matrix and a disjoint set of character patterns of same size from the database for recognition purpose. The final output after a successful *epoch* is given below.

```
---------------------------------------------

---->average error per cycle = 0.042964 <---

---->error last cycle = 0.009869 <---

---->error last cycle per pattern= 0.006474 <-

------------>total epochs = 7302 <---

------------>total patterns = 105041 <---

---------------------------------------------
```

The recognition accuracies obtained for the forty four basic Malayalam handwritten characters based on above said features using class-modular neural network classifier are tabulated in Table 7.3. The graphical representation of these recognition results based on different features using class-modular neural network as shown in figure 7.8(a-d)

Character Number	Character	Recognition Accuracy (%)		
		SSPD Feature	Z-VD Feature	FZ-NVD Feature
1	അ [-ah]	79.89	98.04	98.32
2	ആ [-aah]	62.45	64.25	69.55
3	ഇ [-yi]	98.04	86.59	87.71
4	ഉ [-uh]	76.82	77.09	79.61
5	ഋ [-eru]	64.25	68.16	75.14
6	എ [-eh]	87.99	88.27	93.58
7	ഏ [-aeh]	86.59	75.41	87.15
8	ഒ [-oh]	75.41	87.15	88.27
9	ക [ka]	94.97	76.53	79.05
10	ഖ [kha]	75.41	75.58	82.12
11	ഗ [ga]	77.37	94.69	95.25
12	ഘ [gha]	74.02	76.25	75.41
13	ങ [nga]	78.21	78.77	82.40
14	ച [cha]	69.83	82.21	82.96
15	ഛ [chha]	75.41	78.21	79.61
16	ജ [ja]	74.30	75.14	76.82
17	ഝ [jha]	82.96	83.52	87.99
18	ഞ [nja]	84.08	81.56	89.38
19	ട [ta]	65.08	67.04	72.62
20	ഠ [tta]	68.44	69.27	74.86
21	ഡ [da]	92.17	94.13	95.25
22	ഢ [dda]	67.04	67.31	74.02

(Continued)

Table 7.3 Recognition accuracies of forty four basic Malayalam handwritten characters based on SSPD, Z-VD, FZ-NVD features using class-modular neural network

Character Number	Character	Recognition Accuracy (%)		
		SSPD Feature	Z-VD Feature	FZ-NVD Feature
23	ണ[nha]	67.04	76.26	85.47
24	ത [tha]	58.94	65.08	74.30
25	ഥ [thha]	89.38	89.38	87.43
26	ദ [dha]	75.98	80.45	80.72
27	ധ[dhha]	86.03	89.11	89.38
28	ന [na]	73.74	89.38	85.75
29	പ [pa]	67.60	69.83	68.77
30	ഫ [pha]	84.64	69.55	72.62
31	ബ [ba]	67.04	64.53	75.41
32	ഭ [bha]	64.80	87.15	87.43
33	മ [ma]	57.82	65.36	67.04
34	യ [ya]	50.56	65.64	68.99
35	ര [ra]	86.59	67.04	72.62
36	ല [la]	53.07	62.01	64.25
37	വ [va]	67.32	88.83	90.78
38	ശ [sha]	67.88	59.49	64.80
39	ഷ[shha]	62.57	64.25	67.04
40	സ [sa]	54.44	67.04	68.44
41	ഹ [ha]	63.13	75.41	78.21
42	ള [lha]	58.10	60.05	62.29
43	ഴ [zha]	63.97	65.64	65.64
44	റ [rha]	75.14	58.66	67.58
Average Recognition accuracy (%)		73.03386	75.57523	78.87182

Table 7.3 Recognition accuracies of forty four basic Malayalam handwritten characters based on SSPD, Z-VD, FZ-NVD features using class-modular neural network

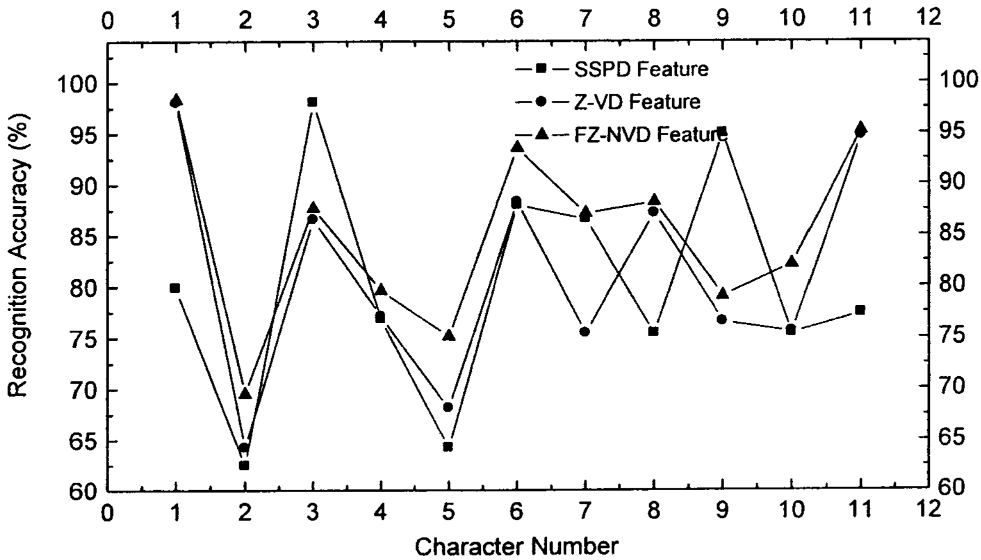


Figure 7.8(a)

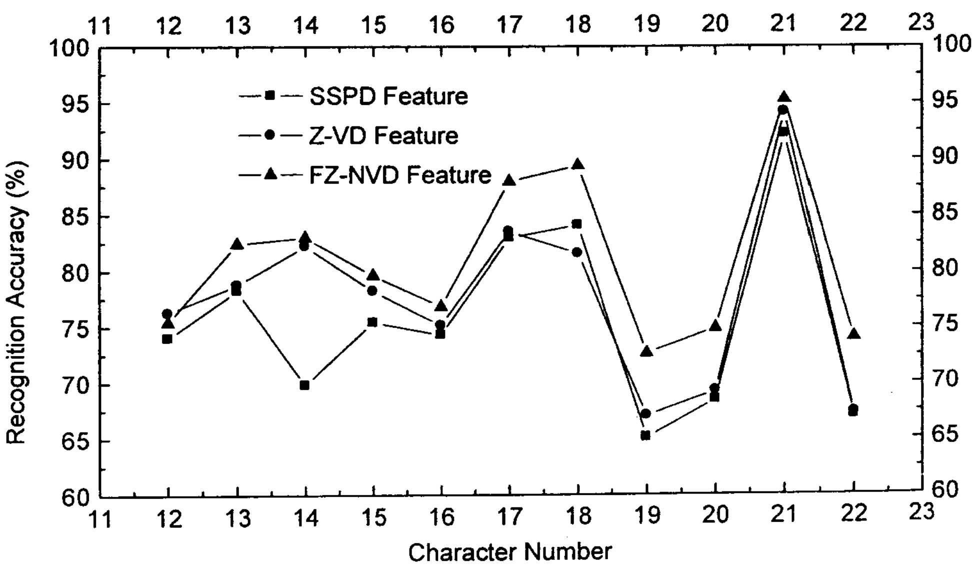


Figure 7.8(b) (Continued)

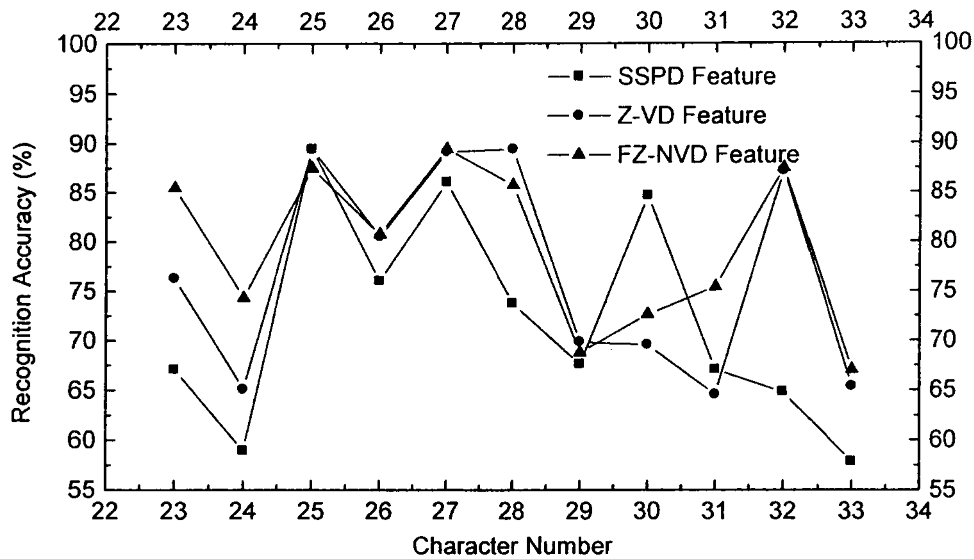


Figure 7.8(c)

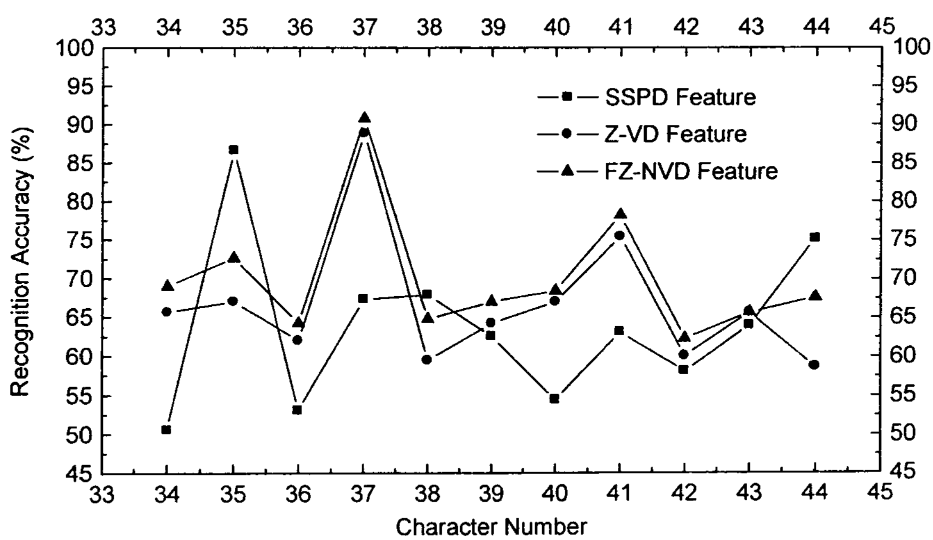


Figure 7.8(d)

Figure 7.8(a-d) Recognition accuracies of forty four Malayalam handwritten characters using SSPD, Z-VD and FZ-NVD features using class-modular neural network

The recognition accuracies on the basis of performance of the classifiers used in this study and the recognition results obtained using different features with three different classifiers are compared and analyzed. Figure 7.9(a-b) shows the recognition accuracies obtained for forty four Malayalam handwritten characters using SSPD features and three different classifiers. Figure 7.10 (a-b) shows the recognition accuracies obtained for forty four Malayalam handwritten characters using Z-VD features and the three different classifiers. The recognition accuracies obtained for forty four Malayalam handwritten characters using FZ-NVD features and the three different classifiers including c-Means clustering, k-NN classifier and Class-modular artificial neural network classifier are shown in Figure 7.11(a-b).

From the above classification experiments, the overall highest recognition accuracy (78.87%) is obtained for the FZ-NVD features using class-modular FFMLP. Compared to the recognition results, obtained using *c*-Means clustering (69.00 %) and *k*-NN classifier (75.82%) based on FZ-NVD feature, the class-modular neural network gives better performance. These results indicate that, for large class pattern recognition problems the connectionist model based learning is more adequate than the existing statistical classifiers and clustering algorithms.

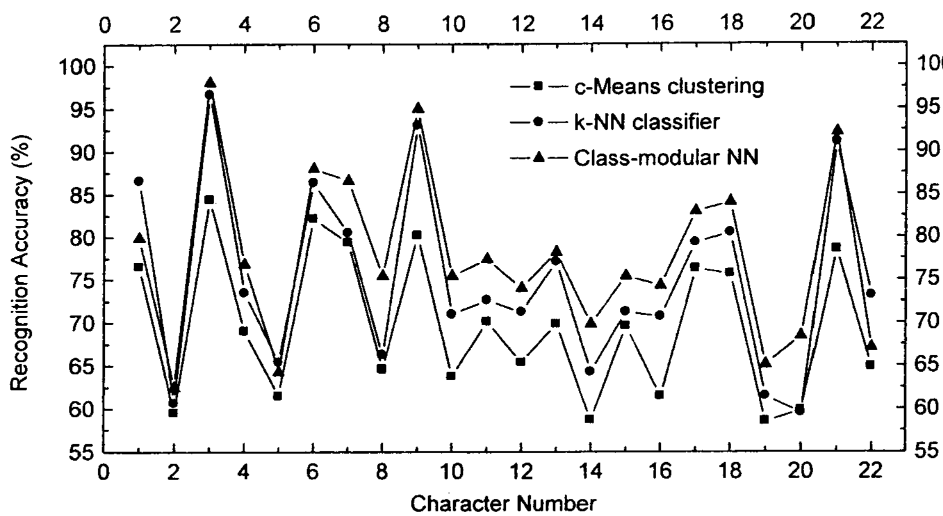


Figure 7.9 (a)

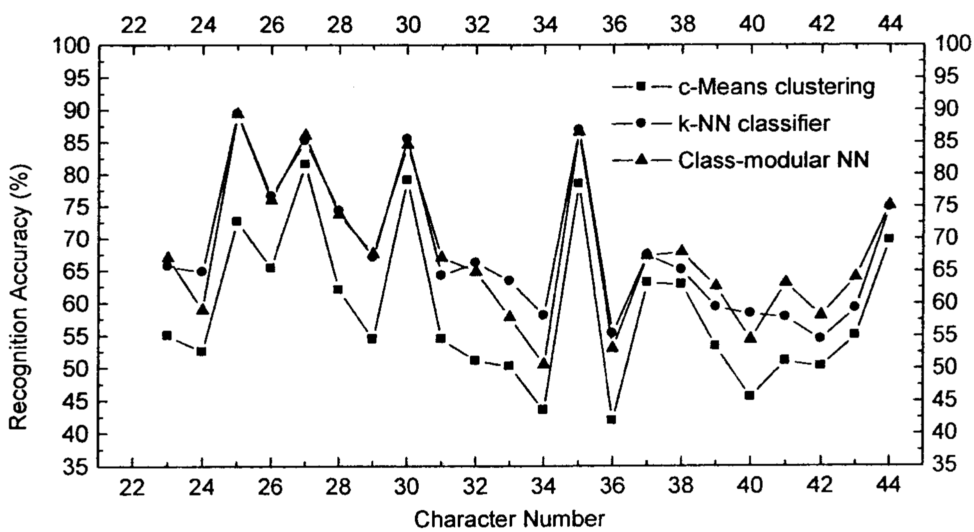


Figure 7.9(b)

Figure 7.9 (a-b) Recognition accuracies of forty four Malayalam handwritten characters based on SSPD features and three different classifiers

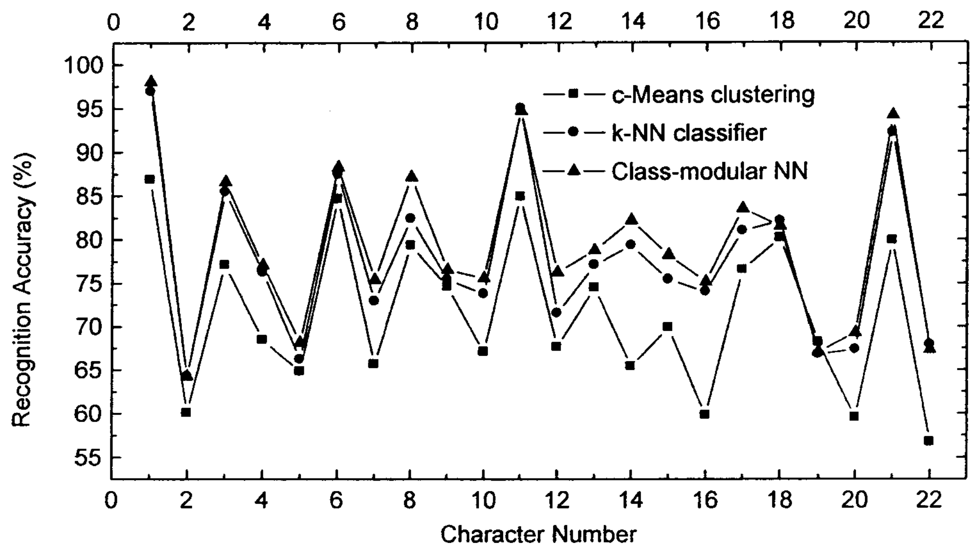


Figure 7.10(a)

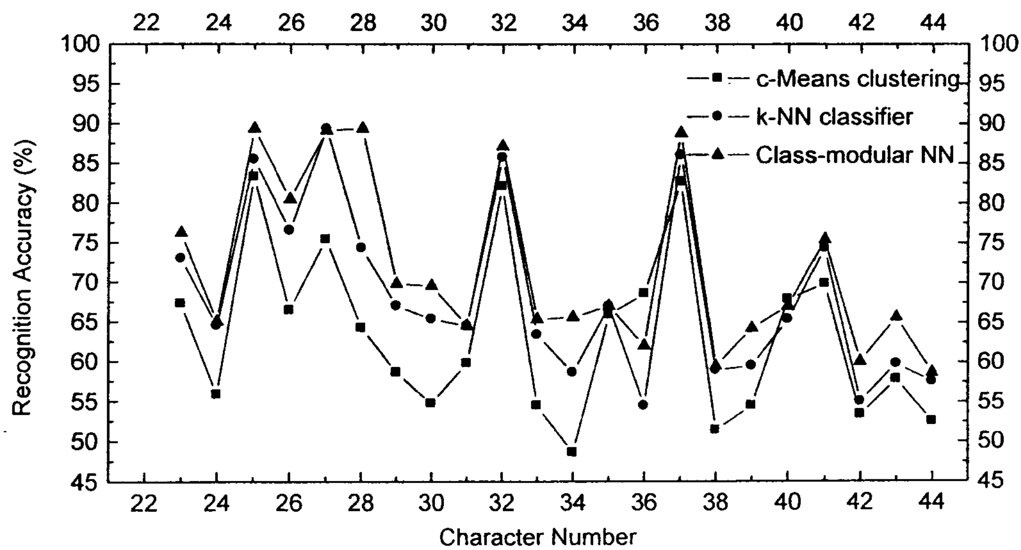


Figure 7.10(b)

Figure 7.10 (a-b) Recognition accuracies of forty four Malayalam handwritten characters based on Z-VD features and three different classifiers

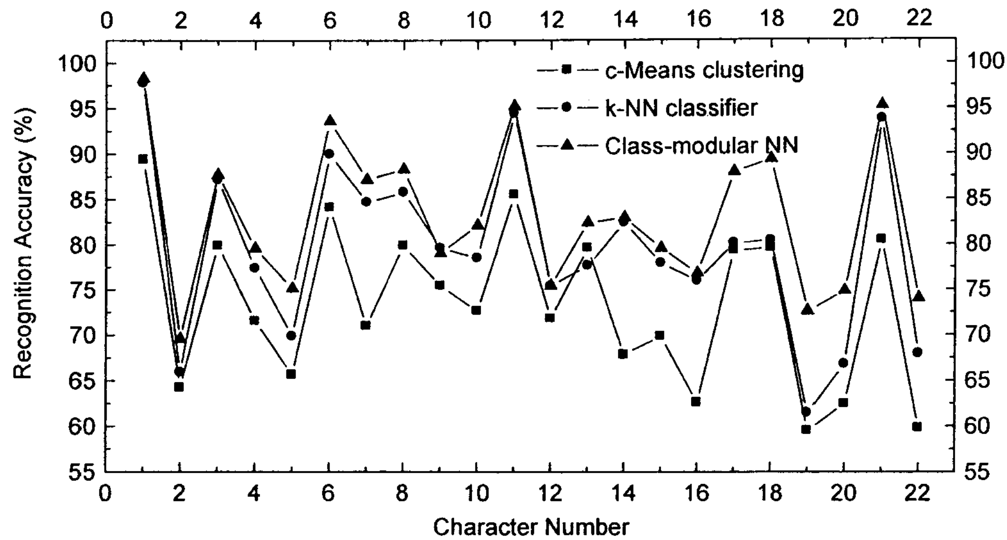


Figure 7.11(a)

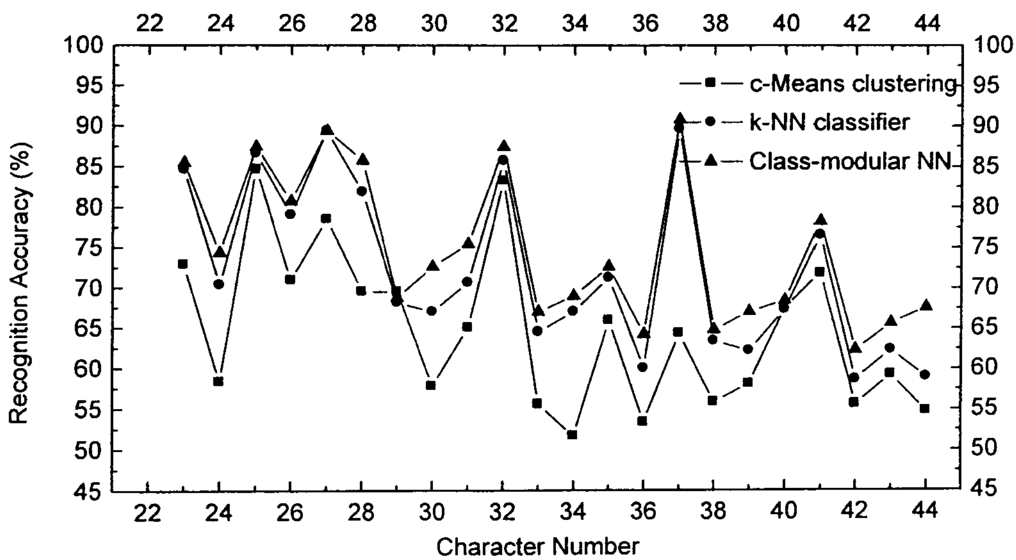


Figure 7.11(b)

Figure 7.11 (a-b) Recognition accuracies of forty four Malayalam handwritten characters based on FZ-NVD features and three different classifiers

7.5 Conclusions

Handwritten character recognition studies based on the parameters developed in chapter 5 and 6 using different classifiers are presented in this chapter. The cluster analysis using the *c*-Means clustering technique is conducted and further the same technique is used for the recognition of character patterns. The credibility of the extracted parameters is also tested with the k-NN classifier. A connectionist model based recognition system by means of class-modular neural network is then implemented and tested using SSPD features extracted from the gray-scale images, Z-VD and FZ-NVD features extracted from the pre-processed character images. The highest recognition accuracy (78.87%) is obtained based on the FZ -NVD feature using class-modular FFMLP classifier. From the above analysis it can be concluded that FZ-NVD features using class-modular FFMLP give better results compared to the other parameters and the classifiers used. These results also specify the need for improving the classification algorithm in order to fully accommodate the small variations present in the extracted features. To this end, FZ-NVD parameter is further used for developing a character recognition system with the help of a combined neuro-fuzzy classifier for better performance in the next chapter.

CHAPTER 8

ADAPTIVE NEURO-FUZZY INFERENCE SYSTEM FOR HANDWRITTEN CHARACTER RECOGNITION

8.1 Introduction

In chapter 7 we have investigated the credibility of different parameters introduced in this work, viz., SSPD, Z-VD and FZ-NVD by performing recognition experiments on the handwritten character database using *c*-Means clustering technique, *k*-NN classifier and class-modular neural network. The performance of the class-modular neural network classifier indicates not only the efficiency of the parameters introduced but also the relevance of connectionist model based classifiers in large class HCR problems. The recognition accuracies obtained in these experiments also reveal the need of an improved system that can effectively handle the large variations present in the handwritten samples caused by the individual writing styles. The best alternative suggested in this study is an adaptive network based fuzzy-inference system. This architecture facilitates the judicial integration of the neural network (recognize patterns and adapt themselves to cope with the changing environments) and fuzzy inference systems (incorporate human knowledge and perform inference mechanism and decision making). Generation of a fuzzy-rule-set from the extracted feature vectors and the decision tree induction based pre-processing of the rule set for input space partitioning are carried out for enhancing the performance of the system.

In real life applications, fuzzy rules are learned by humans from real data and facts. If it is assumed that the fundamental learning mechanism of human is some type of neural network, these fuzzy rules must then reside in the neural networks in brain. Although fuzzy logic and neural networks are generally viewed as different modeling ideas, recently successful attempts are made to incorporate fuzzy rules in neural networks. Fuzzy sets and fuzzy logic remain as a means for representing, manipulating and utilizing uncertain information and to provide a framework for handling uncertainties and imprecision associated with real world problems. Neural networks provide computational power, fault tolerance and learning capability to the systems. Significant research works in developing methods and models in fuzzy logic neural networks (neuro-fuzzy networks) have been carried out by Mamdani, Takagi, Sugeno and Kang [Mamdani.E.H and Assilian, 1975], [Takagi.T and Sugeno.M, 1985], [Sugeno.M and Kang.G.T, 1988], [Jang.J.-S.R, *et al.*, 2002].

In this study, an effort is made to build a recognition system using the integrated and unified model of fuzzy logic and neural network architecture for Malayalam handwritten characters. For this purpose, the FZ-NVD features extracted from handwritten character patterns by fuzzy zoning method, as discussed in chapter 6 are fuzzified and IF…THEN rule set is framed out.

These rule set is pre-processed using the ID3 [Quinlan.J.R, 1986] decision tree induction procedure for preparing them for input space partitioning. These ID3 derived rules are then used for training the Adaptive Neuro-Fuzzy Inference System (ANFIS) [Jang.J.-S.R. *et al.*, 1993]. The recognition results show that this method is more efficient and can be adopted for developing a complete HCR system for Malayalam. This chapter is organized as two sections. The first section describes the procedure for generating the ID3 driven fuzzy-rule set from the FZ-NVD feature vectors. The second section deals with the character recognition experiments performed on the basis of ANFIS.

8.2 Construction of ID3 Driven Fuzzy-rule-set for Handwritten Character Recognition

In general, learning (or any other parameter level adaptation methods) of an adaptive network based fuzzy inference systems only deals with parameter identification. So we require methods for determining the initial system architecture (structure identification) before parameter tuning. The structure identification in fuzzy system modeling involves the following primary issues, viz.,

- Selecting relevant input variables
- Determining an initial system architecture including
    1. Input space partitioning
    2. Number of membership functions (MFs) for each input

3. Number of fuzzy *if.....then* rules

4. Antecedent (premise) part of the rules.

5. Consequent (conclusion) part of fuzzy rules

- Choosing initial parameters for MFs

The following sub sections describe the methods used for generating the fuzzy-rule set from the extracted features followed by the ID3 decision tree induction procedure used in pre-processing these rules for preparing them for input space partitioning.

8.2.1 Fuzzy-rule-set from the Feature Vectors

Fuzzy rules and fuzzy reasoning are the backbone of fuzzy inference systems, which are the most important modeling tool based on fuzzy set theory. Numeric analysis approach of fuzzy system was first presented by Takagi and Sugeno [Takagi.T and Sugeno.M, 1985] and then a lot of studies have been made in this area [Amano.A and Arisuka.T, 1989], [Chen.M.-Y and Linkens.D.H, 2001], [Iyatomi.H and Hagiwara.M, 2002]. The systems using fuzzy theory express rules or knowledge as *if......then* form. Even if they do not need mathematical analysis for modeling they need an appropriate model construction and parameter selection. In this work the fuzzy-zoning based FZ-NVD features extracted in chapter 6 are used for framing the fuzzy-rules. For this purpose each of the nine attributes of FZ-NVD feature vector is fuzzified. Fuzzy membership functions for representing the fuzzy sets SMALL, MEDIUM and LARGE are designed for this fuzzification process. The *Gaussian* membership function used for modeling these fuzzy sets is given by the expression,

$$f(x,\sigma,c) = e^{-(x-c)^2/(2\sigma^2)}$$

where x is the input variable, σ and c are constant parameters. The membership functions for the required fuzzy sets are constructed by the proper selection of the parameter set $\{\sigma, c\}$. We can adjust these parameters for varying the width and centre of the membership function. Here the fuzzy sets SMALL, MEDIUM and LARGE are modeled using the above *Gaussian* function by fixing the value of c as 0.25, 0.5 and 0.75 respectively. The value of σ is fixed as 0.1 in all the three cases.

Figure 8.1 illustrates the symmetric *Gaussian* membership functions modeled to represent the fuzzy sets SMALL, MEDIUM and LARGE defined by $f(x, 0.1, 0.25)$, $f(x, 0.1, 0.5)$ and $f(x, 0.1, 0.75)$. The FZ-NVD feature vectors extracted for forty four basic Malayalam handwritten characters are then converted into fuzzy rules. For this purpose the feature vectors are normalized to 0-1 range. The membership value for each attribute in the feature vector is found out using the membership functions corresponding to the fuzzy sets SMALL, MEDIUM and LARGE. The fuzzy membership function which gives maximum membership value for each attribute in the feature vector is identified. These attributes are then replaced with the fuzzy set corresponding to the highest valued membership function. The antecedent parts of fuzzy rules are framed by this process and the class number (1 to 44) representing each character pattern is set as the consequent part of each rule.

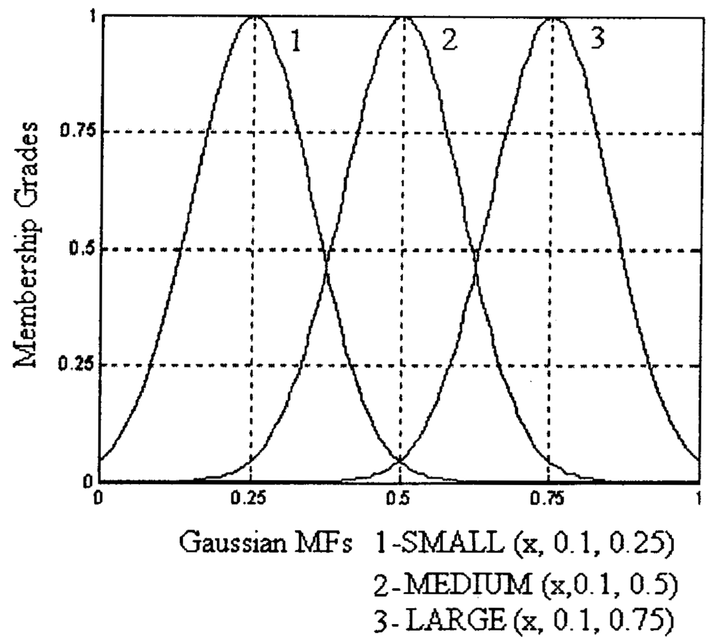


Figure 8.1 ***Gaussian*** **membership functions representing the fuzzy sets SMALL, MEDIUM and LARGE**

8.2.2 Decision Tree Induction Based Pre-processing of Fuzzy-rule-set

In a decision tree each branch node represents a choice between a number of alternatives, and each leaf node represents a classification or decision. The key idea of decision tree induction is to determine the attribute that best classifies the training data and then to use this attribute at the root of the tree. This process is repeated for each branch. Here we perform *top-down, greedy search* through the space of possible decision boundaries. The decision tree mechanism is transparent and we can easily follow a tree structure to explain how a decision is made. The decision trees are robust to classification errors and attribute errors in training data and can be used even if instances have missing attributes. This mechanism has been extensively used in machine learning [Hunt.E.B *et al*, 1966]. In the present study, we used the decision tree induction procedure for pre-processing the rule set obtained in section 8.2.1 in order to make them suitable for input space partitioning. The purpose of this pre-processing is to reduce the rule set derived from the feature vectors into an optimum number of rules with maximum information about the class boundaries. Here we used the well-known ID3 decision tree induction procedure proposed by Quinlan [Quinlan.J.R, 1986] to perform this task. This algorithm is framed on the basis of information theory and uses Information Entropy and Gain to determine the most informative attribute in each stage.

The fundamental premise of information theory is that the generation of information can be modeled as a probabilistic process that can be measured in a manner that agrees with intuition. In accordance with this supposition, a random event e that occurs with probability $p(e)$ is said to contain

$$I(e) = \log \frac{1}{p(e)} = -\log p(e)$$

units of information. The quantity $I(e)$ often is called the self-information of e. Generally, the amount of self-information attributed to event (e) is inversely related to the probability of e. If p (e) =1 (that is, event always occurs), I (e) = 0 and no information is attributed to it. That is, because no uncertainty is associated with the event, no information would be transferred by communicating that event has occurred. The overall level of uncertainty (termed Entropy) is given by the expression,

$$E(e) = -\sum_i p_i(e) \ \log_2 p_i(e)$$

The ID3 family of decision tree induction algorithms use the concept of entropy and information gain based on the information theory to decide which attribute, shared by a collection of instances, is to be used to split the rule set. Attributes are chosen repeatedly in this way until a complete decision tree that classifies every input is obtained. The following section describes the information Entropy and Gain which are the main factors that determine the decision tree induction in the ID3 procedure.

8.2.2.1 Entropy and Information Gain for Decision Tree Induction

In order to define information gain precisely, we begin by defining a measure commonly used in information theory, called entropy. One interpretation of entropy from information theory is that it specifies the minimum number of bits of information needed to encode the classification of an arbitrary element of the training set S (i.e., a member of S drawn at random with uniform probability). The entropy of S relative to this classification is defined as,

$$E(S) = \sum_{i=1}^{c} -p_i \log_2 p_i$$

S : Starting data set

c : Number of target classes

p_i : Proportion of examples in S belonging to target class i

It is also noted that, if all instances in S belong to the same class, then $E(S) = 0$ and if S contains the same number of instances for each class, then $E(S) = 1$. The entropy thus obtained is used for finding the Information Gain.

The Information Gain is defined as a measure of the effectiveness of an attribute in classifying the training data. It is simply the expected reduction in entropy caused by partitioning the examples according to this attribute. In general, the information gain, Gain (S, A) of an attribute A, relative to a collection of examples S, is defined as

$$\text{Gain}(S, A) = E(S) - \sum_{v \in \text{Values}(A)} \frac{|S_v|}{|S|} E(S_v)$$

Values (A): Set of all possible values of attribute A

S_v : Subset of S for which A has value v

$|S|$: Size of S

$|S_v|$: Size of S_v

The first term in the above expression is the entropy of the original collection of the input rules S and the second term is the expected value of the entropy after S is partitioned using attribute A. The expected entropy described by the second term is the sum of the entropies of each subset S_v, weighted by the fraction of examples $\frac{|S_v|}{|S|}$ that belong to S_v. The Gain (S, A) is therefore the expected reduction in entropy caused by knowing the value of attribute A. In other words, Gain (S, A) is the information provided about the target function value, given the value of some other attribute A. In the information theoretical view, it is the number of bits saved when encoding the target value of an arbitrary member of S, by knowing the value of attribute A. The succeeding section describes the implementation details of the ID3 algorithm.

8.2.2.2 Implementation of Decision Tree Induction Using ID3 Procedure

In the ID3 algorithm an attribute is selected for each node of the tree. This selection is made by finding the attribute that is most useful for classifying examples. A good quantitative measure of the worth of an attribute is the statistical property called information gain. The information gain measures how well a given attribute separates the training examples according to their target classification. ID3 uses this Information Gain measure for selecting a particular candidate attribute at each step while growing the tree.

As the first step, the attribute that has the highest information gain on the training set is identified. Use this attribute as the root of the tree, and then create a branch for each of the values that the attribute can take. The same process is repeated for each of the branches using the subset of the training set that is classified by this branch. This process will reduce the input rule set into a set of minimum number of rules with highest information about the target classes. The pseudo-code for the recursive ID3 procedure is given below. The input to the procedure is a collection of instances (rules) and their correct classification and output is a decision tree with each of its paths representing a decision rule. This output tree represents the pre-processed (reduced) rule set that is capable of partitioning the input-space, by identifying the mutually exclusive regions representing the target classes.

Algorithm: ID3 decision tree induction

Input: Fuzzy-rule set framed from the FZ-NVD feature vectors representing each character patterns. Initially the procedure will be executed by taking the entire training rules with complete attributes as input.

Examples: A m x n matrix where m is the number of rules (samples) in the training set and n is the feature vector dimension (number of attributes in each rule).

Target_class: A m sized vector with the target class information.

Attribute: The attribute set in the current pass and its size will reduce after each pass.

Output : A decision tree and, each path of it represents a decision rule.

```
ID3 (Examples, Target_class, Attributes)
Create a new node Root for the tree;
If all members of Examples are in the same class c
        Root = single-node tree with label = c;
else if Attribute is empty
        Root = Single node tree with label = most common value of
                Target_class in Examples;
else
        A: = member of Attributes that maximizes Gain (Examples, A);
        A is decision attribute for Root;
        For each possible value v of A
            Add a new branch below Root, testing for A = v;
            Examples_v := subset of Examples with A = v;
              If Examples_v is empty
                    below the new branch add a leaf with
                    label = most common value of Target_class in
                    Examples;
              else
                    below the new branch add subtree
                    ID3 (Examples_v, Target_class, Attributes – {A});
        return Root;
```

To perform the decision tree induction procedure, the fuzzy rules framed for each character patterns with the class numbers as consequent parameters are given as input to the ID3 procedure. Since the feature vectors used to frame these fuzzy rules are based on fuzzy zoning, each attribute in the rule represents the nine fuzzy regions of the character image as discussed in chapter 6. Each attribute can have any of the discrete values SMALL, MEDIUM or LARGE. This is achieved by the help of the fuzzy sets modeled using the *Gaussian* membership function as described in section 8.2.1. Here nine attributes are present in the antecedent part of each rule and are named as Zone1, Zone2, ..., Zone9. Any of the target class number (0 to 44) is assigned in the consequent part of each rule. So each rule or instance is considered as a combination of attribute-value pairs. To perform decision tree induction the ID3 algorithm is executed by taking these fuzzy if-then rules framed for each character as input. Each path in the output decision tree represents a decision rule. *Inorder tree traversal* algorithm is used to identify each rule from the out put decision tree. Experimental results show that as an average case the decision tree induction procedure reduces the input rule set into roughly one third of it. This final ID3 driven rule set is also capable of partitioning the input space into mutually exclusive regions in an impressively fast manner by the help of a recognition algorithm. Hence the above obtained ID3 driven fuzzy-rule set is given as input to ANFIS for learning. The detailed architecture and the learning algorithm used to implement ANFIS for the recognition of Malayalam handwritten characters are discussed in the succeeding sections.

8.3 Adaptive Network Based Fuzzy Inference System for Handwritten Character Recognition

This section describes the fuzzy inference system architecture along with the learning algorithm used for the recognition of Malayalam handwritten characters. The architecture used here is referred to as adaptive network-based fuzzy inference system or semantically equivalently, Adaptive Neuro-Fuzzy Inference System (ANFIS). Fuzzy inference system is a popular computing framework based on the concepts of fuzzy set theory, if-then rules and fuzzy reasoning. The basic structure of fuzzy inference system consists of three components: A rule base, which contains a selection of fuzzy rules; a database, which defines the membership functions used in the fuzzy rules; and a reasoning mechanism, which performs the inference procedure upon the rules and given facts to derive a reasonable output or conclusion. With crisp inputs and crisp outputs, a fuzzy inference system implements a nonlinear mapping from its input space to the output space. This mapping is accomplished by a number of if-then rules, each of which describes the local behavior of this mapping. In particular, the antecedent of a rule defines a fuzzy region in the input space, while the consequent specifies the output of that fuzzy region. A block diagram of fuzzy inference system is shown in figure 8.2.

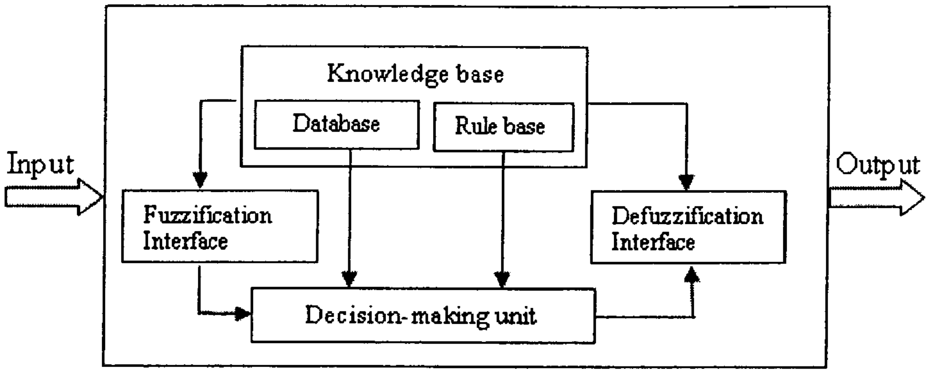


Figure 8.2 A block diagram of fuzzy inference system

In the present study, we used the *Sugeno* fuzzy model (TSK fuzzy model) proposed by Takagi, Sugeno and Kang [Takagi.T and Sugeno.M, 1985], [Sugeno.M. and Kang.G.T ,1988] for the implementation of the fuzzy inference system. Even though different consequent constituents result in different fuzzy inference systems, their antecedents are always the same. The antecedent of a fuzzy rule defines a local fuzzy region, while the consequent can be a membership function (Mamdani and Tsukamoto fuzzy models), a constant value (zero-order *Sugeno* model), or a linear equation (first-order *Sugeno* model).

A typical fuzzy rule in a *Sugeno* fuzzy model has the form

$$\textit{If } x \text{ is } A \text{ and } y \text{ is } B \textit{ then } z = f(x, y)$$

where A and B are fuzzy sets in the antecedent, while $z = f(x, y)$ is a crisp function in the consequent.

Usually $f(x, y)$ is a polynomial in the input variables x and y, but it can be any function as long as it can appropriately describe the output of the model within the fuzzy region specified by the antecedent of the rule. In this study $f(x, y)$ is considered as a first-order polynomial and hence the resultant fuzzy inference system is the first-order *Sugeno* fuzzy model.

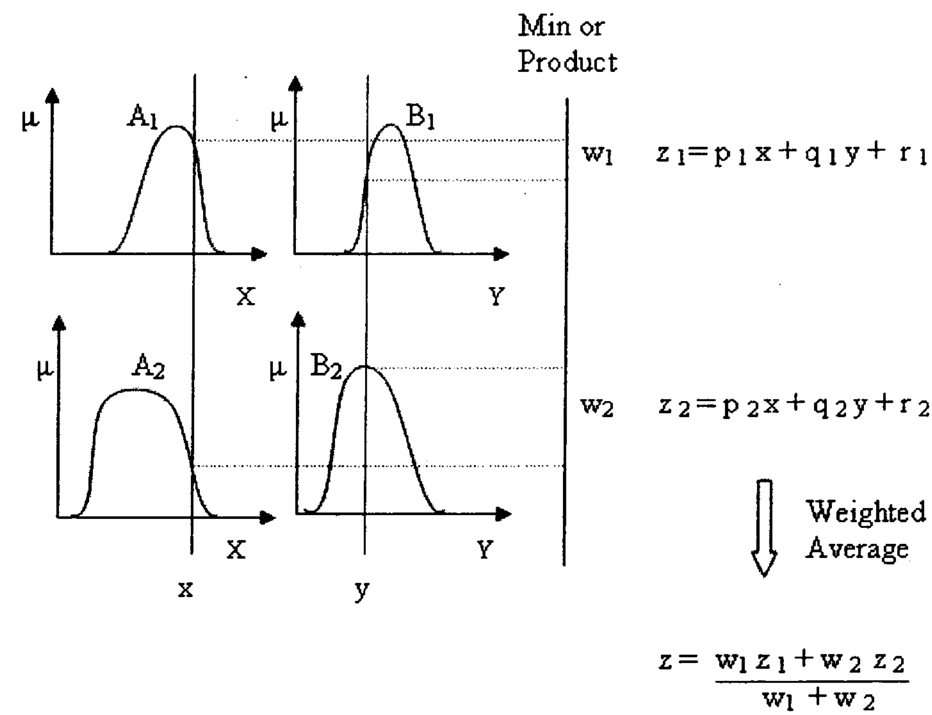


Figure 8.3 Fuzzy reasoning procedure for first-order *Sugeno* fuzzy model with two rules

Figure 8.3 shows the fuzzy reasoning procedure for the *Sugeno* fuzzy model. Since each rule has a crisp output, the overall output is obtained via weighted average. Thus the process of defuzzification is avoided. In practice the weighted average operator is replaced with the weighted sum operator (that is, $z = w_1z_1 + w_2z_2$ in the figure 8.3) to reduce computation further, especially in the training part of the system. It is hence identified that, with out the time-consuming and mathematically intractable defuzzification operation, the *Sugeno* model is the most suitable candidate for the handwritten character feature based fuzzy modeling. The following section describes the general ANFIS architecture and the gradient based learning procedure used for training the network.

8.3.1 General ANFIS Architecture

The generalized ANFIS architecture [Jang J.-S.R, 1993] is depicted in this section. For simplicity, we assume that the fuzzy inference system under consideration has two inputs x and y and one output z. Suppose that the rule base contains two fuzzy if-then rules of first-order *Sugeno* fuzzy model as given below.

Rule1: *If* x is A_1 and y is B_1, *then* $f_1 = p_1x + q_1y + r_1$

Rule 2: *If* x is A_2 and y is B_2, *then* $f_2 = p_2x + q_2y + r_2$

The *Sugeno* fuzzy model based ANFIS architecture is shown in figure 8.4. Here nodes of the same layer have similar functions as described below. We denote the output of the i^{th} node in layer l as $O_{l,i}$. The operations performed on each layer of the ANFIS network is explained below.

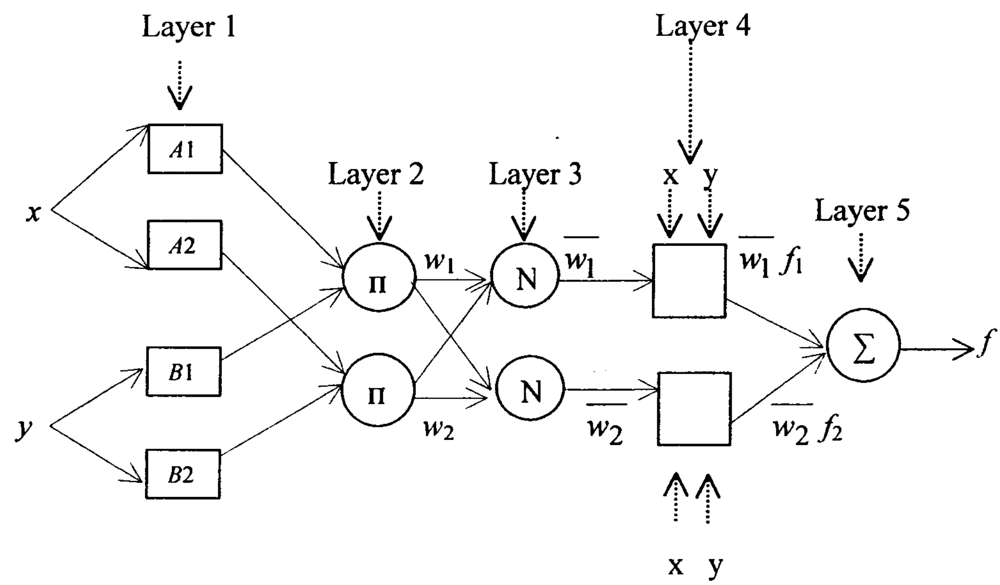


Figure 8.4 ***Sugeno*** **fuzzy model based ANFIS architecture**

Layer 1:

Every node i in this layer is an adaptive node with a node function

$$O_{1,i} = \mu A_i(x), \quad \text{for } i = 1, 2 \text{ or}$$

$$O_{1,i} = \mu B_{i-2}(y), \quad \text{for } i = 3, 4.$$

where x (or y) is the input to node i and A_i (or B_{i-2}) is the linguistic label associated with this node function. In other words, $O_{1,i}$ is the membership function of a fuzzy set A (=A_1 or A_2 or B_1 or B_2), and it specifies the degree to which the given input x (or y) satisfies the qualifier A. Here the membership function for A can be any appropriate parameterized membership function. Parameters in this layer are referred to as *premise parameters.*

Layer 2:

Every node in this layer is a fixed node labeled Π, whose output is the product of all the incoming signals:

$$O_{2,i} = w_i = \mu_{Ai}(x)\ \mu_{Bi}(y), \quad i = 1,2.$$

Each node output represents the firing strength of a rule. In general, any other T-norm operators that perform fuzzy AND can be used as the node function in this layer.

Layer 3:

Every node in this layer is a fixed node labeled *N*. The i^{th} node calculates the ratio of the i^{th} rule's firing strength to the sum of all rules' firing strengths:

$$O_{3,i} = \overline{w}_i = \frac{w_i}{w_1 + w_2}, \quad i = 1, 2$$

The outputs of this layer are called *normalized firing strengths.*

Layer4:

Every node *i* in this layer is an adaptive node with a node function

$$O_{4,i} = \overline{w}_i f_i = \overline{w}_i (p_i x + q_i y + r_i)$$

Where $\overline{w}_i$ is a normalized firing strength from layer 3 and $\{p_i, q_i, r_i\}$ is the parameter set of this node. Parameters in this layer are referred to as *consequent parameters.*

Layer5:

The single node in this layer is a fixed node labeled Σ that computes the overall outputs as the summation of all incoming signals,

$$\text{Overall output} = O_{5,i} = \Sigma_i \overline{w}_i f_i = \frac{\Sigma_i w_i f_i}{\Sigma_i w_i}$$

The above constructed adaptive network architecture represents a functionally equivalent *Sugeno* fuzzy model.

8.3.2 Gradient Based Learning Procedure for ANFIS

As discussed in section 8.3.1, ANFIS is a network structure consisting of nodes and directional links through which the nodes are connected. Moreover, part or all of the nodes are adaptive, which means their outputs depend on the parameters(s) pertaining to these nodes. It is a multilayer feed forward network in which each node performs a particular function on incoming signal as well as a set of parameters pertaining to this node. The formulas for node function may vary from node to node, and the choice of each node function depends on the overall input-output function. Here the adaptive network only indicates the flow direction of the signal between nodes; no weights are associated with the links. The parameter set of an adaptive network is the union of the parameter sets of each adaptive node. The basic learning rule of an adaptive network is based on the gradient decent and chain rule proposed by Werbos [Werbos.P, 1974]. In the present study we used the modified gradient based learning procedure derived by Jang [Jang.J.-S.R, 1985]. This learning procedure updates the parameter set according to the given training data in order to achieve a desired input-output mapping.

8.3.3 ANFIS Implementation for Handwritten Character Recognition

The implementation details of the ANFIS architecture and gradient based learning procedure used in the present study are described in this section. The i^{th} fuzzy *if.....then* rules framed from the FZ-NVD feature can be expressed as,

$$\textit{If } x_1 \text{ is } A_{i,1} \text{ and } x_2 \text{ is } A_{i,2},\ldots, x_n \text{ is } A_{i,n} \textit{ then}$$

$$f_i = (q_{i,1}\, x_1 + q_{i,2}\, x_2 + \ldots + q_{i,n}\, x_n + q_{i,n+1})$$

where $x_1, x_2, \ldots, x_n$ are input feature values, $A_{i,1},\ldots, A_{i,n}$ are any one of the membership function corresponding to the fuzzy sets SMALL, MEDIUM or LARGE and $\{ q_{i,1}, q_{i,2},\ldots, q_{i,n}, q_{i,n+1}\}$ is the parameter set. Here the size of the feature vector n = 9. The detailed description of the processes performed on each layers of the network architecture is given below.

Layer 1:

The feature values $x_1, x_2,\ldots, x_9$ are given as input to the network. At first, each of these feature values is given to the three *Gaussian* membership functions designed to represent the fuzzy sets SMALL, MEDIUM and LARGE. The general form of the membership function is given by,

$$\mu_{i,j}(x_i) = e^{-(x_i - c_{ij})^2/(2\sigma_{ij}^2)}$$

for $i = 1$ to n and

$j = 1$ to 3

where c_{ij} and σ_{ij} are the constant parameters of the i^{th} attribute of the j^{th} membership function. The values of c for the membership functions SMALL, MEDIUM and LARGE are fixed as 0.25, 0.5 and 0.75 respectively while that of width σ is fixed as 0.1 for all the cases as discussed in section 8.2.1. Output of each node in this layer is the membership value acquired while passing the input feature values to the above defined *Gaussian* membership functions corresponding to SMALL, MEDIUM and LARGE.

Layer 2:

The number of nodes in this layer is fixed as equal to the number of ID3 derived rules. Here each node represents an ID3 derived fuzzy rule. The input to each of the nodes in this layer is controlled by the attributes of the fuzzy rule (representing the node). That is if the i^{th} attribute of the rule representing the j^{th} node in this layer is X (X can take any of the values SMALL, MEDIUM or LARGE) then the corresponding membership value (representing the fuzzy set X) from the i^{th} node in the layer one is given to the j^{th} node of this layer. For example, if the first attribute of the rule representing the first node in this layer is SMALL then the membership value of the fuzzy set SMALL from the first node of the layer one is given as input to the first node in this layer. And if the second attribute of the rule is LARGE then the membership value of the fuzzy set LARGE from the second node of the layer one is given as input to the second node of this layer. The same process is repeated for all the attributes of the rule, and the product of all the incoming values gives the final node value. This input selection method for nodes in layer 2 is illustrated in figure 8.5 with the continuous lines representing the established connections. This proposed method incorporates the information from the ID3 driven rule set to the ANFIS and it will definitely help in partitioning the input-space in a fast and accurate way. Thus for each node in this layer compute,

$$w_i = \prod_{j=1}^{n} e^{-(x_j - c_{ij})^2/(2\sigma_{ij}^2)}$$

for $i = 1$ to N and

$j = 1$ to n

where N is the number of nodes in the layer 2 (number of ID3 derived rules) and n is the size of the feature vector (number of attributes in each rule). Here each node output represents the firing strength of a rule.

Layer 3:

In this layer, for each node, the ratio of the i^{th} rule's firing strength to the sum of all rules firing strengths (normalized firing strength) is computed using the expression,

$$\overline{w_i} = \frac{\prod_{j=1}^{n} e^{-(x_j - c_{ij})^2/(2\sigma_{ij}^2)}}{\sum_{i=1}^{N}\prod_{j=1}^{n} e^{-(x_j - c_{ij})^2/(2\sigma_{ij}^2)}}$$

Layer4:

Every node i in this layer is an adaptive node with a node function

$$O_{4,i} = \overline{w}_i f_i = \overline{w}_i (q_{i,1}\, x_1 + q_{i,2}\, x_2 + \ldots + q_{i,n}\, x_n + q_{i,n+1})$$

Where $\overline{w}_i$ is a normalized firing strength from layer 3 and $\{q_{i,1, \ldots}, q_{i,n+1}\}$ is the parameter set of this node. Parameters in this layer are referred to as *consequent parameters.*

Layer 5:

The single node in this layer is a fixed node labeled $\sum$ that computes the overall outputs as the summation of all incoming signals,

$$z = \sum_{i=1}^{N} \overline{w_i} f_i$$

where $f_i = (q_{i,1}\, x_1 + q_{i,2}\, x_2 + \ldots + q_{i,n}\, x_n + q_{i,n+1})$, $\overline{w_i}$ is the normalized firing strength of w_i, and N is the total number of rules.

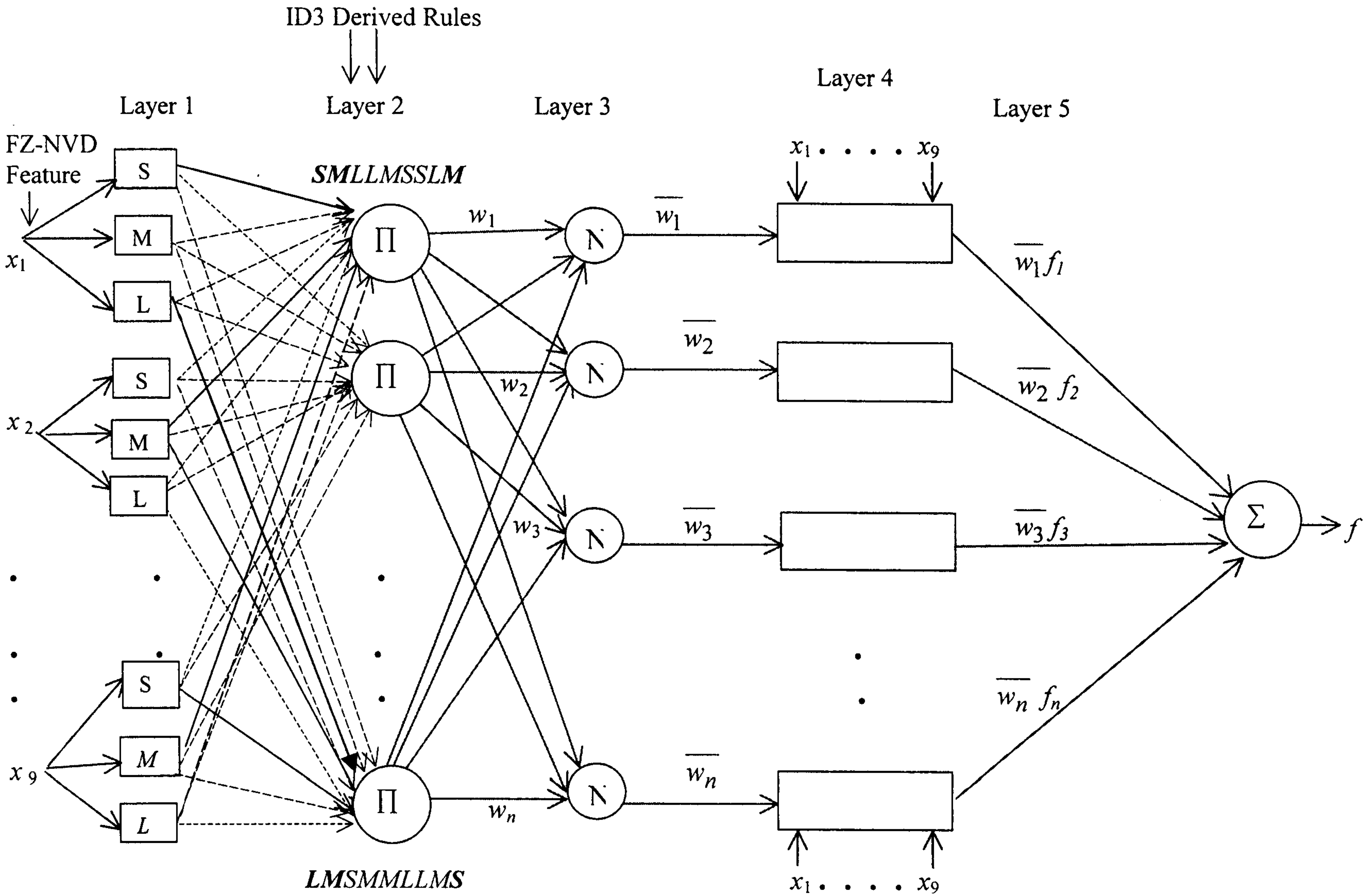


Figure 8.5 ANFIS architecture for Malayalam handwritten character recognition
(Continuous lines represent the established connections)

Above network is trained using the gradient based learning procedure (in Bach/off-line mode) proposed by Jang J.-S.R. as discussed in section 8.3.2. This learning process tries to minimize the error between the actual and the target output using the error function,

$$E = \sum (\text{Actual output} - \text{Target output})^2$$

$$= \sum_{p=1}^{N} (z_p - d_p)^2$$

where z_p denotes the output of the p^{th} training data, d_p denotes the target output and N is the total number of training data. The learning procedure with the error function E is implemented to achieve the desired input-output mapping, by updating the antecedent parameters according to the given training data. If the training data set has p samples, then the input data is considered as a matrix of size p x n where n is the size of the feature vector or the number of attributes of the fuzzy rule framed from it. A membership function matrix is of size N x n, where N is the total number of ID3 derived rules and n is the number of attributes in each rule. The error correction expressions for the m^{th} iteration with respect to c_{ij} and σ_{ij} are given by,

$$c_{ij}(m) = c_{ij}(m\text{-}1) - \eta_1 \frac{\partial E(m)}{\partial c_{ij}}$$

$$\sigma_{ij}(m) = \sigma_{ij}(m\text{-}1) - \eta_2 \frac{\partial E(m)}{\partial \sigma_{ij}}$$

$$q_{ij}(m) = q_{ij}(m\text{-}1) - \eta_3 \frac{\partial E(m)}{\partial q_{ij}}$$

where η_1, η_2 and η_3 are learning parameters which are constant throughout the learning procedure. To find the gradient-decent we, differentiate E with respect to c_{ij}, σ_{ij} and q_{ij}, to get,

$$\frac{\partial E(m)}{\partial c_{ij}} = \frac{\partial E(m)}{\partial e}\frac{\partial e(m)}{\partial \overline{w}_i}\frac{\partial \overline{w}_i(m)}{\partial c_{ij}}$$

i.e., $$\frac{\partial E(m)}{\partial c_{ij}} = 2\sum_p (z_p - d_p)\overline{w}_i\left(f_i - \sum_i f_i w_i\right)\left(\frac{x_j - c_{ij}}{\sigma_{ij}^2}\right)$$

$$\frac{\partial E(m)}{\partial \sigma_{ij}} = \frac{\partial E(m)}{\partial e}\frac{\partial e(m)}{\partial \overline{w}_i}\frac{\partial \overline{w}_i(m)}{\partial \sigma_{ij}}$$

i.e., $$\frac{\partial E(m)}{\partial \sigma_{ij}} = 2\sum_p (z_p - d_p)\overline{w}_i\left(f_i - \sum_i f_i w_i\right)\left(\frac{(x_j - c_{ij})^2}{\sigma_{ij}^3}\right)$$

$$\frac{\partial E(m)}{\partial q_{ij}} = \frac{\partial E(m)}{\partial e}\frac{\partial e(m)}{\partial q_{ij}}$$

i.e., $$\frac{\partial E(m)}{\partial q_{ij}} = 2\sum_p (z_p - d_p)\overline{w}_i\, x_j$$

The following section describes with the simulation experiment conducted for the recognition of Malayalam handwritten characters using ANFIS and presents the main results.

8.3.3.1 Simulation Experiments and Results

Simulation experiment is conducted to implement the ANFIS algorithm in MATLAB. In chapter 7, it has been experimentally shown that FZ-NVD parameter gives better recognition results in all the three recognition algorithms used, viz., c-Means clustering technique, k_NN classifier and class-modular neural network. Therefore the credibility of the ANFIS based character recognition system developed in this chapter is verified using the FZ-NVD parameters. In training, the error tolerance is fixes as 0.01 and learning constant η is fixed as 0.1. After the convergence the final parameter set is stored in a data file and further used in the testing phase. In this work we used a set of 15,752 samples of forty four Malayalam handwritten characters taken from 358 different writers for training and a disjoint sample dataset of same size is used for testing.

The Confusion Matrix obtained as the result of the recognition experiment conducted using FZ-NVD parameters and ANFIS architecture with gradient based learning is given in table 8.1(a-d). The average recognition accuracies of forty four Malayalam characters are given in Table 8.2. The overall recognition accuracy obtained is 82.35 %. Table 8.3 shows the average recognition accuracies obtained in various experiments conducted in this work based on different parameters and recognition schemes. From this table it is clear that ANFIS with FZ-NVD parameters gives better results compared to other recognition algorithms.

Characters		അ [-ah]	ആ [-aah]	ഇ [-yi]	ഉ [-uh]	ഋ [-eru]	എ [-eh]	ഏ [-aeh]	ഒ [-oh]	ക [ka]	ഖ [kha]	ഗ [ga]	ഘ [gha]	ങ [nga]	ച [cha]	ഛ [chha]	ജ [ja]	ഝ [jha]	ഞ [nja]	ട [ta]	ഠ [tta]	ഡ [da]	ഢ [dda]
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
അ [-ah]	1	353	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	1	0	0	0	0
ആ [-aah]	2	0	291	0	0	0	0	0	0	0	0	0	0	0	0	18	11	0	2	0	0	0	18
ഇ [-yi]	3	0	0	318	0	0	0	0	0	0	0	0	0	0	0	0	12	0	0	0	0	0	0
ഉ [-uh]	4	0	0	32	284	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ഋ [-eru]	5	0	0	0	0	280	0	0	0	0	0	0	0	0	0	27	0	0	0	0	0	0	0
എ [-eh]	6	0	0	0	0	0	342	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ഏ [-aeh]	7	0	0	0	0	0	18	316	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ഒ [-oh]	8	0	0	0	0	0	0	0	330	0	0	0	0	0	0	0	8	0	0	0	0	0	0
ക [ka]	9	0	0	0	0	0	0	0	38	291	0	0	0	0	0	0	0	0	0	0	0	0	0
ഖ [kha]	10	9	0	0	0	0	7	0	18	6	299	2	0	0	0	0	0	0	0	8	0	0	0
ഗ [ga]	11	0	0	0	0	0	0	0	0	0	0	341	0	0	0	0	0	0	0	0	0	5	0
ഘ [gha]	12	0	0	0	0	0	0	0	0	0	0	0	281	0	0	0	0	0	0	0	0	58	19
ങ [nga]	13	0	0	0	0	0	2	0	0	0	0	0	0	307	0	0	0	0	17	0	0	0	0
ച [cha]	14	0	0	0	0	0	0	0	0	0	0	0	0	0	312	32	0	0	0	0	0	0	0
ഛ [chha]	15	28	0	0	0	0	0	0	0	0	0	0	0	0	0	288	0	0	0	0	0	0	0
ജ [ja]	16	26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	277	0	0	0	0	0	0
ഝ [jha]	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	327	0	0	0	0	0
ഞ [nja]	18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	321	0	0	0	0
ട [ta]	19	0	0	0	42	0	0	0	0	0	0	0	0	0	0	0	0	0	0	262	0	0	0
ഠ [tta]	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	78	269	0	0
ഡ [da]	21	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	342	6
ഢ [dda]	22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	45	313

Table 8.1 (a) Confusion Matrix (Continued)

Characters		ണ[nha]	ത[tha]	ഥ[thha]	ദ[dha]	ധ[dhha]	ന[na]	പ[pa]	ഫ[pha]	ബ[ba]	ഭ[bha]	മ[ma]	യ[ya]	ര[ra]	ല[la]	വ[va]	ശ[sha]	ഷ[shha]	സ[sa]	ഹ[ha]	ള[lha]	ഴ[zha]	റ[rha]
		23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44
അ [-ah]	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
ആ[-aah]	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	18	0	0	0	0	0
ഇ [-yi]	3	0	0	0	28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ഉ [-uh]	4	0	0	12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	30	0	0
ഋ [-eru]	5	0	0	0	0	0	0	0	0	0	0	0	0	0	23	0	0	0	0	0	0	28	0
എ [-eh]	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ഏ[-aeh]	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	9	0
ഒ [-oh]	8	3	0	0	0	0	0	0	0	0	17	0	0	0	0	0	0	0	0	0	0	0	0
ക [ka]	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	29	0	0	0	0	0
ഖ [kha]	10	3	0	2	0	0	1	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0
ഗ [ga]	11	0	0	8	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ഘ [gha]	12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ങ [nga]	13	0	12	0	0	0	11	0	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0
ച [cha]	14	0	0	0	0	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ഛ[chha]	15	0	0	9	0	0	0	0	0	14	0	0	0	0	19	0	0	0	0	0	0	0	0
ജ [ja]	16	23	0	0	0	0	0	0	0	0	0	0	32	0	0	0	0	0	0	0	0	0	0
ഝ[jha]	17	0	9	0	0	0	11	0	0	0	0	0	7	0	0	4	0	0	0	0	0	0	0
ഞ [nja]	18	27	0	0	0	0	6	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0
ട [ta]	19	0	0	0	42	0	0	0	0	0	12	0	0	0	0	0	0	0	0	0	0	0	0
ഠ[tta]	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11
ഡ [da]	21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ഢ [dda]	22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8.1 (b) Confusion Matrix (Continued)

Characters		അ [-ah]	ആ[-aah]	ഇ [-yi]	ഉ [-uh]	ഋ [-eru]	എ [-eh]	ഏ[-aeh]	ഒ [-oh]	ക [ka]	ഖ [kha]	ഗ [ga]	ഘ [gha]	ങ [nga]	ച [cha]	ഛ[chha]	ജ [ja]	ഝ[jha]	ഞ [nja]	ട [ta]	ഠ[tta]	ഡ [da]	ഢ [dda]
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
ണ[nha]	23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ത [tha]	24	0	0	0	0	0	0	0	0	0	0	0	0	19	0	0	0	4	6	0	0	0	0
ഥ [thha]	25	0	0	0	0	0	0	0	0	0	0	34	0	0	0	0	0	0	0	0	0	0	0
ദ [dha]	26	0	0	0	0	0	0	0	30	7	0	0	0	0	0	0	0	0	0	8	0	0	0
ധ[dhha]	27	0	0	0	0	0	0	0	0	0	0	38	0	0	0	0	0	0	0	0	0	0	0
ന [na]	28	0	0	0	0	0	0	0	0	0	0	5	0	7	0	0	0	0	0	0	0	0	0
പ [pa]	29	0	0	0	0	0	0	0	0	0	0	0	0	0	42	0	0	0	0	0	0	0	0
ഫ [pha]	30	0	0	0	0	0	0	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ബ [ba]	31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	27	0	0	0	0
ഭ [bha]	32	0	0	0	0	0	0	0	24	0	0	0	0	0	0	0	0	0	0	0	0	0	0
മ [ma]	33	0	0	0	0	0	0	0	52	0	0	0	0	0	0	0	0	0	0	0	0	0	0
യ [ya]	34	28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16	0	0	0	0	0
ര [ra]	35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ല [la]	36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
വ [va]	37	0	0	0	0	0	0	0	1	0	5	3	0	0	0	0	0	1	0	2	0	0	0
ശ [sha]	38	0	0	0	0	0	0	0	0	0	0	37	0	0	0	0	10	0	0	0	0	0	0
ഷ[shha]	39	61	0	0	0	0	0	0	0	42	0	0	0	0	0	0	0	0	0	0	0	0	0
സ [sa]	40	0	0	0	0	0	0	0	0	0	0	0	0	0	51	0	0	0	42	0	0	0	0
ഹ [ha]	41	0	0	0	0	0	18	0	0	0	0	0	0	0	0	22	0	0	0	0	0	0	0
ള [lha]	42	0	0	72	11	0	0	0	0	0	0	0	0	0	0	0	31	0	0	0	0	0	0
ഴ [zha]	43	0	0	0	0	69	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	49
റ [rha]	44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	104	0	0

Table 8.1 (c) Confusion Matrix (Continued)

Characters		ണ [nha]	ത [tha]	ഥ [thha]	ദ [dha]	ധ [dhha]	ന [na]	പ [pa]	ഫ [pha]	ബ [ba]	ഭ [bha]	മ [ma]	യ [ya]	ര [ra]	ല [la]	വ [va]	ശ [sha]	ഷ [shha]	സ [sa]	ഹ [ha]	ള [lha]	ഴ [zha]	റ [rha]
		23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44
ണ[nha]	23	319	39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ത [tha]	24	13	279	0	0	0	0	0	0	0	0	0	21	16	0	0	0	0	0	0	0	0	0
ഥ [thha]	25	0	0	324	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ദ [dha]	26	0	0	0	313	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ധ[dhha]	27	0	0	0	0	320	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ന [na]	28	0	27	0	0	0	317	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0
പ [pa]	29	0	0	0	0	0	0	259	12	0	0	0	0	0	0	38	0	0	0	7	0	0	0
ഫ [pha]	30	0	0	0	0	0	0	0	271	0	8	0	0	0	0	0	0	0	0	62	0	0	0
ബ [ba]	31	19	0	0	0	0	0	0	0	283	0	0	0	0	0	0	0	0	29	0	0	0	0
ഭ [bha]	32	0	0	0	0	0	0	0	0	0	315	0	19	0	0	0	0	0	0	0	0	0	0
മ [ma]	33	0	0	44	0	0	0	0	0	0	0	262	0	0	0	0	0	0	0	0	0	0	0
യ [ya]	34	0	0	0	0	0	0	0	0	0	41	0	273	0	0	0	0	0	0	0	0	0	0
ര [ra]	35	12	24	0	0	0	29	0	0	0	0	0	10	283	0	0	0	0	0	0	0	0	0
ല [la]	36	0	0	47	0	0	0	0	0	0	0	59	0	0	252	0	0	0	0	0	0	0	0
വ [va]	37	4	0	0	0	0	1	0	0	0	2	1	0	0	0	338	0	0	0	0	0	0	0
ശ [sha]	38	0	0	18	17	0	0	0	0	0	13	0	0	0	0	0	263	0	0	0	0	0	0
ഷ[shha]	39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	255	0	0	0	0	0
സ [sa]	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	265	0	0	0	0
ഹ [ha]	41	0	0	0	0	0	0	0	24	0	0	0	0	0	0	0	0	0	0	294	0	0	0
ള [lha]	42	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	244	0	0
ഴ [zha]	43	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	240	0
റ [rha]	44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	254

Table 8.1 (d)

Table 8.1 (a-d): The confusion matrix indicating the recognition of forty four Malayalam handwritten characters using ANFIS architecture

Character Number	Character	Recognition accuracy (%)	Character Number	Character	Recognition accuracy (%)
1	അ [-ah]	98.60	23	ണ[nha]	89.11
2	ആ[-aah]	81.29	24	ത [tha]	77.93
3	ഇ [-yi]	88.83	25	ഥ [thha]	90.50
4	ഉ [-uh]	79.33	26	ദ [dha]	87.43
5	ഋ [-eru]	78.21	27	ധ[dhha]	89.38
6	എ [-eh]	95.53	28	ന [na]	88.55
7	ഏ[-aeh]	88.27	29	പ [pa]	72.35
8	ഒ [-oh]	92.17	30	ഫ [pha]	75.69
9	ക [ka]	81.28	31	ബ [ba]	79.05
10	ഖ [kha]	83.52	32	ഭ [bha]	87.99
11	ഗ [ga]	95.25	33	മ [ma]	73.18
12	ഘ [gha]	78.49	34	യ [ya]	76.26
13	ങ [nga]	85.75	35	ര [ra]	79.05
14	ച [cha]	87.15	36	ല [la]	70.39
15	ഛ[chha]	81.45	37	വ [va]	94.41
16	ജ [ja]	77.37	38	ശ [sha]	73.46
17	ഝ[jha]	91.34	39	ഷ[shha]	71.23
18	ഞ [nja]	89.97	40	സ [sa]	74.02
19	ട [ta]	73.18	41	ഹ [ha]	82.12
20	ഠ[tta]	75.14	42	ള [lha]	68.16
21	ഡ [da]	95.53	43	ഴ [zha]	67.04
22	ഢ [dda]	87.43	44	റ [rha]	70.95
Average Recognition Accuracy (%)					82.34841

Table 8.2: The overall recognition accuracies of forty four Malayalam handwritten characters based on FZ-NVD features and ANFIS architecture

Sl. No	Classifiers	Recognition Accuracy (%)		
		SSPD Feature	Z-VD Feature	FZ-NVD Feature
1	*c*-Means clustering	64.15	67.43	69.00
2	*k*-NN classifier	71.61	73.10	75.82
3	Class-modular Neural Network	73.03	75.58	78.87
4	ANFIS	------	------	82.35

Table 8.3 Average recognition accuracies of forty four Malayalam handwritten characters based on different features and classifiers

8.4 Conclusions

A handwritten character recognition system using ANFIS is designed and developed. A fuzzy-rule set is framed from the fuzzy-zoned normalized vector distance (FZ-NVD) features extracted from handwritten characters using *Gaussian* Membership Functions defined for the fuzzy sets LARGE, MEDIUM and SMALL. To perform the parameter adjustment in learning, ANFIS need partitioning of the input space. In order to facilitate this process we used a decision tree induction procedure named ID3. The entire fuzzy rules framed from the feature vectors are given as input to this ID3 algorithm and the reduced optimum rule set obtained as the output of this process is then used as input to the inference part of ANFIS architecture. The Network is trained with the help of a gradient based learning rule using a set of 15,752 samples of the forty four Malayalam handwritten characters taken from 358 different writers and a disjoint sample dataset of same size is used for testing. The entire system is implemented using MATLAB codes. This work is the first of its kind in Malayalam and the overall recognition accuracy obtained for the forty four basic Malayalam handwritten characters is 82.35%. This is highly promising compared to the recognition results obtained in various experiments conducted in this study using different parameters and classifiers.

CHAPTER 9

CONCLUSION

A well-structured and standard database of isolated Malayalam handwritten character images is developed in this work. The constructed database has the potential to be used as a benchmarking resource for the handwritten character recognition research. Besides, this work is the first of its kind in Malayalam.

Different pre-processing algorithms capable of preparing the original gray-scale character images in the database for feature extraction studies are implemented. Initially, the foreground extraction from the gray-scale character image samples is performed with Otsu's global thresholding technique. To bring uniformity among the character patterns, the image samples are normalized in to 66 x 42 pixel size using Affine Transformation and Bilinear Interpolation algorithm. A modified version of rotation invariant rule based thinning algorithm proposed by Ahmed.M. and Ward.R. is applied on character images to acquire one pixel wide skeletons of the character patterns.

A new method for modeling Malayalam handwritten characters based on the gray-scale images is introduced. Reconstructed state-spaces in the form of State-Space Maps (SSMs) for character images are generated for one, four and eight directional space variations. SSMs with eight directional space variations are found to be well informed and hence it is used further in the feature extraction process. Sate-Space Point Distribution (SSPD) features are extracted from the SSMs of character samples. The extracted parameters are found promising and used further in the recognition experiments.

The global characteristics of Malayalam handwritten character patterns are analyzed based on the aspect ratio, existence ratio and the centroid of the character curve. The experimental results indicate that it is extremely difficult to classify the character patterns based on the global features since these features considerably deviate from the norm. Therefore the feature extraction studies in designing parameters that are capable of representing the information content regions in a character pattern based on their parts are conducted. As a result character image patterns are modeled based on the region decomposition or zoning.

Two methods for handwritten character modeling using perceptual zoning of the pre-processed character images and the vector distance as the parameter measure are proposed and evaluated. At first a zoned vector distance (Z-VD) feature is extracted and these features are then fine tuned with the fuzzy-zoned normalized vector distance (FZ-NVD) measures. It is found that the distribution patterns obtained from these features are almost similar for repeated samples of the same character written by different writers and varies from character to character. These parameters are effectively employed to model Malayalam handwritten characters and further utilized for recognition purpose.

Handwritten character recognition studies based on the SSPD features extracted from the grayscale images, Z-VD and FZ-NVD parameters extracted from the pre-processed character images using different classifiers are conducted. The cluster analysis using the c-Means clustering technique is performed on the extracted feature vectors and the same technique is used for the recognition also. The credibility of the extracted parameters is also tested with *k*-Nearest Neighbour (*k*-NN) classifier. A connectionist model based recognition system by means of class-modular neural network is implemented and tested using above three parameters. The recognition rates obtained using the FZ-NVD features are found to be more promising and it is further used for developing a recognition scheme with the help of Adaptive-Network-Based Fuzzy Inference System (ANFIS) to achieve better performance.

A Handwritten character recognition system using ANFIS is designed and developed. For this purpose, a fuzzy-rule set is framed from the fuzzy-zoned normalized vector distance (FZ-NVD) features using Gaussian membership functions defined for the fuzzy sets LARGE, MEDIUM and SMALL. Here we used Sugeno fuzzy model to design the inference part of ANFIS. For executing the parameter adjustment in learning procedure, the ANFIS needs the input data space to be divided according to the target classes. In order to facilitate this operation we used a decision tree induction procedure named ID3 proposed by Quinlan. The fuzzy rules framed from the feature vectors are given as input to this ID3 algorithm and the final rule set obtained as the output of this process is then used as input to the inference part of ANFIS architecture.

The Network is trained with the help of a gradient based learning rule using a set of 15,752 samples of the forty four Malayalam handwritten characters taken from 358 different writers and a disjoint sample dataset of same size is used for testing. The entire system is implemented using MATLAB codes. This work is a first of its kind in Malayalam and the overall recognition accuracy obtained for the forty four basic Malayalam handwritten characters is 82.35%. Some of the recent handwritten character research findings reported for Indian languages are given in Table 9.1 for comparison.

Sl. No	Language and number of classes	Method	Feature Vector Dimension	Recognition Accuracy (%)
1	Bangla (49 class)	Rahman.AFR *et al*, 2002	High level features (multistage- coarse processing required)	88.38
2	Bangla (50 class)	Bhowmik.T.K *et al.*, 2004	200 100	84.33 80.58
3	Devanagari (25 class)	Arora.S. *et al.*, 2006	25	88.20
4	Hindi (36 class)	Hanmandlu.M. *et al.*, 2006	24	85.00
5	Kannada (48 class)	Pal.U *et al.*, 2006	400 64	85.71 80.87
6	Malayalam (44 class)	Work presented in this thesis	9	82.35

Table 9.1 Recent contributions to HCR in Indian languages

In the present study, the proposed ANFIS based recognition system uses nine dimensional FZ-NVD features for recognizing forty four Malayalam handwritten characters. Comparing the feature vector size and the number of character classes used in each work, it is evident that the proposed method outperforms other techniques and can be effectively implemented as a high speed HCR system for Malayalam.

One of the future research directions is to design the parameters for modeling compound characters with special symbols and modifiers. The out comes of this research can also be exploited for handwritten word recognition with an intelligent character segmentation algorithm. Hence, the system methodology developed in this study using ANFIS can be extended in developing a full-fledged HCR.

Most of the systems often rely on a lexicon during the recognition stage. Fact is that, whatever be the recognition strategy, contextual knowledge (linguistic, domain or any other pertinent information) needs to be incorporated into the recognition process to reduce the ambiguity. Lexicon is such a source of linguistic and domain knowledge. A new architecture, for incorporating lexicon based domain specific knowledge to the developed system, may also be investigated.

The ultimate goal of handwriting recognition research is to have machines that can read any handwritten text with the same recognition accuracy as human but at a faster rate. We hope a complete Malayalam HCR system will evolve in public domain with this work as the starting point.

APPENDIX - I

MATLAB implementation of State-Space Map and State-Space Point Distribution parameter extraction.

```
% This progam generate the state-space map with eight directional space
% variations for Malayalam handwritten character images of 66x42 pixel size.
clc;
imWidth=66;    %Image Width
imHeight=42;   %Image Height
for nDir=1:44
    for nSamples=1:358
        A=zeros([42 66]);
        X=zeros([1 2772]);
        Y1=zeros([1 2772]);
        Y2=zeros([1 2772]);
        Y3=zeros([1 2772]);
        Y4=zeros([1 2772]);
        Y5=zeros([1 2772]);
        Y6=zeros([1 2772]);
        Y7=zeros([1 2772]);
        Y8=zeros([1 2772]);
        C=zeros([42 66]);

im=double(imread(strcat('F:\1\',num2str(nDir),'\pSample\new1\bb',num2str(nSamp
les),'.bmp')));
        imTemp = padarray(im,[1 1],0,'both');
        Z=zeros([1,2688]);
        k=0;
        for i=2: 41
            for j=2:65
                k=k+1;Z(k)=im(i,j);

                Y1(k)=im(i-1,j);
                Y2(k)=im(i,j+1);
                Y3(k)=im(i+1,j);
                Y4(k)=im(i,j-1);
                Y5(k)=im(i-1,j-1);
                Y6(k)=im(i-1,j+1);
                Y7(k)=im(i+1,j+1);
                Y8(k)=im(i+1,j-1);

            end
        end
        figure

plot(Z,Y1,'k.',Z,Y2,'k.',Z,Y3,'k.',Z,Y4,'k.',Z,Y5,'k.',Z,Y6,'k.',Z,Y7,'k.',Z,Y
8,'k.','markersize',3);
        axis([0 300 0 300]);
        active.BackgroundColor = [1 0 0];
        active.Enable = 'on';
        active.ForegroundColor = [1 0 0];
    end
end
```

```
% This progam will generate state-space point distribution graph for
% handwritten character imges of size 66x42 with eight directional sapce
variations

clc;
imWidth=66;    %Image Width
imHeight=42;   %Image Height

for nDir=1:44
    for nSamples=1: 358
        A=zeros([16 16]);
        X=zeros([1 2772]);
        Y1=zeros([1 2772]);
        Y2=zeros([1 2772]);
        Y3=zeros([1 2772]);
        Y4=zeros([1 2772]);
        Y5=zeros([1 2772]);
        Y6=zeros([1 2772]);
        Y7=zeros([1 2772]);
        Y8=zeros([1 2772]);
        Z=zeros([1 2772]);
        C=zeros([42 66]);

im=imread(strcat('F:\1\',num2str(nDir),'\pSample\new1\bb',num2str(nSamples),'.
bmp'));
        X=reshape(im,[1 2772]);
        imTemp = padarray(im,[1 1],0,'both');
        k=0;
        for i=2: 41
            for j=2:65
                k=k+1;Z(k)=im(i,j);
                Y1(k)=im(i-1,j);
                Y2(k)=im(i,j+1);
                Y3(k)=im(i+1,j);
                Y4(k)=im(i,j-1);
                Y5(k)=im(i-1,j-1);
                Y6(k)=im(i-1,j+1);
                Y7(k)=im(i+1,j+1);
                Y8(k)=im(i+1,j-1);

            end
        end

        for i=1: 2764
            [a,b]=cordVal(Z(i),Y1(i),16,16);
            A(a,b)=A(a,b)+1;
            [a,b]=cordVal(Z(i),Y2(i),16,16);
            A(a,b)=A(a,b)+1;
            [a,b]=cordVal(Z(i),Y3(i),16,16);
            A(a,b)=A(a,b)+1;
            [a,b]=cordVal(Z(i),Y4(i),16,16);
            A(a,b)=A(a,b)+1;
            [a,b]=cordVal(Z(i),Y5(i),16,16);
            A(a,b)=A(a,b)+1;
            [a,b]=cordVal(Z(i),Y6(i),16,16);
            A(a,b)=A(a,b)+1;
            [a,b]=cordVal(Z(i),Y7(i),16,16);
            A(a,b)=A(a,b)+1;
            [a,b]=cordVal(Z(i),Y8(i),16,16);
            A(a,b)=A(a,b)+1;
        end

        for i=1:16
            for j=1:16
                B((16-(i-1)),j)=A(i,j);
            end
        end

        figure;
        C=reshape(B',256,1);
        disp(C);
        D=1:256;
        plot(D(:),C(:),'r.-','markersize',1);
    end
end

function [xVal, yVal]=CordVal(iVal,jVal,ht,wd)
xVal=floor(iVal/ht)+1;
yVal=floor(jVal/wd)+1;
```

```
%This progam extarcts state-space point distribution parameters for Malayalam
%handwritten character image of size 66x42 pixels with eight directional sapce
%variations


clc;
imWidth=66;     %Image Width
imHeight=42;    %Image Height
for nDir=1:44
    clear B;
    clear C;
    clear D;
    fp1=fopen(strcat('F:\1\sspd\ttnew8nnew',num2str(nDir),'.dat'),'w');
    for nSamples=1:358
        A=zeros([16 16]);
        B=zeros([16 16]);
        C=zeros([16 16]);
        X=zeros([1 2772]);
        Y1=zeros([1 2772]);
        Y2=zeros([1 2772]);
        Y3=zeros([1 2772]);
        Y4=zeros([1 2772]);
        Y5=zeros([1 2772]);
        Y6=zeros([1 2772]);
        Y7=zeros([1 2772]);
        Y8=zeros([1 2772])
        D=zeros([1 66]);

im=double(imread(strcat('F:\1\',num2str(nDir),'\pSample\new1\bb',num2str(nSamp
les),'.bmp')));
        X=reshape(im,[1 2772]);
        imTemp = padarray(im,[1 1],0,'both');
        k=0;m=0;n=0;
        for i=2: 41
            for j=2:65
                k=k+1;
                Z(k)=im(i,j);
                Y1(k)=im(i-1,j);
                Y2(k)=im(i,j+1);
                Y3(k)=im(i+1,j);
                Y4(k)=im(i,j-1);
                Y5(k)=im(i-1,j-1);
                Y6(k)=im(i-1,j+1);
                Y7(k)=im(i+1,j+1);
                Y8(k)=im(i+1,j-1);
            end
        end
        for i=1:2764
            [a,b]=cordVal(Z(i),Y1(i),16,16);
            A(a,b)=A(a,b)+1;
            [a,b]=cordVal(Z(i),Y2(i),16,16);
            A(a,b)=A(a,b)+1;
            [a,b]=cordVal(Z(i),Y3(i),16,16);
            A(a,b)=A(a,b)+1;
            [a,b]=cordVal(Z(i),Y4(i),16,16);
            A(a,b)=A(a,b)+1;
            [a,b]=cordVal(Z(i),Y5(i),16,16);
            A(a,b)=A(a,b)+1;
            [a,b]=cordVal(Z(i),Y6(i),16,16);
            A(a,b)=A(a,b)+1;
            [a,b]=cordVal(Z(i),Y7(i),16,16);
            A(a,b)=A(a,b)+1;
            [a,b]=cordVal(Z(i),Y8(i),16,16);
            A(a,b)=A(a,b)+1;
        end
        for i=1:16
            for j=1:16
                B((16 -(i-1)),j)=A(i,j);
            end
        end
        C=B;
        D=mean(C,2);
        for i=1:16
            fprintf(fp1,'%f ',D(i));
        end
        fprintf(fp1,'\n');
    end
    fclose(fp1);
end
```

APPENDIX - II

MATLAB implementation of fuzzy-zone based feature extraction technique.

```
% This program finds the fuzzy-zoning based Normalised Vector Distance feature
% for the forty-four Mlayalam handwritten characters.The nine fuzzy-region
% descripton is shown in the table given below.

%-----------------------------------------------
%          | A    B   C    D   E   F   G     H   I
% ---------|-------------------------------------
%    hRg   | 36  36  36   73  73  73  100   100 100
%    wRg   | 36  73  100  36  73  100  36   73  100
%    iRg   | 1   1   1    27  27  27   64   64   64
%    jRg   | 1   27  64   1   27  64   1    27   64
% -----------------------------------------------
%
%
clear all;
clc;
imHeight=42;
imWidth=66;
A=zeros([42 66]);
for nDir=1:44
    fp=fopen(strcat('F:\1\data2\fuzzy-zone\fz',num2str(nDir),'.dat'),'w');
    for nSamples=1:15

X=double(imread(strcat('F:\1\data2\',num2str(nDir),'\t',num2str(nSamples),'.bmp')));
        for i=1:42
            for j=2:65
                A(i,j)=X(i,j-1);
            end
        end
        %First Region - Region A
        hRg=17;      %imHeight*0.36;%region total width(end val)
        wRg=25;      %imWidth*0.36;%region total width(end val)
        iRg=1;       % Region i start
        jRg=1;       % Region j start
        Dist=0;
        cNum=0;
        avgDist=0;
        for i=iRg:hRg
            for j=jRg:wRg
                if(A(i,j)~=255)
                    fValue=getFuzzy1(i,j,imHeight,imWidth,1);
                    Dist=Dist+ sqrt((i-1)^2 +(j-1)^2)*fValue;
                    cNum=cNum+1;
                end
            end
        end
        if (Dist~=0)
            avgDist=Dist/cNum;
        end
        fprintf(fp,' %1f  ',avgDist);

        %2nd Region - Region B
        hRg=17;      %floor(imHeight*0.36);%region total width(end val)
        wRg=50;      %floor(imWidth*0.73);%region total width(end val)
        iRg=1;       % Region i start
        jRg=16;      %floor(0.27*imWidth); % Region j start
        Dist=0;
        cNum=0;
        avgDist=0;
        for i=iRg:hRg
            for j=jRg:wRg
                if(A(i,j)~=255)
                    fValue=getFuzzy1(i,j,imHeight,imWidth,2);
                    Dist=Dist+ sqrt((i-1)^2 +(j-1)^2)*fValue;
                    cNum=cNum+1;
                end
            end
```

```
end
if (Dist~=0)
    avgDist=Dist/cNum;
end
fprintf(fp,' %1f  ',avgDist);

%3rd Region- Region C
hRg=17;      %floor(imHeight*0.36);%region total width(end val)
wRg=66;      %floor(imWidth*1);%region total width(end val)
iRg=1;       % Region i start
jRg=50;      %floor(imWidth*0.64);% Region j start
Dist=0;
cNum=0;
Dist=0;
avgDist=0;
for i=iRg:hRg
    for j=jRg:wRg
        if(A(i,j)~=255)
            fValue=getFuzzy1(i,j,imHeight,imWidth,3);
            Dist=Dist+ sqrt((i-1)^2 +(j-1)^2)*fValue;
            cNum=cNum+1;
        end
    end
end
if (Dist~=0)
    avgDist=Dist/cNum;
end
fprintf(fp,' %1f   ',avgDist);

%4rd Region- Region D
hRg=34;      %floor(imHeight*0.73);%region total width(end val)
wRg=25;      %floor(imWidth*0.36);%region total width(end val)
iRg=17;      %floor(imHeight*0.27); % Region i start
jRg=1;       % Region j start
Dist=0;
cNum=0;
avgDist=0;
for i=iRg:hRg
    for j=jRg:wRg
        if(A(i,j)~=255)
            fValue=getFuzzy1(i,j,imHeight,imWidth,4);
            Dist=Dist+ sqrt((i-1)^2 +(j-1)^2)*fValue;
            cNum=cNum+1;
        end
    end
end
if (Dist~=0)
    avgDist=Dist/cNum;
end
fprintf(fp,' %1f   ',avgDist);

%5rd Region- Region E
hRg=32;      %floor(imHeight*0.73);%region total width(end val)
wRg=50;      %floor(imWidth*0.73);%region total width(end val)
iRg=17;      %floor(imHeight*0.27); % Region i start
jRg=25;      %floor(imWidth*0.27); % Region j start
Dist=0;
cNum=0;
avgDist=0;
for i=iRg:hRg
    for j=jRg:wRg
        if(A(i,j)~=255)
            fValue=getFuzzy1(i,j,imHeight,imWidth,5);
            Dist=Dist+ sqrt((i-1)^2 +(j-1)^2)*fValue;
            cNum=cNum+1;
        end
    end
end
if (Dist~=0)
    avgDist=Dist/cNum;
end
fprintf(fp,' %1f   ',avgDist);

%6rd Region- Region F
hRg=34;      %floor(imHeight*0.73);%region total width(end val)
wRg=66;      %1*imWidth;%region total width(end val)
iRg=17;      %floor(imHeight*0.27); % Region i start
jRg=41;      %floor(imWidth*0.64);% Region j start
Dist=0;
cNum=0;
avgDist=0;
```

```
        for i=iRg:hRg
            for j=jRg:wRg
                if(A(i,j)~=255)
                    fValue=getFuzzy1(i,j,imHeight,imWidth,6);
                    Dist=Dist+ sqrt((i-1)^2 +(j-1)^2)*fValue;
                    cNum=cNum+1;
                end
            end
        end
        if (Dist~=0)
            avgDist=Dist/cNum;
        end
        fprintf(fp,' %1f   ',avgDist);
        %7rd Region-Region G
        hRg=42;      %imHeight*1;     %region total width(end val)
        wRg=25;      %floor(imWidth*0.36);   %region total width(end val)
        iRg=34;      %floor(imHeight*0.64);  % Region i start
        jRg=1;       % Region j start
        Dist=0;
        cNum=0;
        avgDist=0;
        for i=iRg:hRg
            for j=jRg:wRg
                if(A(i,j)~=255)
                    fValue=getFuzzy1(i,j,imHeight,imWidth,7);
                    Dist=Dist+ sqrt((i-1)^2 +(j-1)^2)*fValue;
                    cNum=cNum+1;
                end
            end
        end
        if (Dist~=0)
            avgDist=Dist/cNum;
        end
        fprintf(fp,' %1f   ',avgDist);

        %8rd Region-Region H
        hRg=42;      %imHeight*1;     %region total width(end val)
        wRg=50;      %floor(imWidth*0.73);   %region total width(end val)
        iRg=34;      %floor(imHeight*0.64);  % Region i start
        jRg=25;      %floor(imWidth*0.27);   % Region j start
        Dist=0;
        cNum=0;
        avgDist=0;
        for i=iRg:hRg
            for j=jRg:wRg
                if(A(i,j)~=255)
                    fValue=getFuzzy1(i,j,imHeight,imWidth,8);
                    Dist=Dist+ sqrt((i-1)^2 +(j-1)^2)*fValue;
                    cNum=cNum+1;
                end
            end
        end
        if (Dist~=0)
            avgDist=Dist/cNum;
        end
        fprintf(fp,' %1f   ',avgDist);

        %9rd Region- Region I
        hRg=imHeight;        % region total width(end val)
        wRg=imWidth;         % region total width(end val)
        iRg=25;              %floor(imHeight*0.64);  % Region i start
        jRg=41;              %floor(imWidth*0.64);   % Region j start
        Dist=0;
        cNum=0;
        avgDist=0;
        for i=iRg:hRg
            for j=jRg:wRg
                if(A(i,j)~=255)
                    fValue=getFuzzy1(i,j,imHeight,imWidth,9);
                    Dist=Dist+ sqrt((i-1)^2 +(j-1)^2)*fValue;
                    cNum=cNum+1;
                end
            end
        end
        if (Dist~=0)
            avgDist=Dist/cNum;
        end
        fprintf(fp,'%1f   \n',avgDist);
    end
    fclose(fp);
end
```

APPENDIX – III

MATLAB Implementation of ID3 Decision Tree Induction Procedure and ANFIS.

```
% ID3 Decision tree induction priocedure for pre-processing the rule set obtained
% from the feature vectores. These reduced optimum rule set with maximum information
% about the class boundaries are given as input to the inference part of ANFIS
% Recursive function call
function t = idn3(mat,attribute_list)
[sr,sc] = size(mat);
num_attributes = sc - 2;
num_target_class = 44;
attribute_name=['atr1','atr2','atr3','atr4','atr5','atr6','atr7','atr8','atr9'];
fval = ['L','M','S'];
% if all members of examples are in the same class c
% number of entries for each target class
tar = zeros(1,num_target_class);
tarnum = str2num(mat(:,sc-1:sc));
for i = 1 : num_target_class
    tar(1,i) = length(find(tarnum == i));   % stores number of entries for each
target class
end
% tar(1:num_target_class) = 358;
sclasy = find(tar(1,:) == sr);
if length(sclasy)         % if all members are in the same class
    % t = tree('Label',sclasy);
    t = sclasy;
elseif  sc == 2       % if attribute list is empty
    maxvc = find(tar(1,:) == max(tar(1,:)));
    % t = tree('Label',maxvc(1));
    t = maxvc;
    clear maxvc;
else
    maxgain = -10;
    for i = 1 : num_attributes   % finding the maximum value of infromation gain
        gai = infogain(mat,i);
        if gai > maxgain maxgain = gai; index = i; end
    end
    clear maxgain;
    for i = 1 : 3  % for each possible value of attribute
        rex = find(mat(:,index) == fval(i));
        if (length(rex))
            %sc,length(rex),index
            nmat = zeros(length(rex),sc-1);
            nmat = mat(rex,1:index - 1);      % new matrix
            nmat = mat(rex,index + 1 : sc); % with out the selected attribute;
            [nsr,nsy] = size(nmat);
            attribute_list1 = setdiff(attribute_list,[attribute_list(index)]);
            if i == 1 t1 = idn3(nmat,attribute_list1);
            elseif i == 2 t2 = idn3(nmat,attribute_list1);
            elseif i == 3 t3 = idn3(nmat,attribute_list1); end
            clear nsr,
            clear nsy;
            clear nmat;
            clear attribute_list1;
        else
            maxvc = find(tar(1,:) == max(tar(1,:)));
            if i == 2 t2 = maxvc; end
            if i == 3 t3 = maxvc;end
            clear maxvc;
        end
        clear rex;
    end
    t = tree('Label,Lb,Mb,Sb',char(attribute_list(index)),t1,t2,t3);
    clear mat;
    clear num_attributes;
    clear num_target_class;
    clear tar;
    clear tarnum;
    clear sr;
```

```
        clear sc;
        clear sclasy;
    end
        % Function to compute the Infomation Gain for ID3 procedure
        function gain = infogain(mat,atrnum)
        [r,c] = size(mat);
        entr = 0;
        if r
            [rt1,ct1] = find(mat(:,atrnum) == 'L');
            [rt2,ct2] = find(mat(:,atrnum) == 'M');
            [rt3,ct3] = find(mat(:,atrnum) == 'S');

            % calculating the total entropy of the attribute
            v1 = 0;
            v2 = 0;
            v3 = 0;
            if length(rt1) v1 = -(length(rt1)/r) * log2(length(rt1)/r);end
            if length(rt2) v2 = -(length(rt2)/r) * log2(length(rt2)/r); end
            if length(rt3) v3 = -(length(rt3)/r) * log2(length(rt3)/r); end
            entr = v1 + v2 + v3;

            % entropy of each fuzzy state, for the attribute
            tarnum = str2num(mat(:,c-1:c));
            entr1 = 0;
            if length(rt1)
                for i = 1 : 44  % number of target class
                    [x,y] = intersect(find(mat(:,atrnum) == 'L'),find(tarnum == i));
                    if length(x)entr1 = entr1 + -(length(x)/length(rt1)) *
        log2(length(x)/length(rt1)); end
                    % insert command to drop x and y.
                    clear x; clear y;
                end
                entr1 = entr1*(length(rt1)/r);
            end

            entr2 = 0;
            if length(rt2)
                for i = 1 : 44  % number of target class
                    [x,y] = intersect(find(mat(:,atrnum) == 'S'),find(tarnum == i));
                    if length(x) entr2 = entr2 + -(length(x)/length(rt2)) *
        log2(length(x)/length(rt2));end
                    % insert command to drop x and y.
                end
                entr2 = entr2*(length(rt2)/r);
            end

            entr3 = 0;
            if length(rt3)
                for i = 1 : 44  % number of target class
                    [x,y] = intersect(find(mat(:,atrnum) == 'M'),find(tarnum == i));
                    if length(x) entr3 = entr3 + -(length(x)/length(rt3)) *
        log2(length(x)/length(rt3));end
                    % insert command to drop x and y.
                    clear x; clear y;
                end
                entr3 = entr3*(length(rt3)/r);
            end
            gain = entr - entr1 - entr2 - entr3
        else gain = -100;
        end


        % Programe for ID3 decision tree traversal
        % two attributes, root and travlist, indicating entries traversed so far
        function trav(root,trav_list,trav_fval)
        % initialisation of rule data base
        % rule_set = zeros(100,16);
        % index = 0;
        % trav_list = char([]);
        % trav_fval = [];
         global index;
         global rule_set;
        % terminating condietion
        [x,y] = size(trav_list);
        if istree(root) == 0
            index = index + 1;
            for i = 1 : x
                lnum = trav_list(i,4:5);
                num = str2num(lnum);
                rule_set(index,num) = trav_fval(i);
            end
```

```
        else
            % visit the node
            val =branch(root,'1');
            trav_list(x+1,:) = val;
            trav_fval1 = trav_fval;
            trav_fval2 = trav_fval;
            trav_fval3 = trav_fval;
            %trav_fval1 = trav_fval2 = trav_fval3 = trav_fval;
            trav_fval1(x+1) = 1;
            trav_fval2(x+1) = 2;
            trav_fval3(x+1) = 3;
            % further traversel;
            trav(branch(root,'2'),trav_list,trav_fval1);
            trav(branch(root,'3'),trav_list,trav_fval2);
            trav(branch(root,'4'),trav_list,trav_fval3);
        end

% ANFIS Implementation for Malayalam handwritten Character Recognition
% Feature vector Dimension imdim, number of fuzzy states numfstate=3
% (coresponding to the fuzzy sets SMALL, MEDIUM and LARGE)
% Input matrix is d, with the desired output in the last column.
clc;
error = 10;
count = 0;
% Initialisation
b = 0.375;
ipdim = 9;
numfstate = 3;
c = zeros(ipdim,numfstate);
si = zeros(ipdim,numfstate);
c(:,1) = 0.75,c(:,2) = 0.5, c(:,3) = 0.25;
si(:,1) = 0.75,si(:,2) = 0.5, si(:,3) = 0.25;
cons_param = ones(index,ipdim+1);
dat = zeros(1,10);
while count <= 20000
    error = 0.01;
    count = count + 1;
    for ind1 = 1 : 44
        for ind2 = 1 : 15
            clear dat;
            dat = zeros(1,10);

            % Layer 1 :
            % input : input-output vector pair - matrix d
            % output: 27(9*3)dimensionel vector - fuzzy membership values

            dat(1,1:9) = d((ind1-1)* 358 + ind2,1:9);  % input data
            dat(1,10) = 1;
            z = d((ind1-1)*358 + ind2,10); % desired output
            layer1_output = zeros(1,27);

            for i = 1 : ipdim
                for j = 1 : numfstate
                    layer1_output(1,(i-1)*3 + j) = exp(-0.5*(((dat(1,i) -
                                  c(i,j))/si(i,j))^2));
                end
            end

            % layer 2
            % input : rule_set and index - the number of rules is assumed to
            %         be available in the work space as an output of ID3
            %         procedure and the outputs from layer 1.
            %         rule_set values 1 2 3 indicating fuzzy states
            %         SMALL, MEDIUM and LARGE) for each rule)
            % output: Firing strength of each rule - output vector  size is
            %         equal to the number of ID3 derived rules

            fire_strength = zeros(1,index);
            for i = 1 : index
                val = 1;
                for j = 1 : ipdim
                    if rule_set(i,j)
                        in = (j-1)*3 + rule_set(i,j);
                        val = val * layer1_output(in);
                    end
                end
                fire_strength(1,i) = val;
            end
```

```
            % Layer3 : For computing Normalised Firing Strengths of each
            % rule

            layer3_output = zeros(1,index);
            sum = 0;
            for i = 1 : index
                sum = sum + fire_strength(1,i);
            end

            layer3_output(1,:) = fire_strength(1,:)/sum;
            % Layer4 :
            % Consequent parameters cons_param of size(index,10) is
            % initialised
            % dat(1,10)=1;

            layer4_output = zeros(1,index);
            layer4_output = cons_param * dat';
            fi = layer4_output;
            for i = 1 : index
                layer4_output(i,1) = layer4_output(i,1) * layer3_output(1,i);
            end

            % Layer5
            % output : Overall outputs, sum of all incoming values  =
            %          sum(layer4_output(:,1));
            sum = 0;
            for i = 1 : index
                sum = sum + layer4_output(i,1);
            end

            output = sum;
            error = error + abs(z-output);

            % Parameters Updation
            % Update parameters participating in each rule
            % index i for rule number, j for participating attribute

            fiwi = 0;
            for i = 1 : index
                fiwi = fiwi + fire_strength(1,i) * fi(i,1);
            end

            % Updating c
            for i = 1 : index
                for j = 1 : ipdim
                    if rule_set(i,j)
                        dif1 = z - output;
                        dif2 = fi(i,1)-fiwi;
                        dif3 = dat(1,j)-c(1,rule_set(i,j));
                        dif4 = dif3/(si(1,rule_set(i,j))^2);
                        c(j,rule_set(i,j)) = c(j,rule_set(i,j)) - (2 * dif1 *
                                        layer3_output(1,i) * dif2 * dif4);
                    end
                end
            end

            %  Updating sigma
            for i = 1 : index
                for j = 1 : ipdim
                    if rule_set(i,j)
                        dif1 = z - output;
                        dif2 = fi(i,1)-fiwi;
                        dif3 = dat(1,j)-c(1,rule_set(i,j));
                        dif4 = (dif3^2)/(si(1,rule_set(i,j))^3);
                        si(j,rule_set(i,j)) = si(j,rule_set(i,j)) - (2 * dif1*
                                        layer3_output(1,i) * dif2 * dif4);
                    end
                end
            end
            % Updating q
            for i = 1 : index
                for j = 1 : ipdim
                    cons_param(i,j) = cons_param(i,j) + (z - output) *
                                   layer3_output(1,i) * dat(1,j);
                end
            end
        end
    end
end
```

REFERENCES

[1] Abdulla.W.H, Saleh.A.O.M and Morad.A.H, "A pre-processing algorithm for handwritten character recognition", *Pattern Recognition Letters*, vol.7, no.1, pp.13-18, 1988.

[2] Abuhaiba.S.I and Ahmed.P, "A fuzzy graph theoretic approach to recognize the totally unconstrained handwritten numerals", *Pattern Recognition*, vol. 26, no.9, pp.1335-1350, 1993.

[3] Abutaleb.A.S, "Automatic thresholding of gray-level pictures using two dimensional entropy", *Computer Vision, Graphics, and Image Processing*, vol.47, pp.22-32, 1989.

[4] Ahmed.M. and Ward.R, "A rotation invariant rule-based thinning algorithm for character recognition", *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*, vol.24, no.12, pp.1672-1678, 2002.

[5] Aksela.M and Laaksonen.J, "Adaptive combination of adaptive classifiers for handwritten character recognition", *Pattern Recognition Letters*, vol. 28, pp. 136-143, 2007.

[6] Alcorn.T.M and Hoggar.C.W, "Pre-processing of data for character recognition", *Marconi Rev.*, vol.32, pp. 61-81, 1969.

[7] Ali.F and Pavlidis.T, "Syntactic recognition of handwritten numerals", *IEEE Trans. Systems Man Cybernetics*, vol.7, pp.537-541, 1977.

[8] Amano.A and Arisuka.T, "On the use of neural networks and fuzzy logic in speech recognition", *Proc. Int. Joint Conf. on Neural Networks,* pp.301-305, 1989.

[9] Amin.A and Al-Sadoun.H.B, "Hand printed Arabic character recognition system", *Proc. of the 12th ICPR*, pp. 536-539, 1994.

[10] Amin.A, "Off-line Arabic character recognition: The state of the art", *Pattern Recognition*, vol.31, no.5, pp.517-530, 1998.

[11] Anand.R, Mehrotra K, Mohan.C.K and Renka.S, "Efficient classification for multi-class problems using modular neural networks", *IEEE Trans. on Neural Networks,* vol. 6, no.1, pp. 117-124, 1995.

[12] Anasuya Devi.H.K, "Fuzzy neural network for Brahmi character recognition in Epigraphy", *Proc. 2nd National Conf. on Document Analysis and Recognition, Mandya, India*, pp.172-180, 2003.

[13] Andrews.H.C, "Multidimensional rotations in feature selection", *IEEE Trans. Computers*, vol.20, pp.1045-1051, 1971.

[14] Arcelli.C and Baja.G.S.D, "A width independent fast thinning algorithm", *IEEE Trans. PAMI.*, vol. PAMI-7, no.4, pp. 463-474, 1985.

[15] Arcelli.C, Cordella.L.P and Levialdi.S, "From local maximum to connected skeletons", *IEEE Trans. PAMI*, vol.3, no.2, pp.134-143, 1981.

[16] Arcelli.C, "A condition for digital point removal", *Signal Processing*, vol.1, no. 4, pp.283-285, 1979.

[17] Arcelli.C, "Pattern thinning by contour tracing," *Computer Graphics and Image Processing*, vol.17, pp.130-144, 1981.

[18] Arora.S, Malik.L, Bhattacharjee.D and Nasipuri.M, "A novel approach for handwritten Devanagari character recognition", *Proc. IEEE 1st Int. Conf. on Signal and Image Processing, Hubli, India*, vol.1, pp.164-167, 2006.

[19] Ayat.N.E, Cheriet.M and Suen.C.Y "Optimization of the SVM kernels using an empirical error minimization scheme.", *Proc. Int. Workshop on Pattern Recognition with Support Vector Machine, Niagara Falls-Canada*, pp.354-369, 2002.

[20] Baker.G.L and Gollub.J, "Chaotic dynamics - An introduction", *Cambridge University Press*, 1996.

[21] Balm.G.J, "An introduction to optical character reader considerations", *Pattern Recognition*, vol.2, no.3, pp.151-166, 1970.

[22] Bansal.V and Sinha.R.M.K, "Integrating knowledge sources in Devanagari text recognition system", *IEEE Trans. Systems, Man and Cybernetics*, vol.30, no.4, pp.500-505, 2000.

[23] Battiato.S, Gallo.G and Stanco.F, "A new edge adaptive algorithm for zooming of digital images", *Proc. of IASTED Signal processing and Communications SPC 2000, Marbella, Spain*, pp.144-149, 2000.

[24] Belkasim.S.O, Shridhar.M and Ahmadi.A, "Pattern recognition with moment invariants : A comparative study and new results", *Pattern Recognition*, vol.24, pp.1117-1138, 1991

[25] Beun.M, "A flexible method for automatic reading of handwritten numerals", *Philips Tech. Rev.*, vol.33, no.5, pp. 89-101, 1973.

[26] Bhattacharya.U, Das.T.K, Datta.A, Parui.S.K and Chaudhuri.B.B, "A hybrid scheme for handprinted numeral recognition based on a self-organizing network and MLP classifiers", *Int. Journal of Pattern Recognition and Artificial Intelligence, World Scientific*, vol.16, no.7, pp.845-864, 2002.

[27] Bhowmik.T.K, Bhattacharya.U and Parui.S.K, "Recognition of Bangla handwritten characters using an MLP classifier based on stroke features", *Proc. Int. Conf. Neural Information Processing, Kolkata, India*, pp. 814-819, 2004.

[28] Bimbo.A.D, Santini.S and Sanz.J, "OCR from poor quality images by deformation of elastic templates", *Proc. 12th IAPR Int. Conf. Pattern Recognition, Jesusalem, Israel,* vol.2, pp.433-435, 1994.

[29] Bishop.C.M, "Neural networks for pattern recognition.", *Oxford Univ. Press, Oxford – U.K*, 1995

[30] Bokser.M, "Omnidocument technologies", *Proc. IEEE*, vol.80, pp. 1066-1078, 1992.

[31] Borgefors.G, "Distance transformations in arbitrary dimensions", *Computer Vision, Graphics, and Image Processing*, vol.27, pp.321-345, 1984.

[32] Bortolozzi.F, Britto Jr.A de S, Oliveira.L.S and Morita.M, "Recent advances in handwriting recognition", *Proc. of the IWDA, Kolakta, Editors Pal.U, Parui.S.K, Chaudhuri.B.B*, pp.1-30, 2005.

[33] Brink.A.D and Pendock.N.E, "Minimum cross entropy threshold selection", *Pattern Recognition*, vol.29, no.1, pp.179-188, 1996.

[34] Britto.S.Jr., Sabourin.R., Bortolozzi.F and Suen.C.Y, "A two-stage HMM based system for recognizing handwritten numeral strings", *Proc. 6th Int. Conf. on Document Analysis and Recognition, Seattle-USA*, pp.396-400, 2001.

[35] Britto.S.Jr., Sabourin.R., Bortolozzi.F and Suen.C.Y, "Foreground and background information in an HMM-based method for recognition of isolated characters and numeral strings", *Proc. 9^th^ Int. Workshop on Frontiers in Handwriting Recognition, Tokyo, Japan*, pp.371-376, 2004.

[36] Broomhead.D.S and King.G, "Extracting qualitative dynamics from experimental data", *Physica D20*, 1986.

[37] Cai.J and Liu.Z.Q, "Integration of structural and statistical information for unconstrained handwritten numeral recognition", *IEEE Tran. on PAMI*, vol.21, no.3, pp. 263-270, 1999.

[38] Cao.J, Ahmadi.M and Shridhar.M, "Handwritten numeral recognition with multiple features and multistage classifiers", *Proc. IEEE Int. Symposium on Circuits and Systems, London*, vol.6, pp.323-326, 1994.

[39] Chandrasekharan.M, Chandrasekharan.R and Siromoney.G, "Context depended recognition of hand printed Tamil characters", *Proc. Int. Conf. System Man Cybernetics, India*, vol.2, pp. 786-790, 1984.

[40] Chaudhuri.B.B, Pal.U and Mitra.M, "Automatic Recognition of printed Oriya Script", *Sadhana*, vol. 27, part 1, pp. 23-24, 2002.

[41] Chaudhuri.B.B, Kumar.O.A and Ramana.K.V, "Automatic generation and recognition of Telungu script characters", *Journal of IETE*, vol. 37, pp. 499- 511, 1991.

[42] Chaudhuri.B.B, Pal.U, "A complete printed Bangala OCR system", *Pattern Recognition*, vol.31, no.5, pp.531-549, 1998.

[43] Chen.M.H, Lee.D and Pavlidis.T, "Residual analysis for feature detection", *IEEE Trans. on PAMI*, vol.13, pp.30-40, 1991.

[44] Chen.M.-Y and Linkens.D.A, "A systematic neuro-fuzzy modeling frame work with application to material property prediction, *IEEE Trans. Systems, Man and Cybernetics*, vol.31, no.5, pp.781-790, 2001.

[45] Chen.Y.-S and Hsu.W.-H, "A modified fast parallel algorithm for thinning digital patterns", *Pattern Recognition Letters*, vol.7, no.2, pp.99-106, 1988.

[46] Chin.R.T, Wan.H.-K, Stover.D.L and Iverson.R.D, "A one pass thinning algorithm and its parallel implementation", *Computer vision, Graphics, and Image Processing*, vol.40, pp.30-40, 1987.

[47] Chinnuswami.P and Krishnamoorthi.S.G, "Recognition of hand printed Tamil Characters", *Pattern Recognition*, vol.12, pp.141-152, 1980.

[48] Cho.S.B, "Neural-network classifiers for recognizing totally unconstrained handwriting numerals", *IEEE Trans. Neural Networks*, vol.8, no.1, pp.43-53, 1997.

[49] Chow.C.K, "An optimum character recognition system using decision functions", *IRE Transactions on Electronic Computers*, vol.6, pp.247-254, 1957.

[50] Chu.Y.K and Suen.C.Y, "An alternative smoothing and stripping algorithm for thinning digital binary patterns", *Signal Processing*, vol.11, no.3, pp.207-222, 1986.

[51] Danielsson.P.E, "Euclidean distance mapping", *Computer Graphics and Image Processing*, vol.14, pp.227-248, 1980.

[52] Davies.E.R and Plummer.A.P.N, "Thinning algorithms: A critique and a new methodology", *Pattern Recognition*, vol.14, nos.1-6, pp.53-63, 1981.

[53] Deutsch.E.S, "Preprocessing for character recognition", *Proc. IEEE NPL Conf. Pattern Recognition, Teddington*, pp.179-190, 1968.

[54] Devaraj.D and Yagnanarayana.B, "An adaptive fuzzy system model for power system security assessment", *Proc. Int. Conf. Communications, Control and Signal Processing, Bangalore, India*, pp.284-288, 2000.

[55] Dimauro.G, Impedovo.S, Pirlo.G and Salzo.A, "Automatic bank check processing: A new engineered system", *Int. Journal of Pattern Recognition and Artificial Intelligence, World Scientific, Editors-Impedovo.S et al.*, pp.467-503, 1997.

[56] Dinnen.G.P, "Programming pattern recognition", *Proc. West Joint Computer Conf., New York*, pp.94-100, 1955.

[57] Dong.J.X, Krzyzak.A and Suen.C.Y, " Local learning framework for recognition of lowercase handwritten characters", *Proc. Int. Workshop on Machine learning and Data Mining in Pattern Recognition, Leipizing*, Germany, PP.226-238, 2001.

[58] Duda.R.O and Hart.P.E, "Pattern classification and scene analysis", *Wiley Interscience, New York*, 1973.

[59] Duda.R.O, Hart.P.E and Stork.D.G, "Pattern classification", *John Wiley & Sons, Second Edition*, 2001.

[60] Dutta.A.K and Chaudhuri.S, "Bengali alpha-numeric character recognition using curvature features", *Pattern Recognition* vol.26, pp.1757-1770, 1993.

[61] Engin.M, "ECG beat classification using neuro-fuzzy network", *Pattern Recognition Letters*, vol.25, pp.1715-1722, 2004.

[62] ERA, "An electronic reading automaton", *Electronic Engineering*, vol.29, no.4, pp.189-190, 1957.

[63] Friedman.M. and Kandel.A., "Introduction to pattern recognition-statistical, structural, neural and fuzzy logic approach", *World Scientific*, 1999.

[64] Gader.P.D, Forester.B and Ganzberger.M, "Recognition of handwritten digits using template and model matching", *Pattern Recognition*, vol.5, no.24, pp.421-431, 1991.

[65] Gader.P.D, Keller.J.M and Cai.J, "A fuzzy logic system for detection and recognition of street number fields on handwritten postal addresses", *IEEE Trans. on Fuzzy Systems*, vol.3, no.1, pp.83-95, 1995

[66] Garris.M.D and Wilkinson.R.A, "Handwritten segmented characters database: NIST special database3, technical report", *NIST technical report and CDROM*, 1992.

[67] Glauberman.M.H., "Character recognition for business machine", *Electronics*, vol.29, no.2, pp.132-136, 1956.

[68] Gonzalez.R.C and Woods.R.E, "Digital image processing", *Pearson Education, Singapore*, 2002.

[69] Govindan.V.K and Shivaprasad.A.P, "A pattern adaptive thinning algorithm", *Pattern Recognition*, vo.20, no.6, pp.623-637, 1987.

[70] Govindaraju.V, Shekhawat.A and Srihari.S.N, "Interpretation of handwritten addresses in US mail stream", *Proc. 3rd Int. Workshop on Frontiers in Handwriting Recognition, Buffalo, New York*, pp.197-206, 1994.

[71] Guillevic. D and Suen.C.Y, "Cursive script recognition applied to the processing of bank cheques", *Proc. 3rd Int. Conf. on Document Analysis and Recognition, Montreal-Canada*, pp.11-14, 1995.

[72] Guo.Z and Hall.R.W, "Parallel thinning with two-sub iteration algorithms", *Comm. ACM*, vol.32, no.3, pp.359-373, 1989.

[73] Guyon.I, Haralick.R.M, Hull.J.J and Phillips.I.T, "Data sets for OCR and document image understanding research", *Hand Book of Character Recognition and Document Image Analysis, Editors-Bunke.H and Wang.P.S.P, World Scientific*, pp.779-799, 1997.

[74] Ha.T.M and Bunke.H, "Image processing methods for document image analysis", *Hand book of character recognition and document image analysis, Editors-Bunke.H and Wang.P.S.P, World Scientific*, pp.1-47, 1997.

[75] Hanmandlu.M, Mohan.K.R.M, Chakraborty, Goyal.S and Choudhury.R, "Unconstrained handwritten character recognition based on fuzzy logic", *Pattern Recognition*, vol.36, no.3, pp. 603-623, 2003.

[76] Hanmandlu.M, Ramana Murthy.O.V and Agarwal.P, "Fuzzy model based recognition of handwritten Hindi characters", *Proc. IEEE 1st Int. Conf. on Signal and Image Processing, Hubli, India*, vol.1, pp.184-189, 2006.

[77] Haralick.R.M, Watson.L.T and Laffey.T.J, "The topographic primal sketch", *Int. J. Robotics*, vol. 2, pp.50-72, 1983.

[78] Haykin.S, "Neural networks: A comprehensive foundation", *Prentice-Hall of India Pvt.Ltd*, 2004.

[79] Hebb.D.O, "The organization of behavior", *A Neuropsychological Theory, Wiley-New York*, 1949.

[80] Heutte.L, Paquet.T, Moreau.J.V, Lecourtier.Y and Oliver.C, "A structural/statistical feature based vector for handwritten character recognition", *Pattern Recognition*, vo.19, pp.629-641, 1998.

[81] Holbaek-Hanssen.E, Brathen.K and Taxt.T, "A general software system for supervised statistical classification of symbols", *Proc. of the 8Th Int. Conf. on Pattern Recognition, Paris, France,* pp.144-149, 1986.

[82] Holt.C, Stewart.A, Clint.M and Perrott.R.H, "An improved parallel thinning algorithm", *Comm. ACM*, vol.30, no.2, pp.297-306, 1987.

[83] Hu.M.-K., "Visual pattern recognition by moment invariants", *IRE Trans. Information Theory*, vol. IT-8, pp. 179-187, 1962.

[84] Huang L.K and Wang.M.J.J, "Image thresholding by minimizing the measures of fuzziness", *Pattern Recognition*, vol.28, no.1, pp.41-51, 1995.

[85] Hull.J.J, Commike.A and Ho.T.K., "Multiple algorithm for handwritten character recognition", *Proc. First int. workshop on Frontiers in Handwriting Recognition, Montreal*, pp.117-124, 1990.

[86] Hull.J.J., "A database for handwritten text recognition research", *IEEE Trans. PAMI*, vo.16, no.5, pp.550-554, 1994.

[87] Hunt.E.B, Marin.J and Stone.P, "Experiments in induction", *Academic Press, New York,* 1966.

[88] Iyatomi.H and Hagiwara.M, "Adaptive fuzzy inference neural network", *Pattern Recognition*, vol.37, pp. 2049-2057, 2004.

[89] Iyatomi.H and Hagiwara.M, "Scenery image recognition and interpretation using fuzzy inference neural networks", *Pattern Recognition*, vol.35, no.8, pp.1793-1806, 2002.

[90] Jain.A.K and Chandrasekaran.B, "Dimensionality and sample size considerations in pattern recognition practice", *Classification, Pattern Recognition and Reduction of Dimensionality, Editors-Krishnaiah.P.R and Kanal.L.N, Handbook of Statistics, Amsterdam: North-Hollandb,* vol.2, pp. 835-855, 1982.

[91] Jain.A.K, Duin.R.P.W and Mao.J, "Statistical pattern recognition: A review", *IEEE Trans. on PAMI*, vol.22, no.1, pp. 4-37, 2000.

[92] Jamadagni.S and Ghadialy.Z, "Adaptive neuro-fuzzy control of strip temperature in a hot strip mill run out table", *Proc. Int. Conf. Communications, Control and Signal Processing, Bangalore, India*, pp.187190, 2000.

[93] Jang.J.-S.R., "ANFIS: Adaptive-network–based fuzzy inference system", *IEEE Trans. Systems, Man, and Cybernetics*, vol.23, no.3, pp. 665-685, 1993.

[94] Jang.J.-S.R., Sun.C.T and Mizutani.E, "Neuro-Fuzzy and soft computing- A computational approach to learning and machine intelligence", *Prentice-Hall of India Pvt.Ltd*, 2002.

[95] Joshi.A, Ramakrishnan.N, Houstis.E.N and Rice.J.R, "On neurobiological, neuro-fuzzy, machine learning, and statistical pattern recognition techniques", *IEEE Trans. Neural networks*, vol.8, no.1, 1997.

[96] Kahan.S, Pavlidis.T and Baird.H.S, "Building a font and size independent character recognition system", *IEEE Trans. on PAMI*, vol.9, pp.274-288, 1987.

[97] Kahan.S, Pavlidis.T and Baird.H.S, "On the recognition of printed characters of any font and size", *IEEE Trans. on PAMI,* vol.9, pp.274-288, 1987.

[98] Kantz.H and Schreiber.T, "Nonlinear time series analysis", *Cambridge University Press,* 1998.

[99] Kapp.M.N, Freitas.C.O.DeA., Nievol.J.C and Sabourin.R, "Evaluating the conventional and class-modular architectures feedforward neural network for handwritten word recognition", *Proc. XVI Brazilian Symposium on Computer Graphics and Image Processing, (SIBGRAPI)*, 2003.

[100] Kapur.J.N, Sahoo.P.K and Wong.A.K.C, "A new method for gray-level picture thresholding using the entropy of the histogram", *Computer vision, Graphics and Image Processing*, vol.29, pp.273-285, 1985.

[101] [1]Khotanzad.A and Hong.Y.H, "Invariant image recognition by Zernike moments", *IEEE Trans. on PAMI,* vol.12, no.5, pp.489-497, 1990.

[102] [2]Khotanzad.A and Hong.Y.H, "Rotation invariant image recognition using features selected via systematic method", *Pattern Recognition*, vol.23, no.10, pp. 1089-1101, 1990.

[103] Kim.C-H and Lee J-J, "Adaptive network-based fuzzy inference system with pruning", *Proc. SICE Annual Conference, Fukui, Japan*, pp.140-143, 2003.

[104] Kim.H.Y and Kim.J.H., "Handwritten Korean character recognition based on hierarchical random graph modeling", *Proc. 6th International Workshop on Froniers of Handwritten Recognition, Taegon-Korea*, pp.557-586, 1998.

[105] Kimura.F and Shridhar.M, "Handwritten numeral recognition based on multiple algorithms", *Pattern Recognition,* vol.24, no.10, pp.969-983, 1991.

[106] Kirsch.R.A, Cahn.L, Ray.C and Urban.G.J, "Experiments in processing pictorial information with a digital computer", *Proc. East Joint Computer Conf., New York,* pp.221-229, 1957.

[107] Kittler.J and Illingworth.J, "Minimum error thresholding", *Pattern Recognition,* vol.19, no.1, pp.41-47, 1986.

[108] Kittler.J, Hatef.M, Fuin.R, Matas.J, "On combining classifiers", IEEE *Trans. on PAMI*', vol.20, no.3, pp.226-239, 1998.

[109] Kohonen.T, "The self-organizing map", *Proc. IEEE,* vol.778, no.9, pp.1464- 1480, 1990.

[110] Kundu.A, He.Y, Chen.M, "An off-line cursive handwriting recognition system", *IEEE Trans. on PAMI,* vol.20, no.11, pp.1275-1280, 2002.

[111] Kurita.T, Otsu.N and Abdelmalek.N, "Maximum likelihood thresholding based on population mixture models", *Pattern Recognition,* vol.25, no.10, pp.1231-1240, 1992.

[112] Kwan.H.K and Cai.Y, "A fuzzy neural network and its application for pattern recognition", *IEEE Trans. on Fuzzy Systems,* vol.2, no.3, pp.185-192, 1994.

[113] Lam.L and Suen.C.Y, "Structural classification and relaxation matching of totally unconstrained handwritten zip-code numbers", *Pattern Recognition,* vol.21, no.1, pp.19-31, 1988.

[114] Lam.L., Lee.S-W and Suen.C.Y, "Thinning methodologies– A comprehensive survey", *IEEE Trans. PAMI,* vol.14, no.9, 1992.

[115] Lazzerini and Marcelloni.F, "A linguistic fuzzy recognizer of off-line handwritten characters", *Pattern Recognition Letters,* vol.21, no.4, pp.319-327, 2000.

[116] LeCun.Y, Bottou.L, Bengio.Y and Haffner.P, "Gradient-based learning applied to document recognition", *Proc. IEEE,* vol.86, no.11, pp.2278-2324, 1998.

[117] Lee.S.W and Park.J.S, "Non linear shape normalization methods for the recognition of large-set handwritten characters", *Pattern Recognition,* vol.27, pp.895-902, 1994.

[118] Lethelier, Leroux.M., Gilloux.M., "An automatic reading system for handwritten numeral amounts on French checks", *Proc. 3rd Int. Conf. on Document Analysis and Recognition, Montreal-Canada,* , pp.92-97, 1995.

[119] Lin.C.-T and Lee.C, "Neural-networks-based fuzzy logic control and decisions systems", *IEEE Trans. Computers (Special Issue on Artificial Neural Networks)*, vol. 40, pp.1320- 1336, 1991.

[120] Lin.Y and Cunnigham.G.A, "A new approach to fuzzy–neural system modeling", *IEEE Trans. Fuzzy Systems*, vol. 3, pp. 190-197, 1995.

[121] Ling, Lizaraga.M, Gomes.N and Koerich.A. "A prototype for Brazilian bank check recognition", *Int. journal of Pattern Recognition and Artificial Intelligence, Editors- Impedovo.S et al., World Scientific*, pp. 549-569, 1997.

[122] Looney.C.G, "Pattern recognition using neural networks", *Oxford University Press, New York*, 1997.

[123] Mamdani.E.H and Assilian.S, "An experiment in linguistic synthesis with a fuzzy logic controller", *Int. Journal of Man-Machine Studies,* vol.7, no.1, pp. 1-13, 1975.

[124] Mardia.K.V and Hainsworth.T.J, "A spatial thresholding method for image segmentation", *IEEE Trans. on PAMI*, vol.10, no.6, pp.919-927, 1988.

[125] Mavdia.K.V and Hainsworth.T.J, "A spatial thresholding method for image segmentation", *IEEE Trans. PAMI*, vol.10, no.6, pp.919-927, 1988.

[126] McCulloch.W.S and Pitts.W, "A logical calculus of the ideas immanent in nervous activity", *Bulletin of Mathematical Biophysics*, vol.5, pp.115-133, 1943.

[127] Mico.L, Oncina.J, "Comparison of fast nearest neighbour classifier for handwritten character recognition" *Pattern Recognition Letters*, vol.19, no.3-4, pp.351-356, 1999.

[128] Minsky.M.L and Papert.S.A, "Perceptrons: An introduction to computational geometry", *MIT Press, Cambridge-M.A.*, 1988.

[129] Mui.L, Agarwal.A, Gupta.A and Wang.P.S.P, "An adaptive modular neural network with application to unconstrained character recognition", *Document Image Analysis, Editors-Bunke.H, Wang.P.S.P and Baird.H.S, World Scientific, Singapore*, pp. 1189-1203, 1994.

[130] Nackman.L.R, "Two–dimension critical point configuration graphs", *IEEE Trans. on PAMI*, vol. 6, pp.442-450, 1984.

[131] Nadal.C, Legault.R, and Suen.C.Y, "Complementary Algorithms for the Recognition of totally Unconstrained Handwritten Numerals", *Proc. 10th Int. Conf. Pattern Recognition, Athlandic City, NJ, pp.443-*449, 1990.

[132] Nagabhushan.P & Radhika.M.Pai, "Modified region decomposition method and optimal depth decision tree in recognition of non-uniform sized characters- An experimentation with Kannada characters, *Pattern Recognition Letters,* vol. 20, pp.1467-1475, 1999.

[133] Nagabhushan.P, Angadi.S.A and Anami.B.S, "A fuzzy statistical approach to Kannada vowel recognition based on invariant moment", *Proc. 2nd National Conf. on Document Analysis and Recognition, NCDAR-2003, India,* pp.275-285, 2003.

[134] Nagy.G and Tuong.N "Normalization techniques for hand printed numerals", *Scientific Applications, Communications of ACM, Editor-Lawson.C.L*, vol.13, no.8, pp.475-481, 1970

[135] Nakagawa.Y and Rosenfeld.A, "Some experiments on variable thresholding" *Pattern Recognition*, vol.11, no. 3, pp. 191-204, 1979.

[136] Nishina.T and Hagiwara.M, "Fuzzy inference neural network", *Neuro Computing*, vol.14, pp.223 -239, 1997.

[137] Ogawa.H and Taniguchi.K, "Thinning and stroke segmentation for handwritten Chinese character recognition", *Pattern Recognition,* vol.15, no.4, pp.299-308, 1982.

[138] Oh.I.-S and Suen.C.Y, "Distance features for neural networks based of handwritten characters", *Int. Journal Document Anal Recognition*, vol.1, no.2, pp.73-88, 1998.

[139] Oh.I-S and Suen.C.Y, "A Class–modular feed forward neural networks for Handwriting Recognition", *Pattern Recognition*, vol.35, pp.229-244, 2002.

[140] Oliveira de J.J.Jr. and Veloso.L.R.Jr., "Interpolation/extrapolation scheme applied to size normalization of character images", *Proc. 15th Int. Conf. of Pattern Recognition, Barccelona, Spain*, vol.2, pp.577-580, 2000.

[141] Oliveira.L.S, Sabourin.R, "Support vector machines for handwritten numerical string recognition", *Proc. 9th International Workshop on Frontiers in Handwriting Recognition, Kokumbunji, Tokyo, Japan*, pp.39-44, 2004.

[142] Otsu.N, "A threshold selection method from gray-level histograms", *IEEE Trans. Systems, Man and Cybernetics*, vol.9, pp.62-66, 1979.

[143] Ott.E, "Chaos in dynamical systems", *Cambridge University Press*, 1993.

[144] Pal.N.R and Pal.S.K, "Entropic thresholding", *Signal Processing*, vol.16, pp.97-108, 1989.

[145] Pal.S.K and Rosenfeld.A, "Image enhancement and thresholding by optimization of fuzzy compactness", *Pattern Recognition Letters*, vol.7, pp.77-86, 1988.

[146] Pal.U and Chaudhuri.B.B, "Indian script character recognition-A survey", *Pattern Recognition*, vol.37, pp.1887-1899, 2004.

[147] Pal.U and Sarkar.A, "Recognition of printed Urdu script", *Proc. ICDAR, Edinburgh*, pp.598-602, 2003.

[148] Pal.U, Sharma.N, Kimura.F and Pal.S, "Off-line handwritten Kannada character recognition", *Proc. IEEE 1st Int. Conf. on Signal and Image Processing, Hubli, India*, vol.1, pp. 174-177, 2006.

[149] Pal.U. and Chaudhuri B.B, "Automatic recognition of unconstrained off-line Bangla hand-written numerals", *Advances in Multi model Interfaces, Editors-Tan.T, Shi.Y and Gao.W, Springer Verlag Lecture notes on Computer Science*, pp.371-378, 2000.

[150] Parker.J.R, "Gray level thresholding in badly illuminated images", *IEEE Trans. on PAMI*, vol.13, no. 8, pp. 813-819, 1991.

[151] Pavlidis.T, "A thinning algorithm for discrete binary images", *Computer Graphics and Image Processing*, vol.13, pp.142-157, 1980.

[152] Pernkopf.F, "Bayesian network classifiers versus selective k-NN classifier", *Pattern Recognition*, vol.38, pp.1-10, 2005.

[153] Peuker.T.K and Donglas.D.H, "Detection of surface-specific points by local parallel processing of discrete terrain elevation data", *Journal of Computer Graphics and Image Processing*, vol.4, no.4, pp-375-387, 1975.

[154] Pujari.A.K., Naidu.C.D, Rao.M.S and Jinaga.B.C, "An intelligent character recognizer for Telungu scripts using multiresolution analysis and associative memory", *Image and Vision Computing,* vol.22, pp.1221-1227, 2004.

[155] Quinlan.J.R, "Induction of decision trees", Machine Learning, 1(1), pp.81-106, 1986

[156] Rabiner.L.R, "A tutorial on Hidden Markov Models and selected applications in speech recognition", *Proc. of the IEEE*, vol.77, no.2, pp.257-286, 1989.

[157] Rahier M.C and Jespers P.G.A, "Dedicated LSI for micro processor controlled hand carried OCR system", *IEEE J. Solid-State Circuits*, vol.SC-15, no.1, pp.14-24, 1980

[158] Rahman.A.F.R., Rehman.R and Fairhurst.M.C, "Recognition of handwritten Bengali character : A novel multistage approach", *Pattern Recognition*, vol.35, pp.997-1006, 2002.

[159] Rajasekharan.S.N.S and Deekshatulu.B.L, "Recognition of printed Telungu Characters", *Computer Graphics and Image Processing*, vol.6, pp.335-360, 1977.

[160] Rangarajan.K, Shah.M and Brackle.V.D, "Optimal corner detector", *Proc. First Int. Conf. Computer Vision*, pp.90-94, 1988.

[161] Ray.A.K. and Chatterjee.B, "Design of a nearest neighbour classifier system for Bengali character recognition", *Journal of Inst. Elec. Telecom. Eng.*, vol.30, pp.226-229, 1984.

[162] Reiss.T.H, "Recognizing planar objects using invariant image features", *Lecture notes in Computer Science, Springer-Verlag*, vol.676, 1993.

[163] Reiss.T.H, "The revised fundamental theorem of moment invariants", *IEEE Trans. PAMI*, vol.13, pp. 830-834, 1991.

[164] Ridler.T and Calvard.S, "Picture thresholding using an iterative selection method", *IEEE Trans. Systems, Man and Cybernetics,* vol.8, pp.630-632, 1978.

[165] Ripley.B.D, "Pattern recognition and neural networks", *Cambridge University Press*, 1996.

[166] Rocha.J and Pavlidis.T, "Shape analysis model with applications to a character recognition system", *Proc. IEEE Workshop on Applications of Computer Vision, Palm Springs*, 1992.

[167] Rockett.P.I, “An improved rotation-invariant thinning algorithm”, *IEEE Trans. on PAMI*, vol. 27, no.10, 2005.

[168] Rosenblatt.F. “Two theorems of statical separability in the perception”. Symposium on the Mechanization of Thought Process, pp. 421-456, 1959.

[169] Rosenfeld.A and Pfaltz.J, “Distance functions on digital pictures”, *Pattern Recognition*, vol.1, pp.33-61, 1968.

[170] Roy.K *et al.*, “A system for Indian postal automation”, *Proc. of the IWDA 05, Editors Pal.U, Parui.S.K, Chaudhuri.B.B, Kolakta*, pp.249-274, 2005.

[171] Rumelhart.D.E, Hinton.G.E and Williams.R.J, “Learning internal representations by error propagation”, *Parallel and distributed processing: explorations in the Microstructure of Cognition, Eds. Rumelhart.D.E and McClelland.J.L, MIT Press, Cambridge, MA*, 1986.

[172] Sahoo.P.K, Soltani.S and Wong.A.K.C, “A survey of thresholding techniques”, *Computer Vision, Graphics and Image Processing,* vol.41, pp. 233-260, 1988.

[173] Sahoo.P.K, Soltani.S and Wong.A.K.C, “A survey of thresholding techniques”, *Computer Vision, Graphics, and Image Processing,* vol.41, pp.233-260, 1988.

[174] Sasi.S and Bedi.J.S, “Character recognition using fuzzy logic", *Proc. 37th MWSCS-94, Lafayette, Lousiana, USA*, pp. 1399- 1402, 1994.

[175] Schomaker.L and Segers.E., “A method for the determination of features used in human reading of cursive handwriting”, *Proc. 6th Int. workshop on Frontiers in Handwriting Recognition*, pp.157-168, 1998.

[176] Schurmann.J, “Pattern Classification – A unified view of statistical and neural approaches”, *Wiley Interscience*, 1996.

[177] Sekita.I, Toraichi.K, Mori.R , Yamamaoto.K and Yamada.H, “Feature extraction of handwritten Japanese characters by spline functions for relaxation matching”, *Pattern Recognition*, vo.1, pp.9-17, 1988.

[178] Sekuler.R and Blake.R, “Perception”, *3rd Ed., McGraw-Hill Inc.*, 1994.

[179] Senior.A.W and Robinson.A.J, “An off-line cursive handwriting recognition system”, *IEEE Trans. on PAMI*, vol.20, no.3, 1998.

[180] Sethi.I.K and Chatterjee.B, "Machine recognition of hand printed Devanagari numerals", *J. Inst. Elec. Telecom. Engng., India*, vol.22, pp .532-535 , 1976.

[181] Sethi.I.K, "Machine recognition of constrained hand printed Devanagari", *Pattern Recognition*, vol. 9, pp. 69-75, 1977.

[182] Sherman.H, "A quasi-topological method for the recognition of line patterns", *Proc. Int. Conf. on Information Processing, Paris, France*, pp.232-238, 1959.

[183] Shi.M, Fujisawa.Y, Wakabayashi.T and Kimura.F, "Handwritten numeral recognition using gradient and curvature of gray-scale image", *Pattern Recognition,* vol.35, pp.2051-2059, 2002.

[184] Simon.J.C, Backer.E and Sallantin.J, "A structural approach of pattern recognition", *Signal Processing*, vol.2, pp.5-22, 1980.

[185] Simpson.P.K, "Artificial neural systems", *Pergamon Press*, 1990.

[186] Sinha.R.M.K & Mahabala.H , "Machine recognition of Devanagari script", *IEEE Trans. Systems Man and Cybernetics*, vol.9, pp.435 – 449, 1979.

[187] Sinha.R.M.K, "Rule based contextual post-processing for Devanagari text recognition", *Pattern Recognition*, vol. 20, pp. 475 – 485, 1987.

[188] Siromaini.G, Chadrasekharan.R and Chandrasekharan.M, "Machine recognition of Brahmi script.", *IEEE Trans. Systems Man and Cybernetics*, vol.13, 1983.

[189] Siromoney.G, Chandrasekaran R and Chandrasekaran M, "Computer recognition of printed Tamil characters", *Pattern recognition*, vol.10, pp.243-247, 1978.

[190] Smith.R.W, "Computer processing of line images: A survey", *Pattern Recognition*, vol.20, no.1, pp.7-15, 1987.

[191] Solihin.Y and Leedham.C.G, "A new class of global thresholding techniques for handwriting images", *IEEE Trans. on PAMI*, vol.21, pp.761-768, 1999.

[192] Song.F and Smith.S.M, "Combination of adaptive-network-based fuzzy inference system and incremental best estimate directed search", *IEEE Int. Fuzzy Systems Conf.*, pp.392-395, 2001.

[193] Song.H.H, Lee.S.W, "A self-organizing neural tree for learning large-set pattern classification", *IEEE Trans. Neural Networks*, vol.9, no.3, pp.369-380, 1998.

[194] Srihari.S.N, Cohen.E, Hull.J.J and Kuan, "A system to locate and recognize ZIP codes in handwritten addresses", *IJRE*, vol.1, pp. 37-45, 1989.

[195] Srikantan.G, Lam.S.W and Srihari.S.N, "Gradient–based contour encoding for character recognition", *Pattern Recognition*, vol.29, no.7, pp. 1147-1160, 1996.

[196] Stefanelli.R and Rosenfeld.A, "Some parallel thinning algorithms for digital pictures", *Journal of ACM*, vol.18, no.2, pp.255-264, 1971.

[197] Suen.C.Y, Berhold.A.M and Mori.A.S, "Automatic recognition of handprinted characters-The state of the art", *Proc. IEEE*, vol.68, pp.469-187, 1980.

[198] Suen.C.Y, Guo.J and Li.Z.C, "Analysis and recognition of alphanumeric handprints by parts", *IEEE Trans. on Systems, Man and Cybernetics*, vol.24, pp.614-631, 1994.

[199] Suen.C.Y, Nadal.C, Legault.R, Mai.T.A and Lam.L, "Computer recognition of unconstrained handwritten numerals", *Proc. IEEE*, vol.8, no.7, pp.1162-1180, 1992.

[200] Sugeno.M and Kang.G.T, "Structure identification of fuzzy model", *Fuzzy sets and systems*, vol.28, pp.15-33, 1988.

[201] Sukhaswami.M.B and Seetharamula.P and Poojari.A.K, "Recognition of Telugu characters using neural networks", *Int. Journal of Neural Systems*, vol.6, no.3, 1995.

[202] Sushmita.M and Pal.S.K, "Fuzzy multi-layer perceptron inferencing and rule generation", *IEEE Trans. Neural Networks*, vol.6, no.1, 1995.

[203] Suzuki.T and Abe.K, "Binary picture thinning by an interactive parallel two sub-cycle operation", *Pattern Recognition*, vol.10, no.3, pp.297-307, 1987.

[204] Takagi.T and Sugeno.M, "Structure identification of systems and its application to modeling and control", *IEEE Trans. System Man Cybernetics*, vol.15, pp.116- 132, 1985.

[205] Takagi.T and Sugeno.M, "Structure identification of systems and its application to modeling and control", *IEEE Trans. System Man and Cybernetics*, vol. 15, pp. 116- 132, 1985.

[206] Takahashi.H, "A neural net OCR using geometrical and zonal pattern features", *Proc. of First Int. Conf. on Document Analysis and Recognition, Saint-Malo, France*, pp.821-828, 1991.

[207] Takens.F, "Detecting strange attractors in turbulence", *Lecture notes in Mathematics, Editors-Rand.D.A and Young.L.S, Springer, Berlin*, vol.898, 1981.

[208] Tamura H., "A comparison of line thinning algorithms from a digital geometry view point", *Proc. 4th Int. Conf. Pattern Recognition, Kyoto, Japan*, pp.715-719, 1978.

[209] Toriwaki.J.I and Yokoi.S, "Distance transformation and skeletons of digitized pictures with applications", *Progress in Pattern Recognition, Editors- Kandel.L.N and Rosenfeld.A, Amsterdam, Netherlands*, pp.187-265, 1981.

[210] Tou.J.T and Gonzalez.R.C, "Pattern recognition principles", *Addison-Wesley, London*, 1974.

[211] Triendl.E.E, "Skeletonization of noisy hand-drawn symbols using parallel operations", *Pattern Recognition*, vol.2, pp.215-226, 1970.

[212] Trier O.D and Jain A.K, "Goal directed evaluation of binarization methods", *IEEE Trans. on PAMI*, vol.17, pp.1191-1201, 1995.

[213] Trier.O.D and Taxt.J, "Evaluation of binarization methods for document images", *IEEE Trans. on PAMI*, vol.17, no.3, pp.312-315, 1995.

[214] Trier.O.D Jain.A.K and Taxt.J, "Feature extraction methods for character recognition – A survey", *Pattern Recognition*, vol.29, no.4, pp.641-662, 1996.

[215] Tsukumo.J and Tanaka.H, "Classification of handprinted Chinese characters using nonlinear normalization methods", *9th International Conf. on Pattern Recognition*, pp-168-171, 1998.

[216] Tucker.N.D and Evans.F.C, "A two-step strategy for character recognition using geometrical moments", *Proc. 2nd Int. Conf. Pattern Recognition*, pp. 223-225, 1974.

[217] Ugawa.K, Toriwaki.J and Sugino.K, "Normalization and recognition of two-dimensional patterns and linear distortion by moments", *Electron. Comm. Jap.*, pp.34-46, 1967.

[218] Vapnik.V, "The nature of statistical learning theory", *Springer-Verlag, New York, USA*, 1995.

[219] Venishetti.K, Rathod.A.K and Bhaskar.P.V, "Moment of inertia based radial coding features of invariant character recognition using fuzzy min-max neural networks", *IEEE 1st Int. conf. on Signal and Image Processing, Hubli, India*, vol.1, pp.168-173, 2006.

[220] [1]Wang.L and Pavlidis.T, "Detection of curved and straight segments from gray-scale topography", *CVGIP: Image Understanding*, vol.58, pp.352-365, 1993.

[221] [2]Wang.L and Pavlidis.T, "Direct gray-scale extraction for character recognition", *IEEE Trans. PAMI*, vol.15, no.10, pp.1053-1067, 1993.

[222] Wang.X and Ding.X, "A gray-scale image based character recognition algorithm to low quality and low resolution images", *Document Recognition and Retrieval VIII, Electronic Imaging, San Jose, CA, USA*, 2001.

[223] Wang.X, Ding.X and Liu.C, "Gabor filters-based feature extraction for character recognition", *Pattern Recognition*, vol.38, pp.369-379, 2005.

[224] Werbos.P, "Beyond regression : New tools for prediction and analysis in the behavioral sciences", *Ph.D. Dissertation, Harvard University, Cambridge, MA*, 1974

[225] Wolberg.G, "Digital image warping", *IEEE Computer Society Press, Los Alamitos, California*, 1990.

[226] Xie.S.L and Suk.M, "On machine recognition of hand-printed Chinese character by feature relaxation", *Pattern Recognition*, vol.21, no.1, pp.1-7, 1988.

[227] Xu.L. Krzyzak.A and Suen.C.Y, "Methods of combining multiple classifiers and their applications to handwriting recognition", *IEEE Trans. on Systems Man, and Cybernetics*, vol.22, pp.418-435, 1992.

[228] Yamamoto.K and Mori.S, "Recognition of hand printed characters by outermost point method", *Proc. 4th Int. Conf. Pattern recognition, Kyoto, Japan*, pp.794-796, 1978.

[229] Yanowitz.S.D. and Bruckstein.A.M, "A new method for image segmentation", *Computer Vision, Graphics and Image Processing*, vol.46, no. 1, pp.82-85, 1989.

[230] Yokoi.S, Toriwaki.J-I. and Fukumura.T, "Topological properties in digitized binary pictures", *Systems Comput.Contr.*, vol.4, no.6, pp.32-39, 1973.

[231] Zhang.B and Srihari.S.N, "Fast k-Nearest Neighbor classification using cluster-based trees", *IEEE Trans. on PAMI*, vol.26, no.4, pp.525-528, 2004.

[232] Zhang, Fu.M, Yan.H., and Fabri.M.A, "Handwritten digit recognition by adaptive subspace self-organizing map", *IEEE Trans. on Neural Networks*, vol.10, pp.939-945, 1999.

[233] Zhang.G.P, "Neural networks for classification: A survey", *IEEE Trans. on Systems, Man, and Cybernetics-Part C: Applications and Reviews*, vol.30, no .4, pp. 451-462, 2000.

[234] Zhang.T.Y and Suen.C.Y, "A fast parallel algorithm for thinning digital Patterns", *Comm. ACM*, vol.27, no.3, pp.236-239, 1984.

[235] Zuniga.O.A and Haralick.R.M, "Corner detection using the facet model", *Proc. IEEE Conf. on Computer Vision and Pattern Recognition, Washington DC*, 1983.

LIST OF PUBLICATIONS OF THE AUTHOR

[1] **Lajish.V.L**, Kabeer.V and Narayanan.N.K, "Handwritten character recognition using gray-scale based state-space parameters and *k*-NN classifier", *Proc. of the IEEE - International Conference on Signal and Image Processing, (ICSIP-2006), Hubli, India*, vol.1, pp.196-201, 2006.

[2] **Lajish.V.L**, Thasleema.T.M and Narayanan.N.K, "Isolated word–speech recognition using lexicon based Hidden Markov Models", *Proc. of the IEEE - International Conference on Signal and Image Processing, (ICSIP-2006), Hubli, India*, vol.2, pp.988-991, 2006.

[3] **Lajish.V.L** and Narayanan.N.K, "Hidden Markov Model and domain specific (Heuristic) knowledge for recognition of isolated word images", *Proc. Sixth National Conference on Recent Trends in Advanced Computing, (NCRTAC-06), Thirunelveli, India*, pp.9-15, 2006.

[4] **Lajish.V.L**, Shaju.K.K and Narayanan.N.K, "Use of handwriting feature descriptions and artificial neural networks to recognize isolated character images", *Proc. Sixth National Conference on Recent Trends in Advanced Computing, (NCRTAC-06), Thirunelveli, India*, pp.16-19, 2006.

[5] **Lajish.V.L**, Tasleema.T.M and Narayanan.N.K, "Hidden Markov Models for isolated word-speech recognition based on large vocabulary", *Proc. of the National Symposium on Acoustics, (NSA2006), NPL, New Delhi*, 2006.

[6] Sunil Kumar.R.K, **Lajish.V.L** and Narayanan.N.K, "Recognition of vowel and Consonant Vowel (CV) units of Malayalam speech using time domain features and artificial neural networks", *Proc. International Conference on Frontiers of Research on Speech & Music, (FRSM-2006), Uttar Pradesh, India*, pp.58-63, 2006.

[7] Sunil Kumar.R.K, **Lajish.V.L** and Narayanan.N.K, "Vowel phoneme recognition using average energy in the Zerocrossing interval distributions", *Proc. International Conference on Frontiers of Research on Speech & Music, (FRSM-2006), Uttar Pradesh, India*, pp.71-78, 2006.

[8] **Lajish.V.L**, Annapurnewari.K.K and Narayanan.N.K, "Recognition of handwritten word images using lexicon based word modeling and A* algorithm", *Proc. International Conference on Cognition and Recognition, (ICCR-'05), Mandya, India*, pp-581-588, 2005.

[9] **Lajish.V.L**, Suneesh.T.T.K and Narayanan.N.K, "Recognition of isolated handwritten character images using Kolmogrov-Smirnov statistical classifier and *k*-Nearest Neighbour classifier", *Proc. International Conference on Cognition and Recognition, (ICCR-'05), Mandya, India,* pp.526-531, 2005.

[10] Sunil Kumar.R.K, **Lajish.V.L** and Narayanan.N.K, " Zerocrossing based speech pattern recognition using Kolmogrov-Smirnov statistical test and artificial neural networks", *Proc. International Conference on Frontiers of Research on Speech & Music, (FRSM-2005), Bhuvaneshwar, India*, pp.90-98, 2005.

[11] **Lajish.V.L** and Narayanan.N.K, "Computer recognition of off-line handwritten document images - A soft computing perspective", *Proc. National Conference on Geometry, Analysis, Fluid Mechanics and Computer Applications, Kuvempu University Karnataka*, 2004.

[12] Sunil Kumar.R.K, **Lajish.V.L** and Narayanan.N.K, "Artificial neural network classifier for recognizing Malayalam consonant-vowel units", *Proc. National Seminar on Artificial Intelligent and Neural Networks, Centre for Mathematical studies & Computer Applications, Union Christian College, Alwaye, Kerala*, 2004.

[13] Sunil Kumar.R.K, **Lajish.V.L**, Sureshbabu.K, Ramakrishnan.P, and Narayanan.N.K, "Malayalam vowel phoneme recognition using Zerocrossing based distribution patterns", *Proc. First National Symposium on Physics, Kanhangad, Kerala*, vol.1, pp.30-41, 2004.

[14] Rajeevan.N.E, Sunil Kumar.R.K, **Lajish.V.L** and Narayanan.N.K, "Time duration analysis of Malayalam syllables involving selected consonants for computer speech recognition applications", *Journal of Acoustical Society of India*, vol.31, no.1-4, pp.175-177, 2003.

[15] **Lajish.V.L**, Prajeesh.T, Sunil Kumar.R.K, and Narayanan.N.K, "Noise spectrum analysis in transport vehicles for active noise control applications", *Journal of Acoustical Society of India*, vol.31, no.1-4, pp.270-272, 2003.

[16] Sunilk Kumar.R.K, **Lajish.V.L** and Narayanan.N.K, "Vowel phoneme recognition using Zerocrossing based distribution patterns and artificial neural networks", *Proc. National Conference on Signal Processing, Intelligent systems and Networking, (SPIN-2003), Banagalore*, vol.1, pp.49-54, 2003.