

# Handwritten Gujarati Character Recognition using Machine Learning Approach

A Thesis Submitted to  
Nirma University

In Partial Fulfillment of the Requirements for  
The Degree of  
Doctor of Philosophy

in  
Technology and Engineering  
by

Ankit Sharma (12EXTPHDE93)



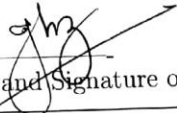
Institute of Technology  
Nirma University  
Ahmedabad-382481  
Gujarat, India  
November 2017

**Nirma University**  
**Institute of Technology**  
**Certificate**


This is to certify that the thesis entitled "Handwritten Gujarati Character Recognition using Machine Learning Approach" has been prepared by Mr. Ankit Sharma under my supervision and guidance. The thesis is his original work completed after careful research and investigation. The work of the thesis is of the standard expected of a candidate for Ph.D. Programme in Engineering and I recommend that it be sent for evaluation.


Date: 17/05/18


  
\_\_\_\_\_  
Name and Signature of the Guide

  
\_\_\_\_\_  
Name and Signature of the Co-Guide

Forwarded Through:

  
\_\_\_\_\_  
Name and Signature of the  
Head of the Department

  
\_\_\_\_\_  
Name and Signature of the  
Dean Faculty of Technology  
and Engineering

  
\_\_\_\_\_  
Name and Signature of the  
Dean Faculty of Doctoral Studies  
and Research

To:  
Executive Registrar  
Nirma University

# Abstract

Handwritten character recognition is an active area of research. Over the past three decades, there has been increasing interest among researchers in problem related to the machine simulation of the human reading process. Optical Character Recognition (OCR) is the tool that is utilized to convert printed or handwritten scanned document into machine readable form/text. Handwritten character recognition is a challenging task and people are striving to convert handwritten literature to computer readable format. Recognising handwritten characters is difficult compared to printed characters because handwritten characters may vary from person to person with respect to the individual writing style, size, curve, strokes and thickness of characters.

Languages have played a major role in Indian history and they continue to influence the lives of the Indians till date. Plentiful research on OCR techniques for Indian languages such as Hindi, Tamil, Bangla, Kannada, Gurumukhi and Malayalam has already been carried out. Development of OCR systems for Gujarati script is still in infancy and hence, there exists many unaddressed challenging problems for research community in this domain. This clearly necessitates the need to attend the task of handwritten Gujarati character recognition. This thesis addresses the issues of handwritten Gujarati character recognition.

Gujarati is the mother tongue of people belong to Gujarat state in India. All over the world more than 65 million people use Gujarati language for their communication purpose. As Gujarat is one of the eminent state of India, Gujarati is a well-known and culturally rich language. Gujarati Character Recognition offers more difficulties like the most other Indian languages relative to the western languages due to these reasons: (a) number of classes are higher, (b) structure of characters in Gujarati script contains curves, holes and strokes which result in significant variations in writing style of different persons, (c) presence of similar looking characters (d) unavailability of

standard dataset for experimentation and validation.

One of the significant contributions of proposed work is towards the development of large and representative datasets for the task of recognising handwritten Gujarati characters and numerals. Benchmark datasets having 88,000 handwritten Gujarati character images and 14,000 handwritten Gujarati numeral images are developed. Special forms are utilized for dataset collection and isolated characters are extracted from these forms. Preprocessing steps including noise removal, size normalization, binarization and thinning are applied on each segmented numeral/character image. Systematic and exhaustive experiments are carried out on these developed datasets using different kinds of features and their fusion. Zone based, projection profiles based and chain code based features are employed as individual features. It is also proposed to use the fusion of these features. Few novel features are also proposed to represent handwritten Gujarati characters. These features include features extracted based on structural decomposition, zone pattern matching and normalized cross correlation. Methods based on artificial neural network (ANN), support vector machine (SVM) and naive Bayes (NB) classifier are used for handwritten Gujarati character and numeral recognition. In case of individual features, chain code based features provided higher recognition accuracy values compared to other features which were 99.25% and 99.47% with polynomial SVM for numerals and characters datasets respectively. In case of fusion based features, fusion of chain code based and zoning based features provided best results compared to other fusion based features. Proposed structural decomposition based features provided highest accuracy of 99.48% with polynomial SVM for handwritten characters. Experimental results show significant improvement over state-of-the-art and validate our proposals.

## Nirma University Institute of Technology Declaration

I, Mr. Ankit Sharma, registered as Research Scholar, bearing Registration Number 12EXTPHDE93 for Doctoral Programme under the Faculty of Technology and Engineering of Nirma University do hereby declare that I have completed the course work, pre-synopsis and my research work as prescribed under R. Ph. D. 3.5.

I do hereby declare that the thesis submitted is original and is the outcome of the independent investigations / research carried out by me and contains no plagiarism. The research is leading to the discovery of new techniques already known. This work has not been submitted by any other University or Body in quest of a degree, diploma or any other kind of academic award.

I do hereby further declare that the text, diagrams or any other material taken from other sources (including but not limited to books, journals and web) have been acknowledged, referred and cited to the best of my knowledge and understanding.

Date: 17/05/18



Signature of Student

---

I agree with the above declaration made by the student.

Date: 17/05/18



Signature of the Guide

  
Signature of the Co-Guide

# Acknowledgements

I take the pleasure to present this research work related to “Handwritten Gujarati Character Recognition using Machine Learning Approach” to the Almighty, for being present in all my endeavors.

I would like to thank Dr. Dipak M. Adhyaru, Guide and Head, Instrumentation and Control Engineering Department, Institute of Technology, Nirma University, Ahmedabad, and Dr. Tanish H. Zaveri, Co-Guide and Professor, Electronics and Communication Engineering Department, Institute of Technology, Nirma University, Ahmedabad for motivating me and providing continuous support throughout my Doctoral studies.

My sincere thanks to the reviewers Prof. Mukesh Zaveri and Dr. Suryakant B. Gupta, who had given their valuable feedback during my Research Progress Committee meetings.

I would like to express my sincere regards to Dr. Alka Mahajan, Director, Institute of Technology, Nirma University, Ahmedabad for her never-ending support and cooperation.

I can not forget to thank Dr. Priyank B Thakkar, who helped me a lot in shaping the draft of the thesis. There are no words to thank the Almighty for gifting me with a wonderful child, Vihaan, who bore the wrath of my journey towards this research. My wife Durga needs special accolade for her patience and constant support provided throughout this work. Finally, my sincere most thanks to the most important and special people of my life - my parents; who have been ever motivating, endearing and highly cooperative in all my endeavors. I thank one and all, who have kept encouraging and motivating me.

**Ankit Sharma**  
**12EXTPHDE93**

# Contents

<b>Certificate</b>	<b>iii</b>
<b>Declaration</b>	<b>v</b>
<b>Abstract</b>	<b>vi</b>
<b>Acknowledgements</b>	<b>viii</b>
<b>List of Tables</b>	<b>xiii</b>
<b>List of Figures</b>	<b>xv</b>
<b>Abbreviations</b>	<b>xvii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Overview of Handwritten Character Recognition System . . . . .	1
1.2 Applications . . . . .	3
1.3 Problem Statement . . . . .	4
1.4 Challenges in designing Handwritten Character Recognition System for Gujarati Script . . . . .	4
1.5 Components of an Handwritten Character Recognition System . . . . .	5
1.5.1 Digitization . . . . .	6
1.5.2 Segmentation . . . . .	6
1.5.3 Preprocessing . . . . .	6
1.5.4 Feature Extraction . . . . .	6
1.5.5 Classification . . . . .	7
1.6 Motivation . . . . .	7
1.7 Gujarati Script from OCR Perspective . . . . .	9
1.8 Objectives . . . . .	12
1.9 Thesis Contributions . . . . .	13
1.10 Organization of the Thesis . . . . .	13
<b>2 Literature Survey</b>	<b>15</b>
2.1 Literature Survey on Gujarati Character Recognition . . . . .	16
2.1.1 Printed Gujarati Character Recognition . . . . .	16
2.1.2 Handwritten Gujarati Character Recognition . . . . .	18
2.2 Literature Survey on Devanagari Character Recognition . . . . .	20
2.2.1 Printed Devanagari Character Recognition . . . . .	20
2.2.2 Handwritten Devanagari Character Recognition . . . . .	22

2.3	Literature Survey on English character recognition . . . . .	24
<b>3</b>	<b>Handwritten Gujarati Dataset Development</b>	<b>27</b>
3.1	Benchmark Databases for HCR Research . . . . .	28
3.2	Handwritten Character Dataset Development . . . . .	29
3.3	Handwritten Numeral Dataset Development . . . . .	32
3.4	Segmentation . . . . .	35
3.5	Preprocessing . . . . .	36
3.5.1	Noise Removal . . . . .	37
3.5.2	Binarization . . . . .	38
3.5.3	Size Normalization . . . . .	38
3.5.4	Thinning . . . . .	38
3.6	Summary . . . . .	39
<b>4</b>	<b>Feature Extraction Techniques</b>	<b>41</b>
4.1	Feature Set Extraction based on Zoning . . . . .	42
4.2	Feature Set Extraction based on Projection Profiles . . . . .	44
4.3	Feature Set Extraction based on String of Chain Code . . . . .	44
4.4	Feature Set Extraction using Fusion of Features . . . . .	47
4.5	Feature Set Extraction using Structural Decomposition Technique . . . . .	48
4.6	Feature Set Extraction by Zone Pattern Matching . . . . .	50
4.7	Feature Set Extraction by Normalised Cross Correlation . . . . .	51
4.8	Summary . . . . .	56
<b>5</b>	<b>Machine Learning Approaches</b>	<b>57</b>
5.1	Naive Bayes Classifier . . . . .	57
5.2	Support Vector Machine . . . . .	59
5.3	Artificial Neural Network . . . . .	61
5.4	Summary . . . . .	63
<b>6</b>	<b>Results and Discussion</b>	<b>65</b>
6.1	Experimental Evaluation . . . . .	65
6.1.1	Performance Evaluation using SoCC, ZbF and PPbF . . . . .	67
6.1.2	Performance Evaluation using Fusion Features . . . . .	70
6.1.3	Performance Evaluation using SD, ZPM and NCC . . . . .	92
6.2	Summary . . . . .	95
<b>7</b>	<b>Conclusions and Future Work</b>	<b>97</b>
7.1	Conclusions . . . . .	97
7.2	Future Scope . . . . .	98
	<b>Works Cited</b>	<b>100</b>
	<b>Publications related to Thesis</b>	<b>111</b>



# List of Tables

2.1	Research efforts in the area of printed Gujarati character recognition	18
2.2	Research efforts in the area of handwritten Gujarati character recognition	19
3.1	Handwritten Gujarati datasets . . . . .	35
4.1	Feature sets . . . . .	47
4.2	Patterns & their representation and value . . . . .	52
4.3	Image combinations chosen for feature vector generation . . . . .	54
5.1	SVM design parameters and their values tested in cross validation of training set . . . . .	61
5.2	ANN design parameters and their values tested in cross validation of training set . . . . .	63
6.1	Performance of prediction models on HGND with SoCC, PPbF and ZbF features . . . . .	68
6.2	Performance of prediction models on HGCD-1 with SoCC, PPbF and ZbF features . . . . .	69
6.3	Accuracy(%) values obtained by prediction models with SoCC on individual numerals (on HGND) . . . . .	71
6.4	Accuracy(%) values obtained by prediction models with ZbF on individual numerals (on HGND) . . . . .	71
6.5	Accuracy(%) values obtained by prediction models with PPbF on individual numerals (on HGND) . . . . .	72
6.6	Accuracy(%) values obtained by prediction models with SoCC on individual characters (on HGCD-1) . . . . .	73
6.7	Accuracy(%) values obtained by prediction models with ZbF on individual characters (on HGCD-1) . . . . .	74
6.8	Accuracy(%) values obtained by prediction models with PPbF on individual characters (on HGCD-1) . . . . .	75
6.9	Performance of prediction models on HGND with individual features and fusion features . . . . .	80
6.10	Performance of prediction models on HGCD-1 with individual features and fusion features . . . . .	81
6.11	Accuracy(%) values obtained by prediction models with CP on individual numerals (on HGND) . . . . .	82
6.12	Accuracy(%) values obtained by prediction models with CZ on individual numerals (on HGND) . . . . .	83

6.13	Accuracy(%) values obtained by prediction models with PZ on individual numerals (on HGND) . . . . .	83
6.14	Accuracy(%) values obtained by prediction models with CPZ on individual numerals (on HGND) . . . . .	84
6.15	Accuracy(%) values obtained by prediction models with CP on individual characters (on HGCD-1) . . . . .	85
6.16	Accuracy(%) values obtained by prediction models with CZ on individual characters (on HGCD-1) . . . . .	86
6.17	Accuracy(%) values obtained by prediction models with PZ on individual characters (on HGCD-1) . . . . .	87
6.18	Accuracy(%) values obtained by prediction models with CPZ on individual characters (on HGCD-1) . . . . .	88
6.19	Performance of prediction models on HGCD-2 with SD, ZPM, NCC features . . . . .	92
6.20	Misclassified character images . . . . .	96

# List of Figures

1.1	Overall architecture of proposed handwritten character recognition system . . . . .	6
1.2	The Brahmic family of scripts used in India . . . . .	10
1.3	Gujarati consonants and vowels . . . . .	11
1.4	Zone separation for Gujarati characters . . . . .	12
1.5	Gujarati numerals and corresponding English numerals . . . . .	12
3.1	A sample filled form utilized for character dataset collection . . . . .	30
3.2	Gujarati characters considered for HGCD-1 and corresponding classes assigned . . . . .	31
3.3	Gujarati characters considered for HGCD-2 and corresponding classes assigned . . . . .	32
3.4	A sample filled form for numerals . . . . .	33
3.5	Variation in writing style of the same person at different instances of time . . . . .	33
3.6	Gujarati numerals considered for HGND and corresponding classes assigned . . . . .	34
3.7	(a) to (j) - indicating variations in writing styles of different writers for digits 0 to 9 respectively . . . . .	34
3.8	Preprocessing steps for segmented character images . . . . .	37
4.1	Numeral image divided in (a) 256 zones (b) 64 zones (c) 16 zones (d) 4 zones . . . . .	43
4.2	(a) represents horizontal profile. (b) represents vertical profile. (c) and (d) represent left diagonal and right diagonal profiles respectively. . . . .	45
4.3	Chain code obtained by finding the starting point through horizontally scanning for numeral images. . . . .	46
4.4	Chain code obtained by finding the starting point through horizontally scanning for character images. . . . .	47
4.5	Gujarati handwritten character recognition process using fusion features	48
4.6	Constitutional components of one of the Gujarati character . . . . .	49
4.7	Features extracted from component 1 image . . . . .	50
4.8	Original, Resized and various Rotated and Flipped versions of Image Halves . . . . .	55
5.1	Neural network architecture . . . . .	62
6.1	Handwritten Gujarati character recognition process . . . . .	66

6.2	Accuracy values averaged over prediction models learnt using SoCC, PPbF and ZbF features (on HGND) . . . . .	69
6.3	Accuracy values averaged over prediction models learnt using SoCC, PPbF and ZbF features (on HGCD-1) . . . . .	70
6.4	Accuracy values averaged over prediction models learnt using SoCC features for individual numerals (on HGND) . . . . .	72
6.5	Accuracy values averaged over prediction models learnt using ZbF for individual numerals (on HGND) . . . . .	76
6.6	Accuracy values averaged over prediction models learnt using PPbF for individual numerals (on HGND) . . . . .	76
6.7	Accuracy values averaged over prediction models learnt using SoCC features for individual characters (on HGCD-1) . . . . .	77
6.8	Accuracy values averaged over prediction models learnt using ZbF for individual characters (on HGCD-1) . . . . .	77
6.9	Accuracy values averaged over prediction models learnt using PPbF for individual characters (on HGCD-1) . . . . .	77
6.10	Accuracy values averaged over prediction models learnt using fusion features (on HGND) . . . . .	79
6.11	Accuracy values averaged over prediction models learnt using fusion features (on HGCD-1) . . . . .	79
6.12	Accuracy values averaged over prediction models learnt using PC features for individual numerals (on HGND) . . . . .	84
6.13	Accuracy values averaged over prediction models learnt using ZC features for individual numerals (on HGND) . . . . .	89
6.14	Accuracy values averaged over prediction models learnt using PZ features for individual numerals (on HGND) . . . . .	89
6.15	Accuracy values averaged over prediction models learnt using ZPC features for individual numerals (on HGND) . . . . .	90
6.16	Accuracy values averaged over prediction models learnt using PC features for individual characters (on HGCD-1) . . . . .	90
6.17	Accuracy values averaged over prediction models learnt using ZC features for individual characters (on HGCD-1) . . . . .	91
6.18	Accuracy values averaged over prediction models learnt using PZ features for individual characters (on HGCD-1) . . . . .	91
6.19	Accuracy values averaged over prediction models learnt using ZPC features for individual characters (on HGCD-1) . . . . .	91
6.20	Accuracy values averaged over prediction models learnt using SD, ZPM and NCC features (on HGCD-2) . . . . .	93
6.21	Accuracy values averaged over prediction models learnt using SD features for individual characters (on HGCD-2) . . . . .	93
6.22	Accuracy values averaged over prediction models learnt using ZPM features for individual characters (on HGCD-2) . . . . .	94
6.23	Accuracy values averaged over prediction models learnt using NCC features for individual characters (on HGCD-2) . . . . .	94
6.24	Sets of similar looking Gujarati characters . . . . .	94

# List of Abbreviations

ANN	Artificial Neural Network
BPNN	Back-propagation Neural Network
BVoDP	Best Values of Design Parameters
CASIA	Institute of Automation, Chinese Academy of Sciences
CEDAR	Center of Excellence for Document Analysis and Recognition
CENPARMI	Centre for Pattern Recognition and Machine Intelligence
CP	SoCC + PPbF
CPZ	SoCC + PPbF + ZbF
CZ	SoCC + ZbF
ETL	Electro-Technical Laboratory
FN	False Negative
FP	False Positive
GHIC	Generalized Hausdorff Image Comparison
GRNN	Generalized Regression Neural Network
HCR	Handwritten Character Recognition
HGCD-1	Handwritten Gujarati Character Dataset-1
HGCD-2	Handwritten Gujarati Character Dataset-2
HGND	Handwritten Gujarati Numeral Dataset
HMM	Hidden Markov Model
HOG	Histogram of Oriented Gradients
HP	Hewlett Packard
KNN	K-Nearest Neighbor
MCE	Minimum Classification Error
MKL-SVM	Multiple Kernel Learning based Support Vector Machine
MLP	Multilayer Perceptron
MNIST	Modified National Institute of Standards and Technology
MQDF	Modified Quadratic Discriminant Function
MSE	Mean Square Error
NB	Naive Bayes
NCC	Normalized Cross Correlation
OCR	Optical Character Recognition
PCA	Principal Component Analysis
PPbF	Projection Profile based Features
PZ	PPbF + ZbF
RBF	Radial Basis Function
SD	Structural Decomposition
SoCC	String of Chain Code
SOM	Self Organizing Map

TN ..... True Negative  
TP ..... True Positive  
SVM ..... Support Vector Machine  
ZbF ..... Zone based Features  
ZIP ..... Zone Improvement Plan  
ZPM ..... Zone Pattern Matching

# Chapter 1

## Introduction

The character recognition has been the most challenging and widely studied topic during recent decades. The rapid development of digital libraries worldwide has given rise to many challenges in the area of document image analysis research and development. One of the major challenges is to convert the scanned document to a textual representation through character recognition technology. Character recognition system translates typewritten, handwritten, scanned copy of images or documents to machine encoded text. This translated machine encoded text can easily be stored, searched, edited and processed in numbers of ways as per our requirements.

### 1.1 Overview of Handwritten Character Recognition System

Development of Optical Character Recognition (OCR) technologies are aimed to convert printed or handwritten characters from flat physical material to digital form such that it should produce machine-readable digital content. Handwritten character recognition is a process of identifying characters from handwritten scanned documents. Development of character recognition algorithm for printed and handwritten characters is a challenging area for research.

There are two approaches for the recognition of handwritten characters, viz. online and offline, depends on the way characters are written. In former case, a special pen is used to write on electronic surface and while writing, writer's pen movement, velocity, accelerations and stroke order are recorded. In case of offline

character recognition, the typewritten/handwritten characters are scanned from a paper document and made available in the form of a binary/gray-scale image to recognition system. In this case, the information available is only in the form of set of pixels which exacerbate the problem. Hence, the recognition rate reported in case of offline character recognition is lower as compare to online character recognition (Plamondon and Srihari).

This thesis presents work based on offline character recognition. Handwritten character recognition helps to reduce human efforts of manually handling and processing documents. Handwritten character recognition system can convert scanned handwritten document into text format for storage, editing, transmission, indexing and coordinating in different applications. For researchers, handwritten character recognition has turned into an extremely challenging as well as appealing topic (Suen, Berthod, and Mori Mori, Suen, and Yamamoto Cheriet et al., *Character recognition systems: a guide for students and practitioners*).

Even with the introduction of new technologies like keypad, writing pad etc, various communication and information collection tasks are done in handwritten form in day to day life due to its easiness. In spite of enormous efforts towards building a paperless work environment and society, we continue to see a lot of documentation work is done with the use of paper and pen (Impedovo). Moreover, writing and drawing on paper appears to be the most direct and natural way of expressing the ideas of our brain as compared to the use of electronic means (Cheriet et al., “Handwriting recognition research: Twenty years of achievement and beyond”).

Because of growth in information and communications technologies, requirement of character recognition tools for Indian scripts has increased. However, it is to be noticed that development of character recognition tools for Indian scripts is more challenging since the shapes of characters in Indian scripts are more complex from the OCR perspective. As mentioned earlier, this thesis focuses on offline recognition of handwritten Gujarati characters. Handwritten Character Recognition (HCR) is a challenging area of research as writing style varies from person to person and even for the same person it varies depending on the speed, mood and the environment. HCR is not as simple task as it might seems, even the eyes of human being makes mistake of 4% when reading in the absence of context (Mantas). Shapes of the



characters varies depending on the writing habit, style, education, region of origin, social environment, mood, health and other conditions of the writer. These factors are the main contributors to errors in reading handwritten characters. Efficiency of character recognition system is also affected by factors such as writing surface, writing pen, scanning method and most prominently by the character recognition algorithm (Mantas). In a nutshell, it can be said certainly that developing an HCR system is a conundrum for researchers.

## 1.2 Applications

Handwritten character recognition is a prime area of research because of its potential use in various applications. It plays a vital role to enhance automation process. An HCR system can facilitate automatic processing of postal codes and addresses, bank cheques, tax and admission forms, post cards, reservation forms and vehicle number plates. Data from these forms can be collected in large volume and automatic processing of these data are of a great significance. By associating the handwritten character recognition system to text to speech convertor it can assist visually impaired persons for reading (Tzanakou).

Gujarati is the official language in the Gujarat state. Many government letters and forms are in Gujarati script. In office environment various data are collected in handwritten form. These data can be very useful for future reference if it can be auto stored in editable format in computer. Many government and private organisations hire special employees to convert handwritten documents into digital format. By using HCR system information collected through this forms can be stored in the digital format. This process is cost effective and time consuming. An HCR system can certainly serve as an ideal solution to these problems.

Address information is written in handwritten form on post cards. HCR system can be used for identification and sorting of the post cards on the basis of place, address, pin code and city. The historical manuscripts written in Gujarati script can be preserved from natural degradation with time by converting them in digital format. It will help to enrich the historical culture across India.

The concept of paperless office can only be realized by transforming the documents digitally. Documents and files that were stored physically are now being

transformed into electronic form in order to facilitate quicker additions, searches, and updates. This also enables long life of such records. However, a large amount of business credentials and communication are still taking place in physical form. In a summary, it can be said that there is a definite need of software which can automatically extract, analyse, and store information from physical documents for later retrieval.

### 1.3 Problem Statement

The thesis focuses on the development of efficient offline handwritten character recognition algorithms for isolated handwritten Gujarati characters in a writer independent environment. It is important to note that effectiveness of features and accuracy of recognition techniques are script dependent, i.e. one set of features and technique may provide accurate results for few scripts but may give erroneous results for the others. The thesis also aims at execution of experimentation on a large scale in order to validate the suitability of features and recognition techniques for Gujarati script. Large and representative datasets are developed for the task of recognizing handwritten Gujarati numerals and characters. Systematic and exhaustive experimentations are performed with proposed techniques on developed datasets.

### 1.4 Challenges in designing Handwritten Character Recognition System for Gujarati Script

There are several issues which make the recognition of handwritten Gujarati characters a challenging task and affect the recognition rate to a considerable extent. The major challenges which HCR system for Gujarati script need to address are discussed below.

**Large character set:** Indian scripts like Devanagari and Gujarati have more number of distinct characters than that in European languages. These large set of distinct characters results in increase in number of classes to classify. Existence of modifiers in Gujarati script makes the recognition task further difficult. Due to existence of modifiers in Gujarati script, it becomes difficult to segment the core characters before applying the recognition algorithm.

**Large number of similar looking characters:** Several similar looking cha-

characters exist in Gujarati script (Chaudhary et al.). It is difficult to recognise them when separated from context. Existence of similar looking characters in Gujarati script also affect the recognition rate to great extent.

**Non availability of benchmark dataset:** Unavailability of an effective and representative Benchmark dataset for handwritten Gujarati characters is one of the major obstacle for the research in the area of handwritten character recognition.

**Extreme variability in the writing style of different writers:** Depending on the locality, educational background, profession and age, large variation is obtained in collected dataset of handwritten characters of Gujarati. Writing style of the person also varies if data are collected at different time instances depending on the mood of the writer. Variations are also possible due to variation in the thickness of the tip of writing pen as well as the colour and quality of ink used. Spacing between the consecutive handwritten characters, skew and slant also affect the performance of character segmentation and recognition algorithm. While writing many time it occurs that two consecutive characters/symbol touch each other, which creates problem in segmentation and then further processing for recognition.

## 1.5 Components of an Handwritten Character Recognition System

The different steps involved in the proposed character recognition system include digitization, preprocessing, feature extraction, feature fusion and classification. Optimization of all the steps is needed in order to achieve the best possible performance. The method adapted for preprocessing affects the feature extraction method and the subsequent classification process. In order to address the problem of HCR, it is required to develop effective feature descriptor and classifier. While designing the recognition system the variability in shapes of characters, handwriting style and complexity of shapes need to be considered. Overall architecture of the proposed system is depicted in Figure 1.1.

### 1.5.1 Digitization

Collection of a good database which represents wide variation of handwriting style and also includes all important classes of the script is a challenging task. In order

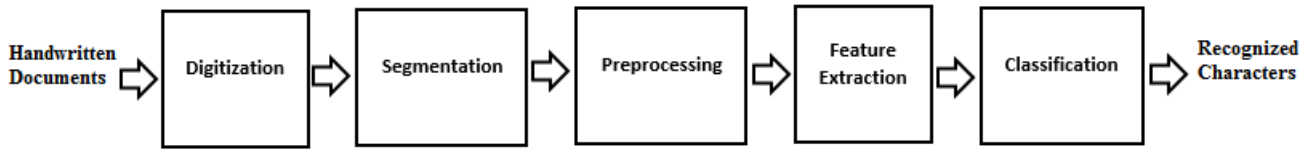


Figure 1.1: Overall architecture of proposed handwritten character recognition system

to fulfil this requirement, handwritten data are gathered from people of different age groups and different education levels. At the time of database collection, no constraints are imposed such as style of writing, type of pen, colour of ink, thickness of pen tip etc. Handwritten database forms are digitized through flat-bed scanner at 300 dpi resolution in color format.

### 1.5.2 Segmentation

Handwritten characters are segmented from these digitized forms and benchmark database is designed. At the end of segmentation stage, a set of symbols, technically referred to as glyphs is obtained that needs to be recognized. Algorithms based on projection profiles and connected component labelling are developed in order to segment handwritten characters from filled forms (Jindal, Sharma, and Lehal).

### 1.5.3 Preprocessing

All the isolated characters of database are passed through preprocessing algorithm in order to enhance the quality of images and make them suitable for feature extraction algorithms. Preprocessing steps include operations such as noise removal, size normalization, binarization and thinning. Some of the feature extraction algorithms can be applied directly on grey scale images while for some, images need to be converted in the binary form.

### 1.5.4 Feature Extraction

Effective feature extraction methods are required in order to achieve high recognition rate. Character recognition approaches can be classified as template based or feature based approach (Pal and Chaudhuri). In template based approach, the degree of correlation between test image and ideal template image is calculated which works as the decision factor. Feature based approaches are more suitable for handwritten character recognition as compared to template based approaches due to presence

of large variation in human writing style (Trier, Jain, and Taxt). In the feature based approach, features can be extracted either from the spatial domain or from the transform domain. Features can be extracted directly from the gray scale images or from the binary images based on the type of feature extraction algorithm. Chain code based, zone based and projection profile based feature extraction algorithms are utilized. An approach based on fusion features is also developed. Novel feature extraction techniques based on structural decomposition, zone pattern matching and normalized cross correlation are also proposed for feature vector generation.

### 1.5.5 Classification

The final step of handwritten character recognition system is classification where unknown class labels of the test data are decided. The classifiers such as Artificial Neural Network (ANN), Support Vector Machine (SVM) and Naive Bayes (NB) classifier are used for this purpose. Naive Bayes (NB) classifier is one of the most fundamental and simple classifier. Support Vector Machines (SVMs) are one of the most robust and powerful classifier for linear and non-linear classification problems. Two different kernels such as Linear kernel and Polynomial kernel are used to map the input space to a higher dimensional feature space to establish a linear decision boundary in the transformed space.

## 1.6 Motivation

According to Human Development survey, very small part of the Indian population can read and write English. This clearly necessitates the need of research in the OCR technology for Indian scripts. The research in this domain has the potential to influence human-machine interface system in a beneficial way, which is really important for developing country like India.

It is always expected that to design the algorithms which make the computer to perform functions like reading, writing, understanding etc, but in spite of intensive research in the area of HCR still the reading capability of computer is far behind that of human being and majority of character recognition system fails to read handwritten characters and words from Indian scripts. Demand of the HCR systems is increased since large amount of handwritten data and information is required to en-

ter to the computer for further processing. Automatic recognition of handwritten characters is expected more nowadays. In the context of e-governance, it is more significant. Furthermore, government policies are supporting regional languages and it is instructed that official transaction should be done in regional languages. Therefore, the automatic interpretation of handwritten Gujarati characters would have widespread benefits.

Many promising research results were reported in the area of handwritten character recognition for Indian languages like Devanagari and Bangla, but only a few works could be traced for Gujarati. Despite the huge efforts dedicated to handwriting recognition, it is still hard to find out the best recognition technique available today for Gujarati handwritten character recognition. Gujarati is one of the official language of India which belongs to the group of Indo-Aryan languages and is written in Gujarati script. The language has literary tradition going back to ten centuries (Topiwala). Another relevant fact is that the oldest running published newspaper in India is a Gujarati daily, "Mumbai Samachar", published since 1822. However, it is a surprising fact that, very little of this wealth of Gujarati literature is available in an electronic form which can allow searching and indexing.

However, complexity of the shapes of Gujarati characters poses several challenges in building effective handwritten character recognition system. It may also be noted that Gujarati handwritten character recognition was almost an untouched area at the time we started to work in this area. These facts motivated us to explore the applications of various feature extraction and recognition techniques to attack this problem of handwritten character recognition of Gujarati script. It is important to note at this point that as a part of this research work we have applied several feature extraction and classification techniques to solve the problem of handwritten Gujarati character recognition.

## 1.7 Gujarati Script from OCR Perspective

In India, there are twenty two scheduled languages, namely Hindi, Sanskrit, Bangla, Gujarati, Kannada, Malayalam, Marathi, Oriya, Punjabi, Assamese, Tamil, Telugu Konkani, Kashmiri, Nepali, Bodo, Dogri, Maithili, Manipuri, Santhali, Sindhi, and Urdu (Jomy, Pramod, and Kannan). Different scripts are used for writing these

languages. Most Indian scripts are originated from ancient Brahmi script through various transformations. Two or more of these languages can be written in one script. For example, Devanagari is used to write Hindi, Konkani, Kashmiri, Marathi, Nepali, Sanskrit, Bodo, Dogri, Maithili and Sindhi. Bangla script is used for writing in Bangali, Assamese and Manipuri languages while Gurumukhi script is used for writing in Punjabi language. Devanagari, Bangla, Gujarati, Kannada, Gurumukhi, Oriya, Malayalam, Telugu and Tamil, all these nine scripts are considered as basic beside the Urdu script.

India is the birthplace of the Devanagari script which is the mother of nine different Indian dialects of which Gujarati is one of them. Gujarati belongs to Devanagari family of languages and is spoken by over 65 million people in Gujarat - a western state of India. Apart from the native speakers of the state of Gujarat in India, Gujarati speaking diaspora is spread across all parts of India and in many parts of the world. Gujarati literature is not only of interest to the Gujaratis but also to the researchers abroad (Dwyer Gandhi, *Hind swaraj, or, Indian home rule* Gandhi, *Satya-na Prayogo - Atmakatha (My Experiments with Truth - Autobiography)*).

Gujarati script is a part of the Brahmic family. Figure 1.2 shows the evolution of Indian scripts from Brahmi script (Ghosh, Dube, and Shivaprasad).

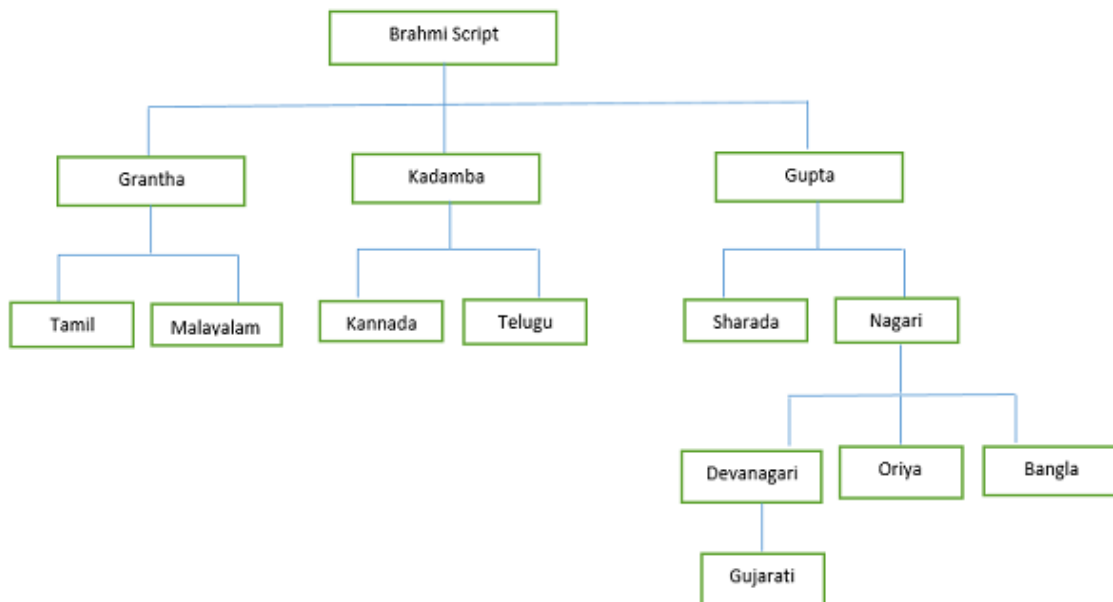


Figure 1.2: The Brahmic family of scripts used in India

Gujarati script is used to write the Gujarati language. Gujarati script is similar

to Devanagari script as most of the words are derived from Sanskrit. Gujarati script has no upper and lower case characters as in English; however the alphabets itself contain more number of symbols than that in English. While line segments (strokes) are the predominant features for English, most of the characters in Gujarati script are formed by curves, holes, and also strokes. Mode of writing in Gujarati script is from left to right and concept of cursive writing is also not present. Most of the characters in Gujarati script are isolated in nature. Gujarati script consist of twelve vowels and thirty six consonants. Aside from basic characters, the additional glyphs which is known as vowel modifiers are used in order to mark the connection of vowels with basic consonants. These Vowel-Consonant conjunctions are present in Gujarati and in many other Indian scripts. It is indicated by connecting a symbol which is unique for every vowel to the consonant and is known as a Matra or dependent vowel modifier. This can be present before, after, above or beneath to the core consonant (Dholakia, Negi, and Mohan, “Progress in Gujarati document processing and character recognition”). Gujarati additionally utilizes conjuncts like the most Indic scripts. A character is called conjunct if that is a half consonant alongside of other consonant. Conjuncts may also be present in half forms also. Figure 1.3 shows the Gujarati consonants and vowels.

One can notice from Figure 1.3 that there are numerous characters which looks similar to one another (Chaudhary et al.). This makes the recognition process difficult. Thus recognition of Gujarati characters includes more complexity than other Latin characters because of complexity in shapes of characters. The structure of Gujarati characters are identical to phonetically resembles characters of Devanagari script. There are some similarities as well as some differences between the Gujarati script and the Devanagari script. Key similarities are (a) there is no differentiation of lower case and upper case. (b) like Devanagari script, the Gujarati script can also be separated into logical zones namely upper, middle and lower as shown in Figure 1.4. This script has two-dimensional compositions of symbols: core characters in the middle strip, optional modifiers above and/or below core characters. Figure 1.4 shows a Gujarati word partitioned into three character zones: A core zone that contains most consonant, half consonant, vowel and conjunct forms (core components), an upper zone containing ascenders or upper modifiers and a lower zone containing



Gujarati Consonants					
ક	ખ	ગ	ઘ	ઙ	ચ
છ	જ	ઝ	ઞ	ટ	ઠ
ડ	ઢ	ણ	ત	થ	દ
ધ	ન	પ	ફ	બ	ભ
મ	ય	ર	લ	વ	શ
ષ	સ	હ	ળ	ક્ષ	ઞ

Gujarati Vowels					
અ	આ	ઇ	ઈ	ઉ	ઊ
એ	ઐ	ઓ	ઔ	અં	અઃ

Figure 1.3: Gujarati consonants and vowels

descenders or lower modifiers. Major difference is that Gujarati does not have a shirolekha or a header line unlike the Devanagari script (Casey and Nagy, *Advances in Pattern Recognition*).

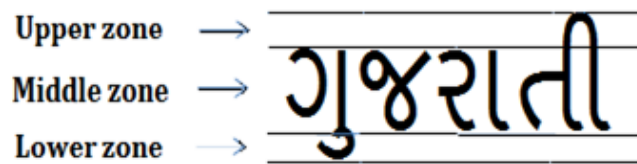


Figure 1.4: Zone separation for Gujarati characters

Gujarati numerals have various shapes and many numerals have close resemblance that creates confusion and have possibilities of incorrect recognition. Ideal shapes of the Gujarati numerals and corresponding English numerals are depicted in Figure 1.5. It is evident from Figure 1.5 that only two Gujarati digits one (1) and five (5) are having straight lines, making recognition task difficult.

૦	૧	૨	૩	૪	૫	૬	૭	૮	૯
0	1	2	3	4	5	6	7	8	9

Figure 1.5: Gujarati numerals and corresponding English numerals

## 1.8 Objectives

Following research objectives are addressed in this thesis:

- To study and analyse the efficiency of various recognition techniques implemented for various Indian scripts.
- To study and analyse the major obstacles and challenges in designing the handwritten character recognition system for Gujarati script.
- To develop a benchmark dataset for handwritten Gujarati characters and numerals which contains maximum possible variability.
- To identify proper segmentation algorithm for extraction of characters from handwritten forms and preprocessing algorithms for processing of isolated character images to make them suitable for feature extraction algorithms.
- To develop, test and compare various features extraction methods on handwritten Gujarati characters in order to determine the best suitable features to accommodate the complexity and variability of handwritten Gujarati characters.
- To investigate possible fusion of features and its impact on recognition accuracy.
- To decide the best combination of feature extraction technique and classifier for handwritten Gujarati character and numeral recognition problem.
- To represent the recognition rate obtained with various feature extraction and recognition techniques in a comparative manner.

## 1.9 Thesis Contributions

Contribution of the thesis in the area of handwritten Gujarati character recognition is summarized through following points:

- Benchmark datasets for handwritten Gujarati numerals and characters were developed.
- Effectiveness of some existing feature extraction methods was investigated and fusion of these features was also proposed for learning prediction models. Few novel feature descriptors for representing handwritten Gujarati characters are also proposed.
- The performance of different feature extraction algorithms, classifiers for handwritten Gujarati character and numeral recognition was compared through a systematic, well-defined as well as exhaustive experimental process.

## 1.10 Organization of the Thesis

The rest of the thesis is organized as follows: chapter 2 presents the literature review. Methodology adapted for handwritten Gujarati character and numeral dataset collection and generation is discussed in chapter 3. Chapter 4 describes several feature extraction techniques implemented in proposed work. A detailed study of these techniques with necessary analysis of their representation capabilities are discussed. Chapter 5 discusses the various prediction models utilized for classification purpose. Experimental results achieved on handwritten Gujarati character and numeral datasets with different feature sets and prediction models are discussed and presented in chapter 6. Finally, chapter 7 concludes the proposed work and also provides discussions on possible extensions for future work.



# Chapter 2

## Literature Survey

Plentiful research on OCR techniques for scripts such as English (Camastra Srihari), Chinese (Dong, Krzyzak, and Suen Liu et al., “CASIA online and offline Chinese handwriting databases”), Japanese (Yamada, Yamamoto, and Saito) and Arabic (Amin Lorigo and Govindaraju Sanossian) has already been carried out. OCR research activities related to Indian scripts are limited. Commercial OCR systems are available for English, Chinese, Japanese and so forth scripts which have the capacity to produce entirely precise content for a wide mixed documents. However OCR system for Indian scripts are not yet commercially available.

Adequate research in the area of OCR techniques for Indian languages such as Hindi, Tamil, Bangla, Kannada, Gurumukhi, Malayalam and Marathi has already been carried out. Research efforts on Gujarati character recognition are few and yet to gain momentum (Pal, Jayadevan, and Sharma). Literature review on Indian OCR indicates that in comparison with Bangla, Hindi, Kannada, Tamil and Telugu scripts, the OCR activities related to Gujarati script is very less (Pal and Chaudhuri). This is really strange and surprising as a lot of research efforts for many other Indian scripts are evident in the literature. This clearly necessitates the need to attend the task of handwritten Gujarati character recognition.

This chapter provides a brief account on research efforts in the domain of printed and handwritten Gujarati character recognition. The Chapter begins with the review of literature in the domain of printed Gujarati character recognition followed by the analysis of the literature on handwritten Gujarati character recognition. Comparison of various methods in terms of feature extraction techniques, classifiers, datasets

and accuracies is also described. The Gujarati script is derived from the Devanagari script, hence, analysis of the research efforts for Devanagari script may provide useful guidelines for the task of Gujarati character recognition. Therefore, survey of literature involving Devanagari character recognition is also presented in this chapter. Literature survey on character recognition for Latin script is provided in Section 2.3.

## 2.1 Literature Survey on Gujarati Character Recognition

### 2.1.1 Printed Gujarati Character Recognition

Sameer Antani and Lalitha Agnihotri (Antani and Agnihotri) originated research in Gujarati OCR. They used regular moments, Hu invariant moments and distribution in binary feature space as feature extraction methods for printed Gujarati characters. They used K-Nearest Neighbor (KNN) classifier and Hamming Distance classifier and achieved recognition rates of 67% and 48% respectively with these classifiers. A zone separation algorithm was proposed by Dholakia et al. (Dholakia, Negi, and Mohan, “Zone identification in the printed Gujarati text”) for printed Gujarati text. The algorithm focused on computing the slopes of all the imaginary lines that joined the top left corners and bottom right corners of all the possible pairs of connected components. The algorithm was applied on 20 lines extracted from 3 different document images and in 19 cases, the zone boundary was detected correctly. Prof S K Shah and A Sharma (Shah and Sharma) implemented a template matching system for printed Gujarati character recognition. Fringe distance is used for the comparison of input image with the template. City-block distance method is used for fringe distance measurement. Experiment was performed on small dataset of 1375 images. They achieved the recognition accuracy of 72.30%.

Jignesh Dholakia et al. (Dholakia, Yajnik, and Negi) used Daubechies D4 wavelet coefficients as features. For the dataset of 4173 characters they achieved 97.59% and 96.71% accuracies with Generalized Regression Neural Network (GRNN) and nearest neighbor classifier respectively. Dholakia and Negi (Dholakia, Negi, and Mohan, “Progress in Gujarati document processing and character recognition”) worked in the area of printed Gujarati document processing and character recognition. Various fea-

ture extraction algorithms with KNN and Artificial Neural Network (ANN) classifiers is implemented by them. They concluded that the combination of wavelets feature and GRNN gives better recognition results. Mr. Mukesh M. Goswami et al. (Goswami, Prajapati, and Dabhi) used the combination of Self Organizing Map (SOM) and KNN to classify printed Gujarati characters. They used dataset of total 693 character images and achieved 82.36% accuracy. Mandar Chaudhary et al. (Chaudhary et al.) used extended version of supervised locality preserving projection (ESLPP) coefficients as features with neural network classifier for the classification of similar looking Gujarati characters. They used dataset of 80 to 100 images for each character and achieved accuracy of 96%. Mukesh Goswami and Suman Mitra (Goswami and Mitra, “Structural feature based classification of printed Gujarati characters”) used structural feature extraction method for Gujarati character recognition. They identified total 30 strokes which formulates almost all printed Gujarati character set. These strokes are detected by using various  $3 \times 3$  pattern mask. Simple rule based algorithm is used for classification purpose. This method was tested on dataset of 4000 printed characters. They achieved accuracy of 95%.

E. Hassan et al. (Hassan, Chaudhury, and Gopal) proposed a binary multiple kernels learning based classification architecture for printed Gujarati and Bangla character recognition. They used a Multiple Kernel Learning based Support Vector Machine (MKL-SVM) classifier with multiple features, namely fringe distance map, shape descriptor, and HOG. Classification accuracies between 95-99% is achieved by them with the application of these different features. Mukesh M. Goswami and Suman K. Mitra (Goswami and Mitra, “Classification of Printed Gujarati Characters Using Low-Level Stroke Features”) utilized low-level stroke features for printed Gujarati character recognition. The database consists of approximately 16,782 samples were used and experiments were performed using KNN classifier. They achieved the recognition accuracy of 98.13%. Table 2.1 represents significant research efforts in the area of printed Gujarati character recognition.

### 2.1.2 Handwritten Gujarati Character Recognition

First attempt in the area of handwritten Gujarati numeral recognition is made by Apurva Desai in 2010. Apurva Desai (Desai, “Gujarati handwritten numeral optical

Table 2.1: Research efforts in the area of printed Gujarati character recognition

Author	Numerals/alphabets	Features	Classifier	Accuracy (%)
Sameer Antani et.al.(1999)	Alphabets	Regular and Hu invariant moments	KNN and Hamming Distance Classifier	67
Prof S K Shah et.al.(2006)	Alphabets	Fringe distance measurement	Template Matching based classifier	72.3
Jignesh Dholakia et.al.(2007)	Alphabets	Daubechies D4 wavelet transform	GRNN and KNN	97.59
Mukesh M. Goswami et.al.(2011)	Alphabets	Pixel intensity	Combination of SOM and KNN classifier	82.36
Mandar Chaudhary et al.(2012)	Alphabets	Supervised locality preserving projection coefficients	MLP with BPNN classifier	96
Mukesh Goswami et.al.(2013)	Alphabets	Structural features	Rule based classifier	95
E. Hassan et al. (2014)	Alphabets	Fringe distance, shape descriptor and HOG	Multiple Kernel Learning based SVM	95-99
Mukesh Goswami et.al.(2016)	Alphabets	Low-level stroke features	KNN	98.13

character reorganization through neural network”) used projection profile based feature vector with Multi-layered feed forward neural network classifier. Accuracy of 81.66% is achieved for handwritten Gujarati numerals. Mamta Maloo and K.V. Kale (Maloo and Kale) attempted handwritten Gujarati numeral recognition by using an affine invariant moments based features along with Support Vector Machine classifier. They used 80 sample images of each Gujarati numeral. Accuracy reported was 91%. Baheti M.J. and K.V. Kale (Baheti, Kale, and Jadhav) used affine invariant moments based features. They obtained 90% and 84% accuracies with KNN classifier and PCA based classifier respectively.

Chhaya Patel and Apurva Desai (Patel and Desai) implemented recognition method for handwritten Gujarati characters and numerals by using structural and statistical features like moment based features, centroid distance based features with KNN classifier. They achieved accuracy of 63.1%. Lipi Shah et al. (Shah et al.) used radial histogram based feature extraction technique and Euclidean Distance classifier for classification purpose. They used dataset of 11,720 characters and achieved accuracy of 26.86%. Hetal R. Thaker, C. K. Kumbharana (Thaker and Kumbharana) used structural features like connected and disconnected components, vertical lines, horizontal lines, diagonal lines, end points, cross points, type of curves, number of closed loops as feature extraction technique for five isolated handwritten Gujarati



characters. Decision tree classifier was used for classification purpose. They have used dataset of 750 characters and achieved accuracy of 88.78%.

Table 2.2: Research efforts in the area of handwritten Gujarati character recognition

Author	Numerals/alphabets	Features	Classifier	Accuracy (%)
Desai et.al.(2010)	Numerals	Projection profiles based feature	Neural network classifier	81.66
Mamta Maloo et al. (2011)	Numerals	Affine invariant moments	SVM	91
Baheti M.J. et al.(2011)	Numerals	Affine invariant moments	KNN	90
Chhaya Patel et.al.(2013)	Alphabets	Structural and statistical features	Binary tree and KNN	63.10
Lipi Shah et.al.(2014)	Alphabets	Radial histogram based features	Euclidean Distance Classifier	26.86
Hetal R. Thaker et.al.(2014)	Alphabets	Structural features	Decision tree classifier	88.78
Ravi Nagar et.al.(2015)	Numerals	Stroke orientation based features	SVM	98.93
Mukesh Goswami et.al.(2015)	Numerals	Low-level stroke features	KNN and SVM	98
Desai et.al.(2015)	Alphabets	Hybrid features	KNN and SVM	86.66

Ravi Nagar and Suman K. Mitra (Nagar and Mitra) proposed feature extraction based on stroke orientation estimation technique for handwritten Gujarati numerals. The efficiency of the feature set is tested using a linear Support Vector Machine classifier and obtained the recognition accuracy of 98.93%. Mukesh M. Goswami and Suman K. Mitra (Goswami and Mitra, “Offline handwritten Gujarati numeral recognition using low-level strokes”) proposed a technique for the extraction of various low-level stroke features, like endpoints, junction points, line segments, and curve segments, and the block-wise histogram of low-level stroke features is used for the recognition of handwritten numerals from Gujarati and Devanagari scripts. Experiments were performed using KNN and Support Vector Machine (SVM) classifiers with radial basis function (RBF) kernel. The average test accuracies obtained on Gujarati and Devanagari database were 98.46% and 98.65%, respectively. Apurva Desai (Desai, “Support vector machine for identification of handwritten Gujarati alphabets using hybrid feature space”) applied image subdivision based approach for feature extraction. He utilized combination of aspect ratio, extent and zone density

based features. Performance accuracy of 86.66% was achieved with SVM classifier. Key research efforts in the area of handwritten Gujarati character recognition are summarized in Table 2.2.

## 2.2 Literature Survey on Devanagari Character Recognition

### 2.2.1 Printed Devanagari Character Recognition

Research work in the area of printed Devanagari character recognition was initiated in 1970s. First attempt was made by R. M. K. Sinha and H. Mahabala (Sinha and Mahabala). They used structural features and syntactic pattern analysis for classification of printed Devanagari characters. They achieved recognition accuracy of 90%. Krishnamachari Jayanthi et al. (Jayanthi et al.) implemented structural feature based approach for recognizing Devanagari characters used in a printed Buddhist text: Saddharmapundarika. They used features like main horizontal line, presence or absence of vertical lines, number of free ends and aspect ratio. Dataset of 4863 characters was used by them and achieved 95% accuracy with binary tree classifier. B. B. Chaudhuri and U. Pal (Chaudhuri and Pal) used two stage approach for Devanagari and Bangla character recognition. In first stage primary grouping of printed Devanagari and Bangla characters was done using stroke based features and a tree classifier. Features used were like vertical line, width of boundary box etc. In second stage run length template matching approach was used. They used dataset of 10,000 words and achieved 97.18% accuracy for Devanagari script. V. Bansal and R.M.K. Sinha (Bansal and Sinha, “Integrating knowledge sources in Devanagari text recognition system”) used statistical and structural features with statistical knowledge source for classification of printed Devanagari characters. They achieved 87% accuracy. Veena Bansal and R. M. K. Sinha (“A complete OCR for printed Hindi text in Devanagari script”) implemented a complete recognition system for printed Devanagari script. Structural feature like core strip, vertical bar, horizontal zero crossings, position and number of vertex points, translation and scale invariant moments up to an order of two are used. They used distance based classifier for classification purpose. They achieved 93% accuracy at character level. Huanfeng Ma and David

Doermann (Ma and Doermann) used combination of Generalized Hausdorff Image Comparison (GHIC) and SVM classifier for printed Devanagari characters. They used dataset of 2727 characters and achieved accuracy around 88%. The bilingual recognition system through Principal Component Analysis (PCA) and SVM classifier has been designed by C. V. Jawahar et al. (Jawahar, Kumar, and Kiran) for printed Hindi and Telugu characters. They considered the pixel intensity of whole image as feature vector and used the PCA for dimensionality reduction. KNN and SVM are used for classification purpose. Linear and quadratic kernels are used by them for SVM based experimentations. They used dataset of 2,00,000 characters and achieved overall accuracy of 96.7%. Venu govindaraju et al. (Govindaraju et al.) used gradient features for printed Devanagari character representation and neural network for classification purpose. They used dataset of 4506 sample images and obtained the accuracy of 96.04%. A. Dhurandhar et al. (Dhurandhar, Shankarnarayanan, and Jawale) used contour based features. They used dataset of 546 printed Devanagari characters and achieved 93.03% accuracy. Suryaprakash Kompalli et al. (Kompalli et al.) used gradient features and neural network classifier. They used dataset of 32,413 printed Devanagari characters and achieved the recognition rate of 61.80%. S. Kompalli et al. (Kompalli, Setlur, and Govindaraju) used GSC (gradient, structural, concavity) based feature extraction method and KNN as classifier for printed Devanagari characters. They used dataset of 9297 characters and achieved 95% accuracy. K. D. Dhingra et al. (Dhingra, Sanyal, and Sharma) used Gabor features and Minimum Classification Error (MCE) approach based classifier for printed Devanagari characters. They used dataset of 30,000 characters and achieved 98.5% accuracy. P. Natarajan et al. (Natarajan, MacRostie, and Decerbo) used derivatives features and Hidden Markov Model (HMM) based classifier. They used dataset of 21,982 printed Devanagari characters and achieved 91.30% accuracy. Sukalpa Chanda et al. (Chanda et al.) used 64-dimensional chain code in first stage and 400-dimensional gradient features in second stage for printed English, Devanagari and Bengali characters. They used SVM classifier and with dataset of 11,123 characters achieved accuracy of 98.51%.

### 2.2.2 Handwritten Devanagari Character Recognition

In past few years a significant contribution is done in research related to the recognition of handwritten Devanagari characters. Many researchers have proposed different approaches for handwritten Devanagari character recognition. N. Sharma et al. (Sharma et al., “Recognition of off-line handwritten Devnagari characters using quadratic classifier”) used histogram of directional chain code as features. They obtained 64 dimensional feature vectors and these are fed to the Quadratic classifier. They used dataset of 11,270 characters and 22,556 numerals, and obtained 98.86% and 80.46% accuracies respectively. S. Arora et al. (Arora et al., “A two stage classification approach for handwritten Devnagari characters”) used structural features and feed forward neural network classifier. For Devanagari handwritten character dataset of 50,000 characters, 89.12% accuracy is achieved by them.

M. Hanmandlu et al. (Hanmandlu, Murthy, and Madasu) used vector distance based feature extraction method and fuzzy sets based classifier. They used dataset of 4750 handwritten Devanagari characters and achieved 90.65% accuracy. U. Pal et al. (Pal et al., “Off-line handwritten character recognition of Devnagari script”) used gradient and Gaussian filters based feature extraction method and Quadratic classifier. They used dataset of 36,172 characters and achieved 94.24% accuracy. U. Pal et al. (“Accuracy improvement of Devnagari character recognition combining SVM and MQDF”) used gradient features with SVM and Modified Quadratic Discriminant Function (MQDF) based classifier for Devanagari handwritten characters. Experiment was performed on dataset of 36,172 characters and achieved 95.13% accuracy. A scheme based on regular expressions and minimum edit distance has been implemented by P. S. Deshpande et al. (Deshpande, Malik, and Arora). They used the dataset of 5000 handwritten Devanagari characters and obtained 82% accuracy. They suggested that regular expressions technique is the most sophisticated method to perform operations like string searching, validation, manipulation, and formatting for all applications which deals with the text data.

U. Pal et al. (Pal, Wakabayashi, and Kimura) used gradient features and Mirror Image Learning (MIL) based classifier. They used dataset of 36,172 handwritten Devanagari characters and achieved 95.19% accuracy. S. Arora et al. (Arora et al.,

“Study of different features on handwritten Devnagari character”) used chain code histogram, four side views and shadow based features with MLP classifier. They used dataset of 1500 handwritten Devanagari characters and achieved 89.58% accuracy. V. Mane and L. Ragha (Mane and Ragha) used Eigen deformation based feature extraction method and elastic matching for classification purpose. They achieved 94.91% accuracy on dataset of 3600 characters. Satish Kumar (Kumar) implemented five feature extraction methods which are chain code, Kirsch directional edges, gradient, distance transform, directional distance distribution. With these features they obtained 80.60%, 92.40%, 88.10%, 92%, 93.50%, and 94.10% accuracies respectively with SVM classifier and 82.20%, 88.70%, 83.30%, 89.70%, 89.60%, and 91.90% accuracies respectively with Multi-layer Perceptron (MLP) classifier. They used the dataset of 25,000 handwritten Devanagari characters. A method based on two stage approach is proposed by Sandhya Arora et al. (Arora et al., “Recognition of non-compound handwritten Devnagari characters using a combination of mlp and minimum edit distance”). They used chain code histogram and shadow features. Combined approach based on Minimum Edit Distance and MLP is used for classification purpose. They used the dataset of 7154 handwritten Devanagari characters and obtained 90.74% accuracy. Karbhari V. Kale et al. (Kale et al.) used Zernike moment based feature descriptors with KNN and SVM classifiers. They used dataset of 27,000 characters. They achieved accuracies of 95.82% and 98.37% using KNN and SVM classifiers respectively. Deepti Khanduja et.al.(Khanduja, Nain, and Panwar) used the combination of structural and statistical features. Structural features considered were like number of endpoints, loops, and intersection points etc. A quadratic curve fitting model is applied on each zone of the statistically partitioned image and feature vector is designed from the coefficients of the optimally fitted curve. They achieved the recognition accuracy of 93.4%.

## **2.3 Literature Survey on English character recognition**

Latin script, is utilized for writing languages such as English, French, Italian, German and some other European languages (Ghosh, Dube, and Shivaprasad). A comprehensive survey on handwritten character recognition for Latin script are provided in

(Plamondon and Srihari), (Arica and Yarman-Vural) and (Bortolozzi et al.).

Wunsch and Laine (Wunsch and Laine) utilized features based on shape descriptor derived from the wavelet transform of a pattern's contour to represent handwritten characters with ANN classifier. Dataset of size 6000 samples of handwritten characters was utilized for experimentation purpose. They achieved recognition accuracy of 99.26%. Lee et al. (Lee et al.) utilized multiresolution features with wavelet transform along with a multilayer cluster neural network for classification of handwritten numerals. They utilized handwritten numeral datasets of Concordia University of Canada, Electro-Technical Laboratory of Japan, and Electronics and Telecommunications Research Institute of Korea for experimentation purpose. They obtained error rates of 3.20%, 0.83%, and 0.75%, respectively for these datasets with their algorithm.

Kavallieratou et al. (Kavallieratou, Fakotakis, and Kokkinakis) utilized horizontal and vertical histograms with radial histogram lexical component based on dynamic acyclic FSAs (Finite-State-Automata) to represent handwritten characters. They achieved recognition accuracy varying from 72.8% to 98.8% on two different databases with this approach. Chen et al. (Chen, Bui, and Krzyzak) implemented shell coefficients as features along with feed-forward neural network to recognize the handwritten numerals. They achieved recognition accuracy of 92.20%.

Bellili et al. (Bellili, Gilloux, and Gallinari) implemented hybrid MLP-SVM method for handwritten digits recognition. They achieved a recognition accuracy of 98.01% for mail zip code digits recognition. Several features including chain code, gradient feature, profile structure feature and peripheral direction contributivity were utilized to represent handwritten digits by Liu et al. (Liu et al., "Handwritten digit recognition: benchmarking of state-of-the-art techniques"). They utilized k-nearest neighbor classifier, neural network classifier and support vector classifier for recognition purpose. They achieved accuracy values of 99.58% and 99.42% by SVCs and non-SV classifiers respectively.

Zhang et al. (Zhang, Bui, and Suen) utilized cascade ensemble classifier system for the recognition of handwritten digits. They utilized seven sets of discriminative features and three sets of random hybrid features with cascade recognition system. With this proposed system, they achieved a reliability of 99.96% with a 99.19% recognition rate with rejection strategy in the last layer of the cascade system. Vamvakas

et al. (Vamvakas, Gatos, and Perantonis) implemented feature extraction technique based on recursive subdivisions of the character image. They implemented a two-stage classification scheme based on the level of granularity of the feature extraction method. They achieved recognition accuracies of 94.73% for the CEDAR Character Database and 99.03% for the MNIST Database.





## Chapter 3

# Handwritten Gujarati Dataset Development

Development of a good dataset that represents wide variations of handwriting styles is one of the most challenging aspect of handwritten character recognition. One of the biggest obstruction for the research in the area of handwritten Gujarati character recognition is the unavailability of standard dataset. The shapes of the components of an alphabetic character set reflect the philosophy in which character set was born. All the characters share general similarities but are different from each other in their shapes. There may be several variations possible for a single character while writing.

Research in the field of Gujarati character recognition is still in budding stage. From the literature review it can be noticed that compare to many other scripts research activities for Gujarati character recognition is very less. One significant reason for the lack of research activities in the area of Gujarati handwritten character recognition is the unavailability of benchmark dataset. To design a handwritten character recognition system, one of the important step is collection of dataset. However, to the best of our knowledge, no such benchmark dataset is available for handwritten Gujarati characters. One of the significant contribution of our work is towards the generation of large and representative dataset of handwritten Gujarati numerals and characters.

This chapter focuses on the steps adapted for generation of an efficient and representative dataset of handwritten Gujarati characters. Handwritten dataset collection and generation methods for characters are described in Section 3.2 and for numerals

are described in Section 3.3. Description of the generated handwritten dataset is provided in same sections. Extraction of characters from scanned handwritten data collection forms is needed in order to create the dataset of isolated characters. An algorithm based on projection profiles and connected component labelling is designed for this purpose. The details of this method are described in Section 3.4.

Preprocessing methods implemented in this work such as noise removal, size normalization, binarization and thinning are described in Section 3.5. These algorithms are needed to enhance the quality of segmented character images and make them suitable for feature extraction algorithms. Conclusion for this chapter is provided in Section 3.6.

### 3.1 Benchmark Databases for HCR Research

Various research groups created several benchmark databases in order to enhance research in the area of handwritten character recognition. Some of the datasets such as MNIST (LeCun, Cortes, and Burges) and CEDAR (Hull), CENPARMI (Suen et al.) etc. have been extensively used in research on recognizing Latin numbers and characters. MNIST database is consist of 70,000 isolated and labelled handwritten digits. Centre of Excellence for Document Analysis and Recognition, SUNY, Buffalo created CEDAR database which contains handwritten words and ZIP codes of handwritten English characters and digits. Centre for Pattern Recognition and Machine Intelligence, Concordia University released the CENPARMI digit database which is consist of 6000 digit images.

Few of the databases like ETL9B database (Saito, Yamada, and Yamamoto), HCL2000 database (Zhang et al.) and CASIA database (Liu et al., “CASIA online and offline Chinese handwriting databases”) are very popular among researchers for work in the area of Chinese character recognition. Electro Technical Laboratory, Japan generated ETL9B database. It contains 200 samples for each character from the Chinese and Japanese character sets. Beijing University of Posts and Telecommunications collected HCL2000 database and this database is consist of 3755 Chinese characters. National Laboratory of Pattern Recognition, Institute of Automation of Chinese Academy of Sciences (CASIA) generated CASIA online and offline Chinese handwriting database.

Some of the databases are available for Indian scripts also. Indian Statistical Institute, Kolkata developed one such database which contains numerals and characters for Bangla and Devanagari scripts (Bhattacharya and Chaudhuri). This database also contains isolated numerals of Oriya script. A database named hpl-tamil-iso-char was developed by HP labs which contains handwritten samples of Tamil characters. HP Tablet PCs were utilized for this database collection and offline version of this database is available for researchers (Bhaskarabhatla and Madhvanath).

A benchmark database is essential in order to conduct successful research in the area of handwritten character recognition. Such a huge and representative benchmark dataset is not available for handwritten Gujarati characters. Hence, the creation of large and representative dataset of handwritten Gujarati characters is one of the noteworthy contribution of proposed work.

## 3.2 Handwritten Character Dataset Development

The availability of dataset that captures variations encountered in real world is a critical issue in any experimental research. Study of the available literature on handwritten Gujarati character/numeral recognition reveals that the most of these past studies have relied on datasets of small size, which emphasizes the need of a large dataset for the task. This makes creation of a dataset consisting of large number of handwritten Gujarati characters a significant contribution to the research.

Handwriting style is affected by the factors that include age, sex, education, profession, writing surface and writing pen. A representative benchmark dataset must include instances that capture these variations. A special form was designed for dataset collection and it was taken care that samples were collected by capturing maximum possible influencing factors. Data collection form was provided to the writers and requested to write Gujarati characters in the form. The purpose of the task was not revealed to the writers to ensure that they wrote in their natural handwriting style. The special form designed for the dataset collection is depicted in Figure 3.1. Printed version of the characters was provided in the dataset collection form, so that people will not ignore or forget to write any character.

Dataset collection was done with the help of writers belonging to different age groups from children to old age. These writers were not only different in their ages,

ક	ક	ઠ	ઠ	ન	ન	વ	વ	ઈ	ઈ
ખ	ખ	ડ	ડ	પ	પ	શ	શ	ઈ	ઈ
ગ	ગ	ડ	ડ	ફ	ફ	ષ	ષ	ઉ	ઉ
ઘ	ઘ	ઢ	ઢ	બ	બ	સ	સ	ઊ	ઊ
ચ	ચ	ણ	ણ	મ	મ	લ	લ	ઋ	ઋ
છ	છ	ત	ત	મ	મ	ળ	ળ		
જ	જ	થ	થ	ય	ય	ક્ષ	ક્ષ		
ઝ	ઝ	ઁ	ઁ	ર	ર	શ	શ		
ઞ	ઞ	ઃ	ઃ	લ	લ	અ	અ		

Figure 3.1: A sample filled form utilized for character dataset collection

but they also belonged to different educational backgrounds such as school children, college students, senior citizens, house wives and persons from various professions. In order to capture variations in the profession of writers, persons having different professions like farmers, clerks, drivers, shop keepers, managers, doctors and teachers were considered while dataset collection. The other disparities were made sure through varying the color and quality of background surface and the tip of the pointer of writing pen along with color and quality of the ink. Each writer was requested to write with his/her own writing pen or from a set of different types of pens in order to provide the variations in type of pen, color of ink, thickness of pin tip etc.

Equal proportion of male and female writers was maintained while form filling and writers belonging to different districts of Gujarat were considered. Handwritten dataset collection process is executed over a span of more than two years. These efforts ensured maximum possible versatility in the images of the dataset. The collection form also ensured that the dataset images are uniformly distributed over different classes.

All filled handwritten forms were scanned at the resolution of 300 dpi using a flatbed scanner and stored in the form of grayscale images. All the forms were saved in JPEG format. Two different datasets were generated from these filled forms. These datasets are referred as Handwritten Gujarati Character Dataset-1 (HGCD-1) and Handwritten Gujarati Character Dataset-2 (HGCD-2) in the rest of the thesis.

HGCD-1 consist of total 88,000 isolated character images, which are divided into

44 classes as shown in Figure 3.2. In HGCD-1, few isolated symbol together generate the complete character. For example class 3 and class 4 symbol together form the character 'Ga'. Similarly class 28 and class 29 together form the character 'La'. Total 2000 images of each symbol is generated.













































						
Class-1	Class-2	Class-3	Class-4	Class-5	Class-6	Class-7
						
Class-8	Class-9	Class-10	Class-11	Class-12	Class-13	Class-14
						
Class-15	Class-16	Class-17	Class-18	Class-19	Class-20	Class-21
						
Class-22	Class-23	Class-24	Class-25	Class-26	Class-27	Class-28
						
Class-29	Class-30	Class-31	Class-32	Class-33	Class-34	Class-35
						
Class-36	Class-37	Class-38	Class-39	Class-40	Class-41	Class-42
						
Class-43	Class-44					

Figure 3.2: Gujarati characters considered for HGCD-1 and corresponding classes assigned

HGCD-2 consist of total 20,500 handwritten Gujarati characters, which are uniformly divided into 41 classes. Gujarati characters considered for HGCD-2 and corresponding classes assigned are shown in Figure 3.3. There are few differences in symbols considered for HGCD-1 and HGCD-2. For example, in HGCD-1, class 3 and class 4 symbols together form the character 'Ga', while in HGCD-2, class 3 symbol alone form the character 'Ga'. Similarly in HGCD-1, class 28 and class 29 symbols together form the character 'La', while in HGCD-2, class 27 symbol alone form the character 'La'. Hence, in HGCD-1 all the characters/symbols consist of single con-

nected component, while in HGCD-2 there are some symbols which consist of more than one connected component.

Gujarati characters:								
Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8	Class 9
Class 10	Class 11	Class 12	Class 13	Class 14	Class 15	Class 16	Class 17	Class 18
Class 19	Class 20	Class 21	Class 22	Class 23	Class 24	Class 25	Class 26	Class 27
Class 28	Class 29	Class 30	Class 31	Class 32	Class 33	Class 34	Class 35	Class 36
Class 37	Class 38	Class 39	Class 40	Class 41				

Figure 3.3: Gujarati characters considered for HGCD-2 and corresponding classes assigned

### 3.3 Handwritten Numeral Dataset Development

Datasets such as NIST (Wilkinson et al.) , MNIST (LeCun et al., “Gradient-based learning applied to document recognition”), CEDAR (Hull) and CENPARMI (Suen et al.) etc. are very popular and have been used extensively in research on recognizing Latin numbers but no such benchmark dataset is available for handwritten Gujarati numerals. The dataset developed through our efforts consist of total 14,000 images of Gujarati numerals. Special form utilized for the collection of handwritten Gujarati numerals is depicted in Figure 3.4.



Figure 3.4: A sample filled form for numerals

Writers were requested to write the numerals in rectangular boxes which are visible in the form. Two copies were filled up from each writer, however, at different and well separated times. This was done to capture variations and impreciseness in the writing styles of the same person. Handwriting style is affected by the mood of

the writer at different instances of time. Figure 3.5 depicts two copies of the form filled by the same person but at the different and well separated times.

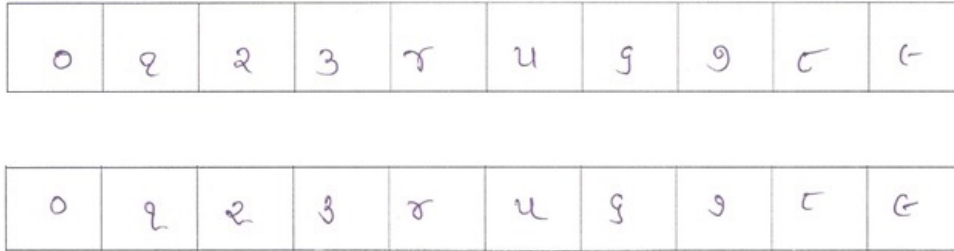


Figure 3.5: Variation in writing style of the same person at different instances of time

Similar to the strategy applied for character dataset collection, the purpose of the task was not revealed to the writers to ensure that they wrote in their natural handwriting style. Numeral dataset collection was also done with the help of writers belonging to different age groups from children to old age. Writers having different educational backgrounds and different professions were considered for dataset collection. Variations in the colour and quality of background surface, variation in tip of the pointer of writing pen was also provided. These efforts ensured maximum possible versatility in the images of the dataset. The collection form also ensured that the dataset images are uniformly distributed over different classes, 10 in this case. The process led to a total of 14,000 numeral images with 1400 images of each class.

All the forms used for numeral dataset collection were also scanned at the resolution of 300 dpi using a flatbed scanner and stored in the form of grayscale images (Desai, “Gujarati handwritten numeral optical character reorganization through neural network”). Segmentation was applied on all the forms in order to get the isolated numeral images. Segmentation algorithm is discussed in Section 3.4. Individual numeral images were finally stored in gray-scale image form. This generated numeral dataset is referred as Handwritten Gujarati Numeral Dataset (HGND) in the rest of the thesis. Figure 3.6 represents the numerals considered and corresponding classes assigned for HGND.

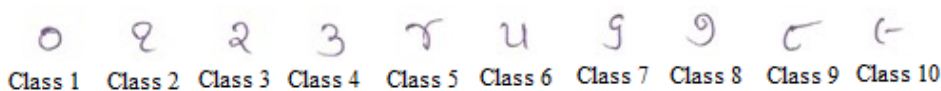


Figure 3.6: Gujarati numerals considered for HGND and corresponding classes assigned

Figure 3.7 was generated by aggregating segmented numerals from different forms of various persons. This figure clearly demonstrates the diversity in the writing style of different volunteering writers.

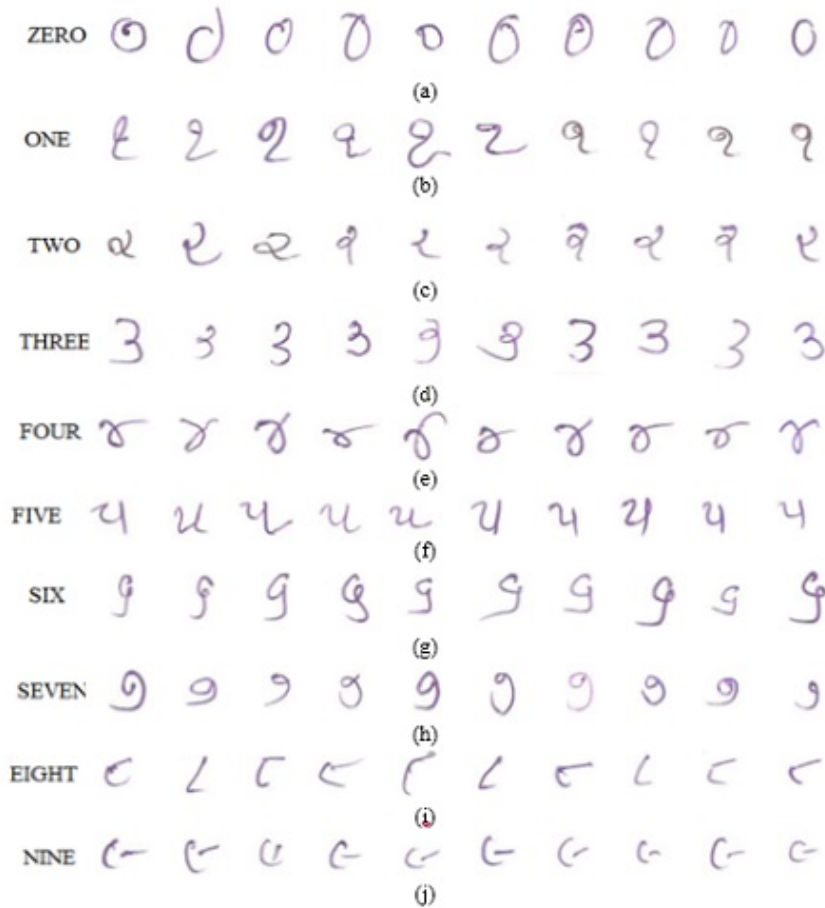


Figure 3.7: (a) to (j) - indicating variations in writing styles of different writers for digits 0 to 9 respectively

Table 3.1 represents all the datasets generated and utilized in this study.

Table 3.1: Handwritten Gujarati datasets

Datasets	Acronym	Number of images	Classes
Handwritten Gujarati Numeral Dataset	HGND	14,000	10
Handwritten Gujarati Character Dataset-1	HGCD-1	88,000	44
Handwritten Gujarati Character Dataset-2	HGCD-2	20,500	41



## 3.4 Segmentation

Segmentation was applied on all the forms in order to get the isolated handwritten images. All the isolated character images were manually verified by a group of volunteers in order to authenticate the correctness of the character segmentation algorithm. Individual character images generated by the segmentation operation were finally stored in gray-scale image form. In order to segment handwritten characters from filled forms, algorithms based on horizontal projection profiles and connected component labelling were developed (Jindal, Sharma, and Lehal). Two separate algorithms were used for generation of isolated character image datasets. Character Extraction Algorithm 1 was used for generation of HGCD-1 while Character Extraction Algorithm 2 was used for generation of HGCD-2 and HGND.

---

### Algorithm 1 Character Extraction Algorithm 1

---

**Input:** Filled data collection form image

**Output:** Segmented character images

- 1: First of all the square grid of the form image is removed. Grid is the biggest connected component in form image. Let  $I_g$  denote the form image having grid and  $I_{wg}$  denote image without grid.
  - 2: Horizontal projection profile is calculated for this image  $I_{wg}$  and all the rows are segmented based on this.
  - 3: Each row is separately processed. The isolated component in each segmented row are labelled using connected component labelling algorithm. Median filtering is applied in order to remove any salt and pepper noise.
  - 4: The minimum bounded rectangles containing the components are detected and cropped one by one from the row image. These segmented character images are placed in proper folders in JPEG format.
- 

Dataset generated through Algorithm 1 is considered as HGCD-1. Handwritten Gujarati characters/symbols considered for this dataset are shown in Figure 3.2.

Dataset generated through Algorithm 2 is considered as HGCD-2. Handwritten numeral dataset (HGND) is also generated through same algorithm. Handwritten Gujarati characters/symbols considered for HGCD-2 are shown in Figure 3.3.

## 3.5 Preprocessing

Preprocessing steps were adapted in order to enhance the quality of the character images. A series of preprocessing steps were applied on all the isolated character and numeral images to make them suitable for feature extraction algorithms. All prepro-

---

**Algorithm 2** Character Extraction Algorithm 2

---

**Input:** Filled data collection form image**Output:** Segmented character images

- 1: First of all the square grid of the form image is removed. Grid is the biggest connected component in form image. Let  $I_g$  denote the form image having grid and  $I_{wg}$  denote image without grid.
  - 2: Horizontal projection profile is calculated for this image  $I_{wg}$  and all the rows are segmented based on this.
  - 3: Each row is separately processed. Vertical projection profile is calculated for each row image and characters are segmented based on this vertical profiles. These segmented character images are placed in separate folders in JPEG format.
- 

cessing operations which were used for improving the quality of character images are described in this section.

For the task of identifying handwritten characters, the most important thing is to bring all the segmented characters in a standard normal form. This is absolutely necessary for handwritten character recognition system as writers may write using different types of pens, papers and they may even follow different writing styles.

All these observations clearly mandate for preprocessing of images before proceeding for the recognition task. Preprocessing steps for segmented character images included median filtering, binarization, resizing and thinning. These steps are summarized in Figure 3.8.

Smoothing of boundaries of characters was carried out using  $3 \times 3$  median filter (Desai, “Gujarati handwritten numeral optical character reorganization through neural network”). It also ensured removal of any salt and pepper noise. Otsu’s method (Otsu) was used to calculate the global threshold in order to binarize character images.

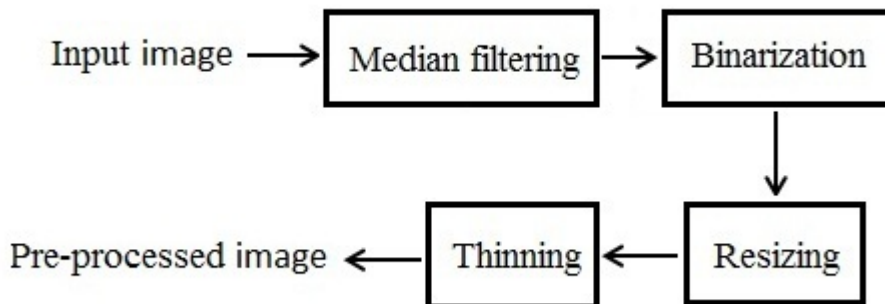


Figure 3.8: Preprocessing steps for segmented character images

Resizing of all the character images to a uniform size was achieved through the bilinear interpolation algorithm (Das et al.). Finally, the thinning operation was applied to obtain one-pixel wide thin image.

### 3.5.1 Noise Removal

There can be different type of noises that may deteriorate quality of the image. Median filtering is a nonlinear method used to remove noise from images. It is widely used as it is very effective at removing noise while preserving edges. The median filter works by moving through the image pixel by pixel, replacing each value with the median value of neighbouring pixels. Hence, filtered image pixel contains the median value in a 3-by-3 neighborhood around the corresponding pixel in the input image. Smoothing of boundaries of characters was carried out using a median filter.

Median filters are quite popular because, for certain types of random noise, these provide excellent noise-reduction capabilities, with considerably less blurring than linear smoothing filters of similar size. Median filters are particularly effective in the presence of impulse noise, also called salt-and-pepper noise because of its appearance as white and black dots superimposed on an image.

### 3.5.2 Binarization

Binarization helps to focus more on the shape of the characters and to avoid unnecessary background details from the character image. Moreover it is required to convert the image into binary form before applying some feature extraction algorithms since some of the feature extraction algorithms work only on binary images. Sezgin and Sankur (Sezgin) surveyed various binarization methods whereas OD Trier, T Taxt (Trier and Taxt) provided evaluation of various binarization algorithms for document images.

Thresholding was used in order to implement the binarization operation. There are two types of thresholding operations, namely local and global thresholding operations (Kasturi, O'gorman, and Govindaraju). In case of local thresholding techniques, threshold is calculated based on the neighbouring pixels, while in case of global techniques threshold is decided from the whole image which provides faster thresholding operation as compare to local thresholding techniques. Otsu's method is based on global thresholding and it is one of the best and fastest thresholding method (Trier

and Text). In this work, Otsu's method was used for binarization.

### 3.5.3 Size Normalization

Normalization operation is required to convert the random sized image into standard sized image. Size of characters varies from person to person and even with the same person from time to time depending on the mood of the writer. Due to available high variability in the character size and shape size normalization is needed before feeding character images to feature extraction algorithm. The image interpolation algorithm with 'bilinear' method was used to resize each isolated character image. The 'bilinear' method produces smoother edges of image compared to other methods like nearest neighborhood method. The output pixel value is a weighted average of pixels in the nearest 2-by-2 neighborhood.

### 3.5.4 Thinning

Thinning operation provides the skeleton of the image, which is one pixel wide and it retains all the significant information related to the shape of the original pattern (Gonzalez). Morphological thinning operation was used for this purpose. It removes pixels so that an object without holes shrinks to a minimally connected stroke, and an object with holes shrinks to a connected ring halfway between each hole and the outer boundary.

## 3.6 Summary

In this chapter, a detailed description of the procedure adapted for collection and preparation of handwritten Gujarati character dataset is provided. Handwritten data samples are collected with the help of writers belonging to different age groups, education background and profession. The special data collection forms were designed for this purpose. Segmentation algorithms and preprocessing steps utilized for the extraction of isolated character/numeral images are also discussed. Preprocessing step enhances the quality of the character images and make them suitable for feature extraction algorithm. Several features utilized to represent handwritten Gujarati characters are discussed in next chapter.

# Chapter 4

## Feature Extraction Techniques

Feature extraction is a process in which we transform an image from space of all images to a new space, where the pattern recognition problem will be easier to solve (Bishop). For any character recognition system feature extraction is one of the very significant aspect in order to achieve high recognition performance. A robust feature extraction algorithm is required which should be able to handle diversity of instances of the same character. A usual approach is that to represent character pattern by a feature vector and then this feature vector is fed to the classifier to classify the feature vector into its classes. Therefore, the feature extraction algorithm maps the two dimensional image to a one dimensional feature vector. This feature vector contains most of the significant information of the image in order to provide small intra class variance and large inter class variance. Feature based approach is prevalent for handwritten character recognition due to the availability of large variation in writing style. There are two types of feature based approaches, first is based on extracting the features from spatial domain, and second is based on extraction of features from transform domain. In case of spatial domain based approaches, features are extracted directly from the pixel representation of the pattern. In this case, statistical and structural features are derived from character patterns. Statistical distribution of points provides the statistical features whereas geometric properties of the characters provides the structural features. Various natural traits of writing such as curves, loops, branch points, end points etc., provides the structural features while various mathematical measures computed over image or the part of the image, such as pixel densities, moments etc. provides the statistical features. In case of transform domain

based approaches, image is transformed to another space and then suitable features are extracted from the transformed images (Pal and Chaudhuri). Transform domain based approaches are preferred for printed character recognition while for handwritten character recognition spatial domain based techniques are more suitable (Casey and Nagy, “Advances in pattern recognition”).

Extraction of features for each of the handwritten character is the most important facet of the character recognition system. The efficiency of any character recognition technique is directly dependent on the effectiveness of the generated feature set that could uniquely represent a character. Feature set generation is the method of converting highly redundant, variable, and diverse data to a small-in-size, robust, abstract, and complete set that conveys all features of the original data. For character recognition, extracting those features that are essential to differentiate among characters is a tedious task. A simple method that generates the most appropriate and complete set of features is always needed. Structure of characters in Gujarati script contains curves, holes and strokes which make large variation in writing style by different persons. This leads to a situation, where it is very difficult to construct unique features for handwritten Gujarati characters. This chapter describes different kinds of feature extraction techniques utilized by us to represent handwritten Gujarati characters.

## 4.1 Feature Set Extraction based on Zoning

One of the feature extraction technique utilized was the zone based feature extraction technique (Impedovo and Pirlo). Zone based feature extraction technique started by dividing the bounding box of character image into uniform zones. Experiments were carried out with 256, 64, 16 and 4 zones (Sharma et al., “Comparative analysis of zoning based methods for Gujarati handwritten numeral recognition”). The size of the feature vector representing a character with this technique was exactly equal to the number of zones. Each zone contributed a value to the feature vector. A number of pixels which were part of the character encompassed by a zone determined this value. The idea for 256, 64, 16 and 4 zones is depicted in Figure 4.1. It is to be noted that, the best results were achieved while using 64 zones and therefore feature based on 64 zones was considered for generation of fusion features. Further to mention,

chapter 5 presents results when feature vector was built using 64 zones. The feature vector generated through this procedure is termed as feature vector generated through Zone based Features (ZbF) in the rest of this thesis. The procedure for constructing ZbF is summarized in Algorithm 3.

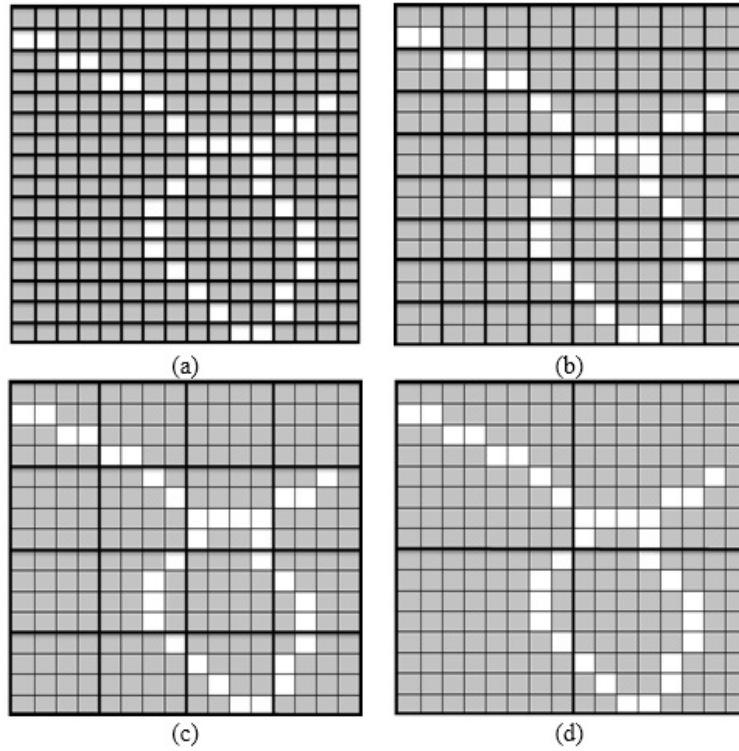


Figure 4.1: Numeral image divided in (a) 256 zones (b) 64 zones (c) 16 zones (d) 4 zones

---

**Algorithm 3** Constructing ZbF

---

**Input:** A preprocessed character image  $I_p$

**Output:** ZbF of length = 64 elements

- 1: Resize input image to a dimension  $16 \times 16$ . Designate the resized image as  $I_r$ .
  - 2: Apply thinning operation on the resized image to generate a 1-pixel wide thinned image. Denote this image as  $I_{rt}$ .
  - 3: Partition  $I_{rt}$  into 64 zones using equal partitioning method. This results in 64 zones, each of size  $2 \times 2$ .
  - 4: Each zone contributes a value in generation of ZbF. Number of pixels which are part of the character encompassed by a zone determine this value. Set of all these values form the ZbF.
  - 5: Return ZbF.
-

## 4.2 Feature Set Extraction based on Projection Profiles

The second technique for extracting features was based on various projection profiles of the character image (Desai, “Gujarati handwritten numeral optical character re-organization through neural network”). Horizontal, vertical, right diagonal and left diagonal projection profiles were used. Idea of these profiles for a  $3 \times 3$  box is depicted in Figure 4.2. This technique resulted in a feature vector of size 94 for each of the handwritten character image of size  $16 \times 16$ . Contribution of horizontal and vertical profiles was 16 features each while two diagonal profiles contributed 31 features each. This can easily be understood, as for a  $16 \times 16$  box, there are 16 horizontal and vertical projections while 31 left and 31 right diagonal projections. Each projection led to a feature indicated by a number of pixels encompassed by that projection in the character image. The feature vector constructed through this procedure is designated as Projection Profile based Features (PPbF) in the remainder of this thesis. The procedure for generating PPbF is summarized in Algorithm 4.

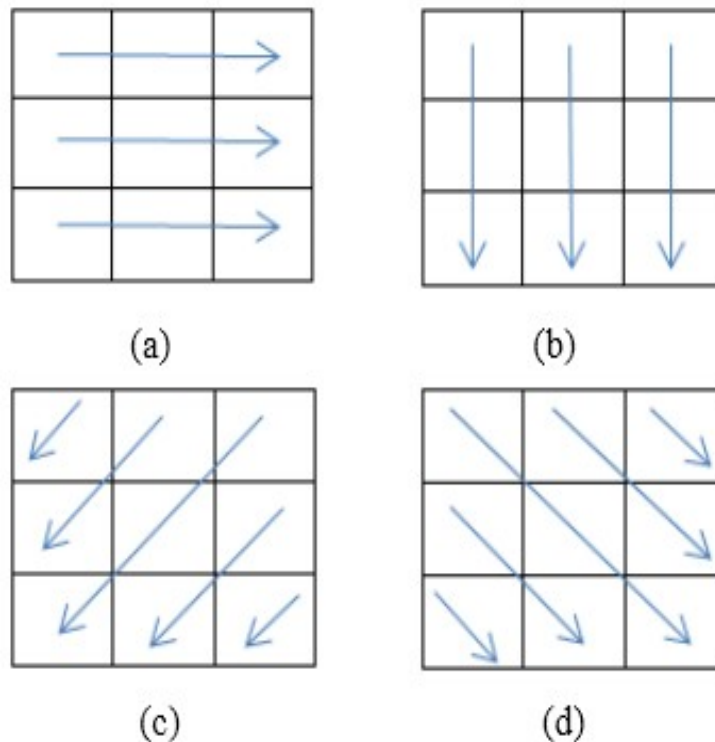


Figure 4.2: (a) represents horizontal profile. (b) represents vertical profile. (c) and (d) represent left diagonal and right diagonal profiles respectively.



---

**Algorithm 4** Constructing PPbF

---

**Input:** A preprocessed character image  $I_p$ **Output:** PPbF of length = 94 elements

- 1: Resize input image to a dimension  $16 \times 16$ . Designate the resized image as  $I_r$ .
  - 2: Apply thinning operation on the resized image to generate a 1-pixel wide thinned image. Denote this image as  $I_{rt}$ .
  - 3: Compute horizontal, vertical, right diagonal and left diagonal projection profiles form  $I_{rt}$  and then generate PPbF by concatenating values of all four projection profiles.
  - 4: Return PPbF.
- 

### 4.3 Feature Set Extraction based on String of Chain Code

The chain code feature was first proposed by H. Freeman and therefore sometimes referred as Freeman code or Freeman chain code (Freeman). Chain code based features have shown effective performance in the recognition of character shapes (Sharma et al., “Recognition of off-line handwritten Devnagari characters using quadratic classifier” Bhattacharya and Chaudhuri Bhattacharya, Ghosh, and Parui). It simply represents the movements of boundary segments in terms of integer numbers. A method based on horizontal scanning of a character image is proposed to uniformly identify beginning of chain code based feature. The idea is depicted in Figure 4.3 and Figure 4.4 for one of the numeral and character image respectively. It can be realised that the image was scanned from bottom to top and left to right until a pixel that was part of the character’s skeleton was found. As soon as such pixel was found, scanning stopped and that pixel was marked as the first pixel. At the same time, the element of chain code corresponding to this first pixel was marked as the beginning of the sequence of chain code. The rest of the elements of chain code sequence were obtained by following the skeleton in a clockwise direction from the identified beginning. The maximum length of the sequence of chain code was fixed to 100 as it was observed that the length of chain code for any character was never more than 100. If the length of the constructed chain code was less than 100, remaining elements were considered 0. Finally, the feature vector consisting of 100 elements was extracted. The feature vector constructed through this procedure is termed as String of Chain Code (SoCC) in the rest of the thesis. The procedure for constructing SoCC

is summarized in Algorithm 5.

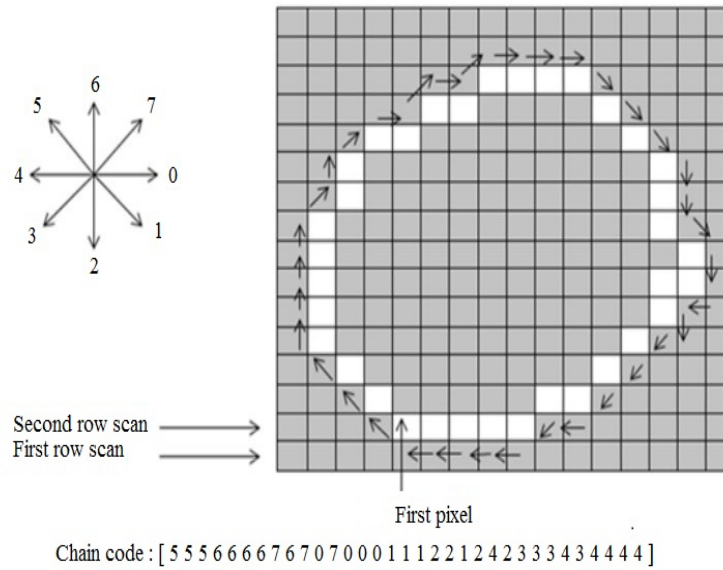


Figure 4.3: Chain code obtained by finding the starting point through horizontally scanning for numeral images.

---

#### Algorithm 5 Constructing SoCC

---

**Input:** A preprocessed character image  $I_p$

**Output:** SoCC of length = 100 elements

- 1: Resize input image to a dimension  $16 \times 16$ . Designate the resized image as  $I_r$ .
  - 2: Apply thinning operation on the resized image to generate a 1-pixel wide thinned image. Denote this image as  $I_{rt}$ .
  - 3: For image  $I_{rt}$ , through the method based on horizontal scanning, identify the beginning of chain code sequence and then compute string of chain code (SoCC) sequence.
  - 4: Return SoCC.
- 

## 4.4 Feature Set Extraction using Fusion of Features

Fusion of SoCC, PPbF and ZbF features is proposed for learning prediction models. Overall process adapted for Gujarati handwritten character recognition using fusion features is summarized in Figure 4.5. As fusion features, all possible combinations of SoCC, PPbF and ZbF were tried. Each of the prediction model was experimented with four fusion feature sets. Individual feature sets and fusion feature sets are summarized in Table 4.1.

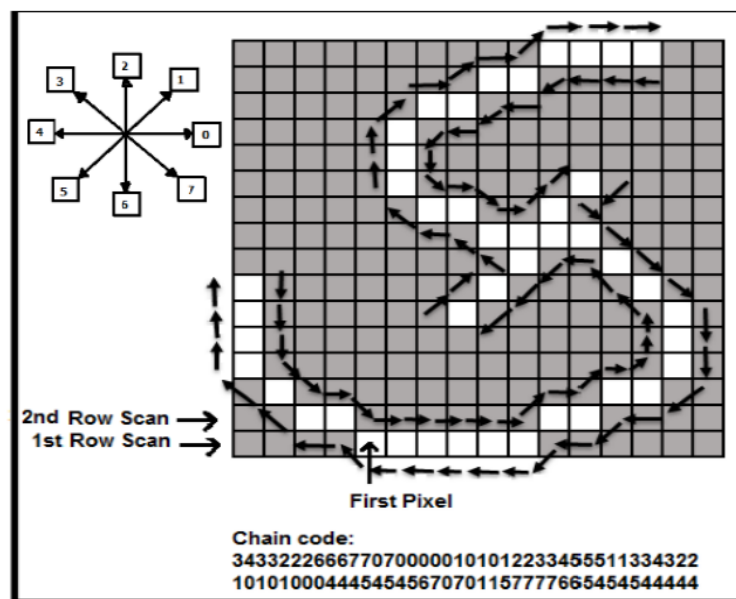


Figure 4.4: Chain code obtained by finding the starting point through horizontally scanning for character images.

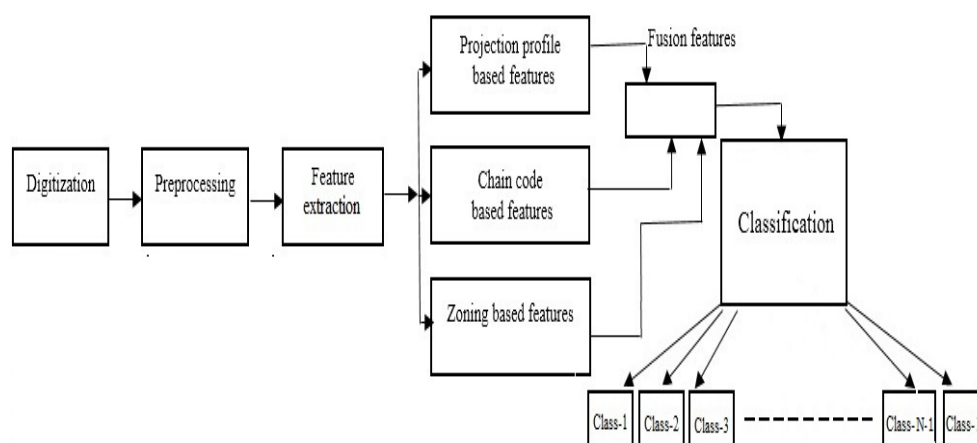


Figure 4.5: Gujarati handwritten character recognition process using fusion features

Table 4.1: Feature sets

Combinations	Name of Features	Acronym	Number of Features
Feature Set 1	String of Chain Code	SoCC	100
Feature Set 2	Projection Profile based Features	PPbF	94
Feature Set 3	Zone based Features	ZbF	64
Feature Set 4 - Fusion Features 1	SoCC + PPbF	CP	194
Feature Set 5 - Fusion Features 2	SoCC + ZbF	CZ	164
Feature Set 6 - Fusion Features 3	PPbF + ZbF	PZ	158
Feature Set 7 - Fusion Features 4	SoCC + PPbF + ZbF	CPZ	258

## 4.5 Feature Set Extraction using Structural Decomposition Technique

Characters in Indian scripts consist of higher number of constituent components as compared to English. Also, all the characters in English script consist of single connected component, while in Indian script like Gujarati, there are large number of characters which consist of more than one connected components which is evident in Figure 3.1. Owing to this significant difference, we proposed to extract features from individual components and then to combine all these features to generate the final feature vector.

Figure 4.6 shows the individual components of one sample handwritten Gujarati character image. Components were extracted by scanning character's image from left to right. For example, component 1 was the left most component followed by component 2 and so on. For each of the individual components of this character, features such as aspect ratio, extent, arch-chord ratio and chain code sequence were extracted.

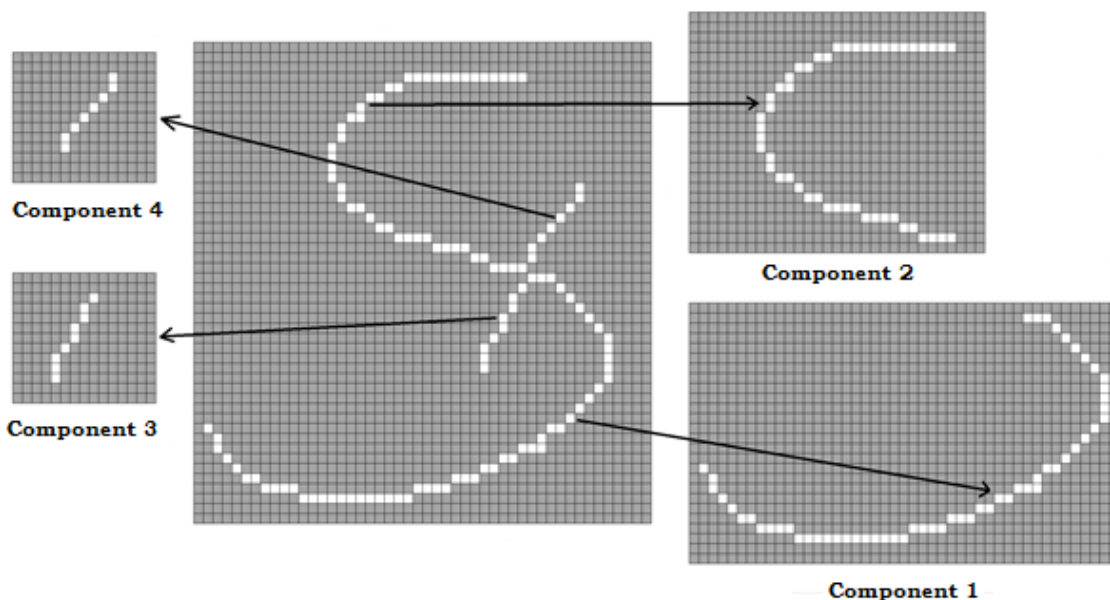


Figure 4.6: Constitutional components of one of the Gujarati character

Figure 4.7 shows the representation of parameters extracted from one of the component image. Aspect ratio is the ratio of height of component image ( $H$ ) to the width of the component image ( $W$ ). Extent was calculated as the ratio of total

numbers of ON (foreground) pixels to the total number of pixels in a bounding box that fits the component. Tortuosity was also used as the feature and the simplest mathematical method to estimate tortuosity is the arc-chord ratio. Arch-chord ratio was calculated by dividing the length of the component ( $L$ ) by the Euclidian distance ( $D$ ) between the two end points of the components. Aspect ratio, extent and arch-chord ratio were used as the first three elements of the component feature vector. Remaining elements of the component feature vector were obtained by calculating the directional chain code sequence for the component image (Freeman). Chain code sequence generation for the component image is shown in Figure 4.7.

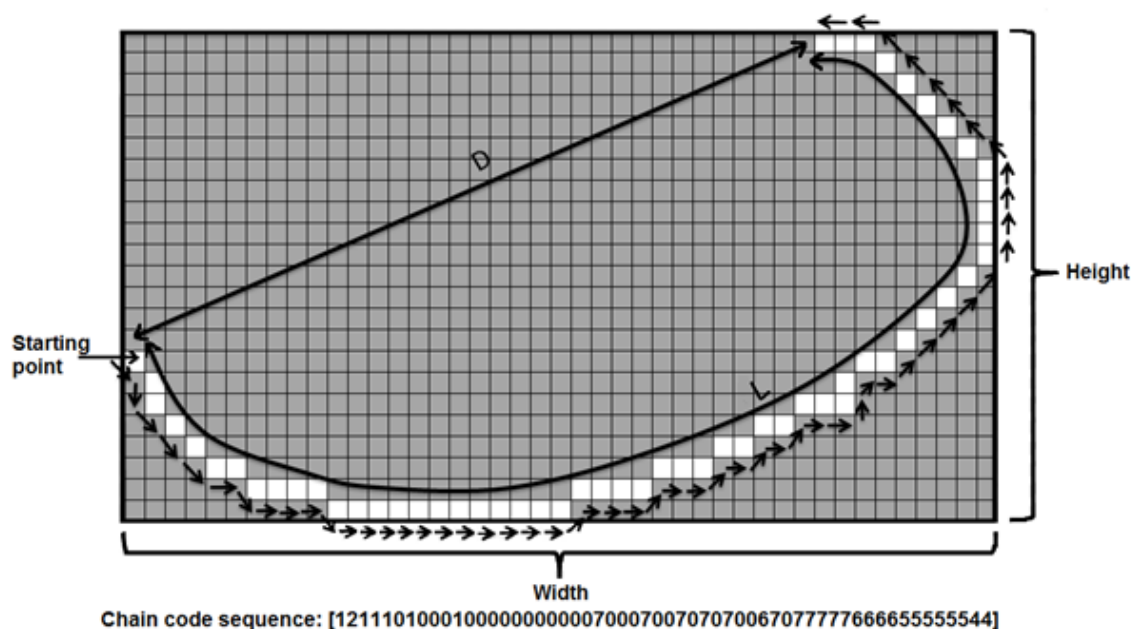


Figure 4.7: Features extracted from component 1 image

These four features were extracted from all the constituent components of the character image and a complete feature vector was generated by concatenating the features extracted from each of the components of the character. Concatenation was achieved by joining the features of components in the following order: Features of component 1, Features of component 2,  $\dots$ , Features of component  $N$ . The feature vector constructed through this procedure was designated as feature vector through Structural Decomposition (SD) in the remainder of this thesis.

The procedure for constructing SD is summarized in Algorithm 6. It is important to mention that maximum length of the feature vector was fixed to 210 as it was observed that the length of the feature vector for any character was never more than

210. If the length of the constructed feature vector was less than 210, remaining positions were filled with 0.

---

**Algorithm 6** Constructing SD
 

---

**Input:** A preprocessed character image  $I_p$

**Output:** SD of length = 210 elements

















- 1: Resize input image to a dimension  $48 \times 48$ . Designate the resized image as  $I_r$ .
  - 2: Apply thinning operation on the resized image to generate a 1-pixel wide thinned image. Denote this image as  $I_{rt}$ .
  - 3: Identify components of  $I_{rt}$ .
  - 4: Compute aspect ratio, extent, arch-chord ratio and chain code sequence for individual component image.
  - 5: Repeat step 4 for each of the components.
  - 6: Construct SD by concatenating features of component 1, component 2,  $\dots$ , component N in exactly this order.
  - 7: Return SD
- 

## 4.6 Feature Set Extraction by Zone Pattern Matching

The next feature extraction technique that is proposed in this thesis is based on Zone Pattern Matching. The feature vector constructed through this technique is designated as feature vector through Zone Pattern Matching (ZPM) in the remainder of this thesis.

The method begins by dividing the bounding box of preprocessed character image in 576 uniform non-overlapping zones of size  $2 \times 2$ . Each zone contributed a pattern value to the feature vector. A pattern value was assigned to each zone as per the details given in Table 4.2. It is evident from Table 4.2 that there are 16 unique pattern values and each zone was matched against these 16 patterns to decide the best matching pattern. A pattern value of the pattern which matches the best with a zone was assigned to a zone. Feature vector was generated by combining all these assigned pattern values. This leads to a feature vector of length 576. The procedure is summarized in Algorithm 7.

Table 4.2: Patterns &amp; their representation and value

Sr.	Pattern and its Representation		Binary Representation	Pattern Value				
	Pattern	Representation						
1		<table border="1" data-bbox="603 465 694 537"> <tr><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td></tr> </table>	0	0	0	0	0000	0
0	0							
0	0							
2		<table border="1" data-bbox="603 560 694 631"> <tr><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td></tr> </table>	0	0	0	1	0001	1
0	0							
0	1							
3		<table border="1" data-bbox="603 654 694 725"> <tr><td>0</td><td>0</td></tr> <tr><td>1</td><td>0</td></tr> </table>	0	0	1	0	0010	2
0	0							
1	0							
4		<table border="1" data-bbox="603 748 694 819"> <tr><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td></tr> </table>	0	0	1	1	0011	3
0	0							
1	1							
5		<table border="1" data-bbox="603 842 694 913"> <tr><td>0</td><td>1</td></tr> <tr><td>0</td><td>0</td></tr> </table>	0	1	0	0	0100	4
0	1							
0	0							
6		<table border="1" data-bbox="603 936 694 1008"> <tr><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td></tr> </table>	0	1	0	1	0101	5
0	1							
0	1							
7		<table border="1" data-bbox="603 1030 694 1102"> <tr><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td></tr> </table>	0	1	1	0	0110	6
0	1							
1	0							
8		<table border="1" data-bbox="603 1124 694 1196"> <tr><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td></tr> </table>	0	1	1	1	0111	7
0	1							
1	1							
9		<table border="1" data-bbox="603 1218 694 1290"> <tr><td>1</td><td>0</td></tr> <tr><td>0</td><td>0</td></tr> </table>	1	0	0	0	1000	8
1	0							
0	0							
10		<table border="1" data-bbox="603 1312 694 1384"> <tr><td>1</td><td>0</td></tr> <tr><td>0</td><td>1</td></tr> </table>	1	0	0	1	1001	9
1	0							
0	1							
11		<table border="1" data-bbox="603 1406 694 1478"> <tr><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td></tr> </table>	1	0	1	0	1010	10
1	0							
1	0							
12		<table border="1" data-bbox="603 1500 694 1572"> <tr><td>1</td><td>0</td></tr> <tr><td>1</td><td>1</td></tr> </table>	1	0	1	1	1011	11
1	0							
1	1							
13		<table border="1" data-bbox="603 1594 694 1666"> <tr><td>1</td><td>1</td></tr> <tr><td>0</td><td>0</td></tr> </table>	1	1	0	0	1100	12
1	1							
0	0							
14		<table border="1" data-bbox="603 1688 694 1760"> <tr><td>1</td><td>1</td></tr> <tr><td>0</td><td>1</td></tr> </table>	1	1	0	1	1101	13
1	1							
0	1							
15		<table border="1" data-bbox="603 1783 694 1854"> <tr><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td></tr> </table>	1	1	1	0	1110	14
1	1							
1	0							
16		<table border="1" data-bbox="603 1877 694 1948"> <tr><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td></tr> </table>	1	1	1	1	1111	15
1	1							
1	1							

---

**Algorithm 7** Constructing ZPM

---

**Input:** A preprocessed character image  $I_p$ **Output:** ZPM of length = 576 elements

- 1: Resize input image to a dimension  $48 \times 48$ . Designate the resized image as  $I_r$ .
  - 2: Apply thinning operation on the resized image to generate a 1-pixel wide thinned image. Denote this image as  $I_{rt}$ .
  - 3: Partition  $I_{rt}$  into 576 blocks/zones using equal partitioning method. This results in 576 zones, each of size  $2 \times 2$ .
  - 4: Compare a zone with 16 possible patterns (shown in Table 4.2). Identify the best matching pattern and consider its pattern value as the value of the block. Repeat this procedure for all the zones.
  - 5: Generate ZPM by concatenating pattern values of all the zones.
  - 6: Return ZPM.
- 

## 4.7 Feature Set Extraction by Normalised Cross Correlation

Cross correlation between two images is a well-known tool for matching two images (Lewis). It tells the degree to which two images are similar. Assume that A and B are images of size  $m \times n$  and  $p \times q$  such that  $p \leq m$  and  $q \leq n$ .

Normalized cross-correlation between A and B is defined as in Equation 4.1.

$$\lambda(x, y) = \frac{\sum_s \sum_t \delta_{A(x+s, y+t)} \delta_{B(s, t)}}{\sum_s \sum_t \delta_{A(x+s, y+t)}^2 \sum_s \sum_t \delta_{B(s, t)}^2} \quad (4.1)$$

where,  $\delta_{A(x+s, y+t)} = A(x + s, y + t) - \bar{A}(x, y)$

$\delta_{B(s, t)} = B(s, t) - \bar{B}$

$s \in \{1, 2, 3, \dots, p\}$  and  $t \in \{1, 2, 3, \dots, q\}$

$x \in \{1, 2, 3, \dots, m - p + 1\}$  and  $y \in \{1, 2, 3, \dots, n - q + 1\}$

$\bar{A}(x, y) = \frac{1}{pq} \sum_s \sum_t A(x + s, y + t)$

$\bar{B} = \frac{1}{pq} \sum_s \sum_t B(s, t)$

The value of cross-correlation coefficient  $\lambda$  ranges in  $[-1, +1]$ . A value of +1 indicates that B is in complete match with  $A(x, y)$  and  $-1$  shows complete disagreement. The matching process is carried out by sliding image B over A and  $\lambda$  is calculated for each coordinate  $(x, y)$ . After calculating  $\lambda$  for each point, the point which exhibits maximum  $\lambda$  is considered as  $\lambda_{max}$ .  $\lambda_{max}$  was calculated between different flipped and



rotated versions of the preprocessed image as per the details in Table 4.3. Feature vector was formed by combining these 34 values of  $\lambda_{max}$ .

To maintain uniformity, all the images were resized to  $48 \times 48$ . Left half ( $48 \times 24$ ), Right half ( $48 \times 24$ ), upper half ( $24 \times 48$ ) and lower half ( $24 \times 48$ ) of every image were extracted. Flipped and rotated versions of all the half images were generated and shown in Figure 4.8.

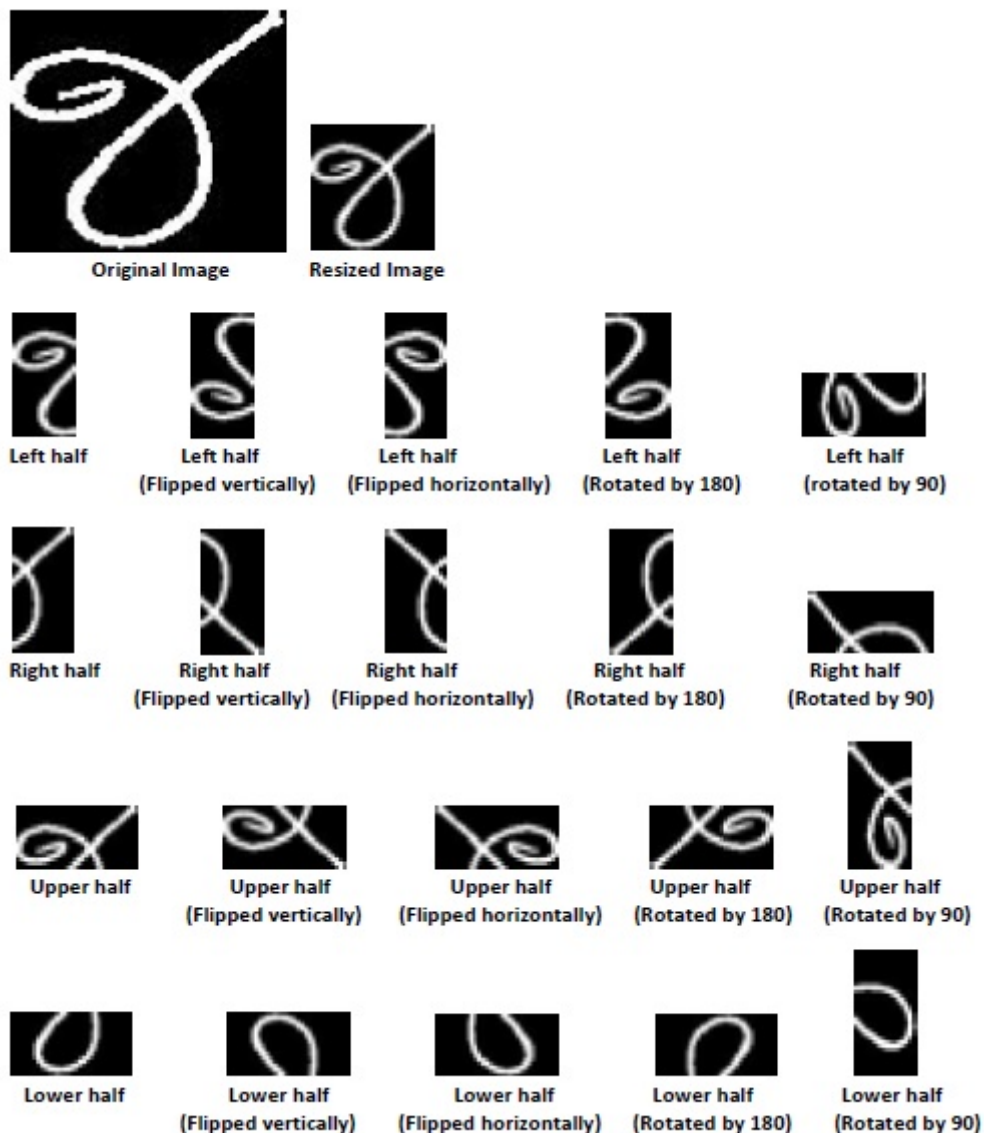


Figure 4.8: Original, Resized and various Rotated and Flipped versions of Image Halves

For every image, a feature vector of length 34 was obtained. The feature vector generated through this procedure is termed as feature vector generated through Normalized Cross Correlation (NCC) in the rest of this thesis. The procedure for

Table 4.3: Image combinations chosen for feature vector generation

Sr. No.	First half of image	Second half of image
1	Left Half	Right half
2	Left Half	Vertical flip of Left half
3	Left Half	Horizontal flip of Left half
4	Left Half	Left half rotated by 180
5	Left Half	Vertical flip of Right half
6	Left Half	Horizontal flip of Right half
7	Left Half	Right half rotated by 180
8	Right half	Vertical flip of Right half
9	Right half	Horizontal flip of Right half
10	Right half	Right half rotated by 180
11	Right half	Vertical flip of Left half
12	Right half	Horizontal flip of Left half
13	Right half	Left half rotated by 180
14	Upper half	Lower half
15	Upper half	Vertical flip of Upper half
16	Upper half	Horizontal flip of Upper half
17	Upper half	Upper half rotated by 180
18	Upper half	Vertical flip of Lower half
19	Upper half	Horizontal flip of Lower half
20	Upper half	Lower half rotated by 180
21	Lower half	Vertical flip of Lower half
22	Lower half	Horizontal flip of Lower half
23	Lower half	Lower half rotated by 180
24	Lower half	Vertical flip of Upper half
25	Lower half	Horizontal flip of Upper half
26	Lower half	Upper half rotated by 180
27	Left half rotated by 90	Upper half
28	Left half rotated by 90	Lower half
29	Right half rotated by 90	Upper half
30	Right half rotated by 90	Lower half
31	Upper half rotated by 90	Left half
32	Upper half rotated by 90	Right half
33	Lower half rotated by 90	Left half
34	Lower half rotated by 90	Right half

constructing NCC is summarized in Algorithm 8.

---

**Algorithm 8** Constructing NCC
 

---

**Input:** A preprocessed character image  $I_p$

**Output:** NCC of length = 34 elements

- 1: Resize input image to a dimension  $48 \times 48$ . Designate the resized image as  $I_r$ .
  - 2: Divide the image into upper half, lower half, right half and left half.
  - 3: Generate the feature vector by taking all the combinations of normalized cross correlation as shown in Table 4.3.
  - 4: Return NCC.
- 

## 4.8 Summary

In this chapter, we have demonstrated the feature sets utilized to represent the shape of handwritten Gujarati character images. Experiments were carried out using different kinds of features and their fusion. Zone based (ZbF), projection profiles based (PPbF) and chain code based (SoCC) features are employed as individual features. It is also proposed to use a fusion of these features for learning prediction models. Three new features are also proposed to represent handwritten Gujarati characters. These features include features extracted based on Structural Decomposition (SD), Zone Pattern Matching (ZPM) and Normalized Cross Correlation (NCC). These features are utilized for learning different prediction models. These prediction models are discussed in next chapter.



# Chapter 5

## Machine Learning Approaches

Machine learning approaches are utilized for the classification of handwritten Gujarati numerals and characters. Classification process is the final stage of the handwritten character recognition system. The task of a classifier is to use the feature vector provided by the feature extractor, and to assign the object to a predefined group or class label. The performance of a classifier depends on the efficiency and selection of features. A training set having huge number of labelled objects of each class is used for classifier learning purpose. The purpose of the classifier is to predict the correct class label for unknown character pattern. Prediction models that were used in this study included Naive Bayes (NB), Artificial Neural Networks (ANN) and Support Vector Machines (SVM).

Bayesian classifier is one of the most fundamental and simple classifier which is based on Bayes' theorem. Artificial Neural Networks and Support Vector Machines are very popular classification algorithms. ANN is being used from late 1980s (LeCun et al., "Backpropagation applied to handwritten zip code recognition") while SVM is popular since late 1990s (Burges). In many cases, SVM outperforms classical neural networks in classification problems (Soman, Diwakar, and Ajay). In case of SVM, the learning task is insensitive to the training samples and performs fine with high dimensional data.

### 5.1 Naive Bayes Classifier

Bayesian classifier predicts the probability of a given test observation belonging to a particular class. This probability is predicted with the help of Bayes' theorem. Bayes'

theorem provides a way to calculate the posterior probability,  $P(C|X)$ , from  $P(C)$ ,  $P(X|C)$  and  $P(X)$  using the below mentioned Equation 5.1.

$$P(C|X) = \frac{P(X|C)P(C)}{P(X)} \quad (5.1)$$

$P(C|X)$  is the probability of hypothesis  $C$  being true given that event  $X$  has occurred. In this study, hypothesis  $C$  denoted a class from the set of probable classes  $0, 1, 2, \dots, N$  and an event  $X$  was a test image. In case of numeral classification value of  $N$  will be 10 depending on number of classes.  $P(X|C)$  is a conditional probability of occurrence of event  $X$  given that hypothesis  $C$  is true and it can be estimated from the training data. Working of naive Bayesian classifier is described below.

Assume that  $X$  denotes the test data and there are  $m$  classes  $C_1, C_2, \dots, C_m$ . Bayesian classifier classifies  $X$  to a class with the highest probability. As per Bayes' theorem

$$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)}, i = 1, 2, \dots, m \quad (5.2)$$

Calculating  $P(X|C_i)$  is computationally extremely expensive when data set has many attributes  $A_1, A_2, \dots, A_n$ . Naive assumption of class conditional independence is made to reduce the computational complexity. This assumption states that values of attributes are independent of one another, given the class label of the observation. Therefore,

$$\begin{aligned} P(X|C_i) &= \prod_{k=1}^n P(x_k|C_i) \\ &= P(x_1|C_i) \times P(x_2|C_i) \times \dots \times P(x_n|C_i) \end{aligned} \quad (5.3)$$

Here,  $x_k$  is used to specify the value of attribute  $A_k$  for observation  $X$ .  $P(x_k|C_i)$  depends on the fitted probability distribution to  $A_k$ . If  $A_k$  is categorical, some discrete probability distribution is fitted to it, while when it is continuous, it is assumed

that  $A_k$  follows some probability density function. In this study, as each  $A_k$  was continuous, it could be assumed that each of them followed some continuous distribution. Gaussian distribution was assumed as that continuous distribution in this study. Based on this assumption, value of  $P(x_k|C_i)$  is calculated using Equation 5.4.

$$f(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2}$$

*so that,* (5.4)

$$P(x_k|C_i) = f(x_k, \mu_{C_i}, \sigma_{C_i})$$

Here  $\mu_{C_i}$  and  $\sigma_{C_i}$  designate mean and standard deviation of attribute  $A_k$  for training observations of class  $C_i$ .  $P(x_k|C_i)$  is then estimated by plugging these two quantities in Equation 5.4 along with  $x_k$ . In order to predict the class label of X,  $P(X|C_i)P(C_i)$  is evaluated for each class  $C_i$ . X is predicted to belong to class  $C_i$ , if and only if

$$P(X|C_i)P(C_i) > P(X|C_j)P(C_j) \quad \text{for } 1 \leq j \leq m; j \neq i. \quad (5.5)$$

## 5.2 Support Vector Machine

SVM focuses on identifying maximum margin hyper-plane as the final decision boundary to separate between positive and negative classes. SVM is essentially a binary classifier. SVM can be enabled for multiclass classification using one-versus-all strategy. This strategy approaches multiclass classification problem by employing series of binary classifiers. A number of binary classifiers (SVM in this study) is equal to the number of classes among which separation is desired.

If total number of classes is  $m$ , then  $m$  SVMs are learnt. For  $i^{th}$  classifier, all observations of training set which belong to class  $C_i$  are considered as positive examples while all remaining observations are considered as negative examples. A test observation is given to each of the SVMs resulting into  $m$  predictions. The prediction with the highest confidence value is considered as the final prediction. It is to be noted that confidence of the prediction is proportional to the distance from

the separating boundary. More the distance from the separating boundary, more the confidence of the prediction.

Each individual binary SVM can be understood from the following discussions. Assume that  $x_i \in R^d, i = 1, 2, \dots, N$  forms a training set with corresponding class labels  $y_i \in \{+1, -1\}, i = 1, 2, \dots, N$ . These training set examples can be further mapped to higher dimensional feature space  $\Phi(x_i) \in H$ . A kernel function  $K(x_i, x_j)$  can perform this mapping  $\phi(\cdot)$ . Resulting decision boundary is then defined as in Equation 5.6.

$$f(x) = \text{sgn}\left(\sum_{i=1}^N y_i \alpha_i \cdot K(x, x_i) + b\right) \quad (5.6)$$

Values of  $\alpha_i$  can be obtained by solving quadratic programming problem shown in Equation 5.7.

$$\begin{aligned} \text{Maximize } & \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j \cdot y_i y_j \cdot K(x_i, x_j) \\ & \text{Subject to } 0 \leq \alpha_i \leq c \\ & \sum_{i=1}^N \alpha_i y_i = 0, i = 1, 2, \dots, N \end{aligned} \quad (5.7)$$

In the above equation,  $c$  is the regularization parameter which controls the trade-off between margin and misclassification error. The polynomial kernel as shown in Equation 5.8 was also employed in this study.

$$\text{Polynomial Function : } K(x_i, x_j) = (x_i \cdot x_j + 1)^d \quad (5.8)$$

where  $d$  is the degree of a polynomial. It can be understood that setting  $d = 1$  in Equation 5.8 yields a linear kernel.

Degree ( $d$ ) of a kernel function and regularization constant ( $c$ ) were considered as design parameters of polynomial SVM. In case of linear SVM, obviously, only regularization constant ( $c$ ) was considered as the design parameter. To determine them efficiently, 5-fold cross-validation of training set was used. During the 5-fold



cross validation of training set, three values of  $d$  and five values of  $c$  were tried. These parameters and their values which were tested are summarized in Table 5.1. The combination of parameters that resulted into best cross-validation performance of training set was considered as the best combination and images in testing set were classified using SVM learnt with these values of parameters.

Table 5.1: SVM design parameters and their values tested in cross validation of training set

Parameters	Values
Degree of Kernel Function ( $d$ )	2, 3, 4
Regularization Parameter ( $c$ )	0.01, 0.1, 1, 10, 100

### 5.3 Artificial Neural Network

A three layer feed forward neural network was utilized for all the experiments. Image of a character described by feature vector was an input to the ANN. A number of neurons in the input layer was set to the number of features in the feature vector. Each neuron in an output layer was representing a class and the one producing the maximum output for the given input was considered the winning neuron. A class label corresponding to the winning neuron was considered as the predicted class label for the given input. The transfer function of each of the neurons in the output layer was log sigmoid while tan sigmoid was employed as the transfer function of neurons of hidden layer. The architecture of the ANN used is depicted in Figure 5.1.

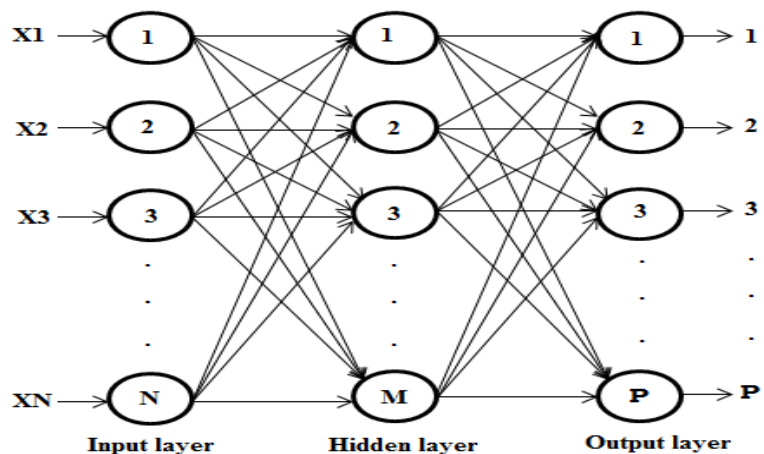


Figure 5.1: Neural network architecture

The adaptive gradient descent was used for updating weights. This rule allowed to change learning rate during the training process which might help in improving the performance. In this algorithm, first of all, initial network output and error were calculated. Next step involved computing new weights and biases using the current learning rate. New output and error were then calculated using these new weights and biases. New weights and biases were discarded and the learning rate was decreased (to 70% of its current value, in this study), if the new error exceeded the old error by more than a predefined ratio (1.04, in this study). Otherwise, learning rate was increased (by 5% of the current value, in the experiments reported in this study) and new weights and biases were kept. The procedure confirmed that the learning rate was increased only to the degree that the network could learn without large increases in error. Near optimal learning rate for the local terrain can be obtained this way. At the same time, learning rate was increased as long as stable learning was assured. When learning rate was too high to assure a decrease in error, it was decreased until stable learning began again.

A number of epochs and number of neurons in the hidden layer were considered as the design parameters of the ANN. To determine them efficiently, 5-fold cross-validation of training set was used. These parameters and their values which were tested are summarized in Table 5.2.

Table 5.2: ANN design parameters and their values tested in cross validation of training set

Parameters	Values
Number of epochs ( $ep$ )	1000, 2000, $\dots$ , 10000
Number of hidden layer neurons ( $n$ )	10, 20, $\dots$ , 100

The combination of parameters that resulted into best cross-validation performance of training set was considered as the best combination and images in the testing set were classified using ANN learnt with these values of parameters.

## 5.4 Summary

In this chapter, we have discussed various prediction models utilized by us for classification purpose. Methods based on Artificial Neural Network (ANN), Support Vector

Machine (SVM) and Naive Bayes (NB) classifier are demonstrated along with the design parameters considered in this study. These prediction models are exercised with various features discussed in chapter 4. Experimental results achieved on handwritten Gujarati character and numeral datasets are represented in next chapter.



# Chapter 6

## Results and Discussion

Handwritten Gujarati numeral and character recognition is attempted through prediction models learnt through individual features and also with fusion features. String of Chain Code (SoCC), Zone based Features (ZbF) and Projection Profile based Features (PPbF) were tried as the individual feature vectors. The task was also addressed using fusion features. As fusion features, all possible combinations of SoCC, PPbF and ZbF as shown in Table 4.1 were tried. Experiments were carried out using three prediction models namely, naive Bayes classifier, Support Vector Machines, and Artificial Neural Network. SVM was employed with linear and polynomial kernels. The overall adapted process is shown in Figure 4.5.

Methods based on Structural Decomposition (SD), Zone Pattern Matching (ZPM) and Normalized Cross Correlation (NCC) are also proposed for feature vector generation. Proposed algorithms consist of operations which include digitization, preprocessing, feature extraction and classification. The overall process adapted for handwritten Gujarati character recognition using all these proposed features is summarized in Figure 6.1.

### 6.1 Experimental Evaluation

The performance of proposed models were evaluated through accuracy and f-measure. Computation of f-measure requires Precision and Recall which can be estimated from True Positives (TP), False Positives (FP), True Negatives (TN) and False Negatives (FN). These measures are defined in Equations 6.1 - 6.4.

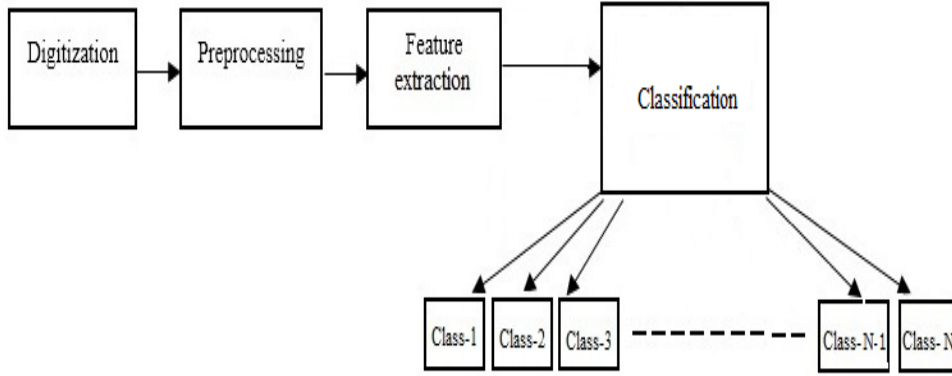


Figure 6.1: Handwritten Gujarati character recognition process

$$Precision_{positive} = \frac{TP}{TP + FP} \quad (6.1)$$

$$Precision_{negative} = \frac{TN}{TN + FN} \quad (6.2)$$

$$Recall_{positive} = \frac{TP}{TP + FN} \quad (6.3)$$

$$Recall_{negative} = \frac{TN}{TN + FP} \quad (6.4)$$

Precision is defined as the weighted average of  $Precision_{positive}$  and  $Precision_{negative}$  while weighted average of  $Recall_{positive}$  and  $Recall_{negative}$  estimates the Recall. The two measures, precision and recall are used together in f-measure to provide a single measurement for a system. F-measure is the harmonic mean of precision and recall. Accuracy and f-measure were estimated through the Equations 6.5 and 6.6 respectively.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (6.5)$$

$$F - measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (6.6)$$

For each of the experiment, the dataset was divided into a training set and a testing set, where 80% observations (images) constituted training set while remaining observations formed testing set. It is already mentioned in Chapter 5 that SVM and

ANN prediction models have some design parameters and proper tuning of these parameters is extremely important for the prediction performance of these models. 5-fold cross validation of training set was performed to identify the Best Values of Design Parameters (BVoDP) for these models. BVoDPs are the values of design parameters of classifier for which highest recognition accuracy is achieved by the classifier algorithm when 5-fold cross validation of training set was applied. Once the best value for these design parameters were identified, prediction models were configured according to these values and then, so configured prediction models were learnt through the entire training set.

### 6.1.1 Performance Evaluation using SoCC, ZbF and PPbF

For experimentation purpose, the generated dataset (HGND and HGCD-1), as specified in Table 3.1 was used. Uniform distribution of samples per class was provided for both numeral and character datasets. Numeral dataset images were uniformly distributed over ten classes as shown in Figure 3.6, which led to a total of 1400 images of each class. Similarly character dataset images were uniformly distributed over 44 classes as shown in Figure 3.2, which led to a total of 2000 images of each class.

Tables 6.1 and 6.2 show performance of prediction models on handwritten Gujarati numeral dataset and character dataset respectively. These tables summarize the results when these prediction models were learnt through SoCC, ZbF and PPbF features. For SVM and ANN, BVoDPs are also indicated in these tables.

For Numeral dataset with SoCC feature, accuracy values achieved with Linear SVM, Polynomial SVM, Naive Bayes and ANN are 98.41%, 99.25%, 94.70% and 96.37% respectively. Average of these accuracy values is 97.19%, which is shown in Figure 6.2. In the same manner accuracy values averaged over prediction models for individual features are depicted in Figure 6.2 (for HGND) and Figure 6.3 (for HGCD-1).

It is evident from these figures that irrespective of the prediction models, SoCC has a definite edge over other individual features and representations. This is remarkable as it establishes SoCC as a robust feature.

Performance of prediction models with SoCC, ZbF and PPbF features on individual numerals are shown in Tables 6.3, 6.4 and 6.5 respectively. Also, the performance

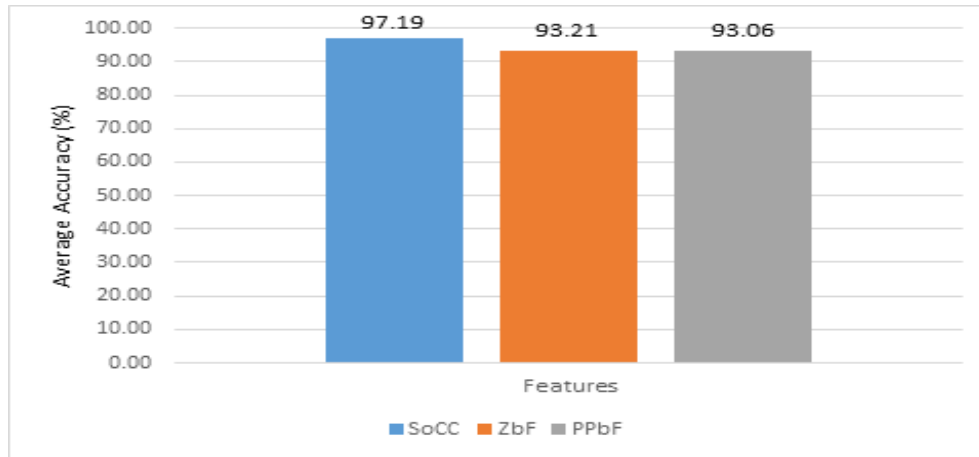


Figure 6.2: Accuracy values averaged over prediction models learnt using SoCC, PPbF and ZbF features (on HGND)

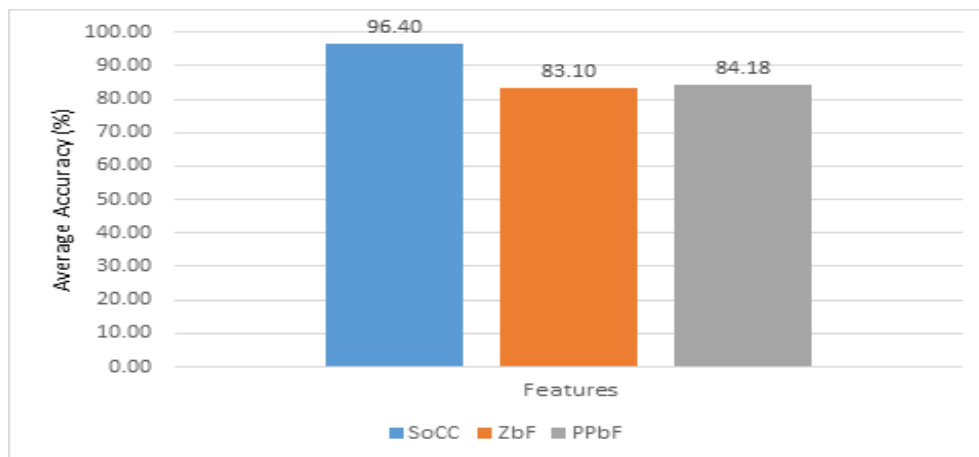


Figure 6.3: Accuracy values averaged over prediction models learnt using SoCC, PPbF and ZbF features (on HGCD-1)



Table 6.1: Performance of prediction models on HGND with SoCC, PPbF and ZbF features

Representation	Prediction Models					
	Linear SVM			Polynomial SVM		
	Accuracy (%)	F-measure (%)	BVoDP	Accuracy (%)	F-measure (%)	BVoDP
SoCC	98.41	98.39	c=0.01	99.25	99.22	c=0.01, d=4
PPbF	93.41	93.51	c=0.01	94.87	94.80	c=1, d=2
ZbF	94.57	94.58	c=0.1	97.07	97.06	c=10, d=2

Representation	Prediction Models					
	Naive Bayes			ANN		
	Accuracy (%)	F-measure (%)	BVoDP	Accuracy (%)	F-measure (%)	BVoDP
SoCC	94.70	94.62	-	96.37	96.28	n=20, ep=8000
PPbF	90.29	90.27	-	93.66	93.51	n=90, ep=3000
ZbF	86.85	86.31	-	94.35	94.34	n=80, ep=5000

of prediction models with same features on individual characters are represented in Tables 6.6, 6.7 and 6.8.

Accuracy values averaged over prediction models learnt using SoCC, ZbF and PPbF features for each numeral is represented in Figures 6.4, 6.5 and 6.6 respectively. In the similar way, accuracy values averaged over prediction models learnt using individual features for each character is depicted in Figures 6.7, 6.8 and 6.9.

### 6.1.2 Performance Evaluation using Fusion Features

Performance of prediction models with fusion features are depicted in tables 6.9 and 6.10 for handwritten numeral and character dataset (HGND and HGCD-1) respectively. Performance achieved with individual features (represented in tables 6.1 and 6.2) is also shown in these tables in order to indicate the comparison of recognition accuracies obtained by individual features and fusion features. It can be perceived that the highest recognition accuracy values of 99.92% and 99.75% were achieved for numeral dataset (HGND) and character dataset (HGCD-1) respectively with fusion of SoCC and ZbF (CZ features) along with polynomial SVM. It is evident from these tables that all the fusion features except PZ further improves the recognition

Table 6.2: Performance of prediction models on HGCD-1 with SoCC, PPbF and ZbF features

Representation	Prediction Models					
	Linear SVM			Polynomial SVM		
	Accuracy (%)	F-measure (%)	BVoDP	Accuracy (%)	F-measure (%)	BVoDP
SoCC	98.98	98.98	c=0.1	99.47	99.47	c=1, d=2
PPbF	85.25	85.23	c=0.01	91.27	91.27	c=1, d=2
ZbF	86.69	86.68	c=0.1	92.78	92.78	c=10, d=2

Representation	Prediction Models		
	Naive Bayes		
	Accuracy (%)	F-measure (%)	BVoDP
SoCC	90.76	89.88	-
PPbF	76.03	75.97	-
ZbF	69.82	69.9	-

Table 6.3: Accuracy(%) values obtained by prediction models with SoCC on individual numerals (on HGND)

Numeral Class	Naive Bayes	ANN	Linear SVM	Polynomial SVM
1	98.75	99.17	98.75	100
2	92.5	95.42	98.75	99.58
3	97.08	97.5	95.83	99.58
4	91.25	91.25	97.08	98.75
5	99.58	98.75	99.58	100
6	99.58	98.75	99.58	100
7	81.25	92.08	96.25	97.5
8	91.67	95	98.75	97.92
9	100	100	100	100
10	95.42	95.83	99.58	99.17

Table 6.4: Accuracy(%) values obtained by prediction models with ZbF on individual numerals (on HGND)

Numeral Class	Naive Bayes	ANN	Linear SVM	Polynomial SVM
1	95.36	97.86	95.71	98.93
2	88.57	95.36	95.71	98.93
3	81.07	95.71	95	96.07
4	82.86	93.21	94.29	96.07
5	93.93	92.5	95	97.5
6	93.93	96.43	95.71	98.57
7	64.29	90.71	90.36	93.57
8	88.57	88.57	92.14	93.93
9	93.57	99.29	98.21	100
10	86.43	93.93	93.57	97.14

Table 6.5: Accuracy(%) values obtained by prediction models with PPbF on individual numerals (on HGND)

Numeral Class	Naive Bayes	ANN	Linear SVM	Polynomial SVM
1	95	95.42	96.67	95.42
2	90.42	92.08	91.25	95
3	87.08	92.08	93.33	95.42
4	85.83	94.17	93.33	95.83
5	92.08	95	92.5	93.75
6	92.08	95	92.5	93.75
7	77.08	88.75	88.75	90.42
8	94.17	93.33	93.33	93.75
9	94.58	97.92	97.5	98.33
10	94.58	92.92	95	97.08

Table 6.6: Accuracy(%) values obtained by prediction models with SoCC on individual characters (on HGCD-1)

Character Class	Naive Bayes	Linear SVM	Polynomial SVM
1	96	99.5	99.75
2	95.25	100	100
3	100	100	100
4	99.5	100	100
5	92.75	97.75	98.5
6	96.25	100	100
7	68	95.75	97.75
8	90.25	100	99.75
9	72.25	96.5	98.5
10	98	99.5	100
11	98.5	99.5	99.5
12	85.75	97	99
13	98.5	99.25	99.5
14	4.25	99	99.25
15	99.25	99.5	99.5
16	98.25	100	100
17	96.5	100	100
18	98.75	99.25	99.75
19	78.5	97	98.75
20	98.75	98.5	100
21	100	99.75	99.75
22	86	98.25	99.5
23	93.25	99.5	99.75
24	94.75	98.75	99
25	100	99.75	99.75
26	81.75	99.75	99.75
27	99.75	99.75	100
28	100	99.75	100
29	99	100	100
30	94	99.25	100
31	99.25	99	99.5
32	93.25	98.75	98.25
33	73.25	97.75	99
34	100	98.5	99.25
35	94.25	100	99.25
36	98.5	99.5	100
37	95	99.75	99.75
38	84.25	99.5	100
39	100	97.75	98.25
40	83.75	95.75	99
41	77.75	98.75	99.25
42	81	98.5	98.5
43	99.75	99.5	100
44	100	99.75	100

Table 6.7: Accuracy(%) values obtained by prediction models with ZbF on individual characters (on HGCD-1)

Character Class	Naive Bayes	Linear SVM	Polynomial SVM
1	61	88	93.5
2	68.25	82.5	91.25
3	79.75	90	93.5
4	90	97.25	98
5	41.25	71	82.75
6	58.5	81.75	90.5
7	71.75	91	95
8	89	96	98.25
9	68.75	91.75	95.25
10	65.25	86	91.5
11	83.25	94.5	96.5
12	78.75	88.5	95.25
13	76	91.25	94.75
14	82.5	91	97
15	80.25	85.75	93
16	65.75	90	96
17	54.75	72.25	88.75
18	81.5	92.25	94.75
19	64.25	82	87.5
20	58.25	82.75	91.75
21	66.25	84.75	87.75
22	54.25	84.25	94.5
23	76.25	85	92.75
24	63.75	87.75	94
25	54.25	78.25	86.5
26	43.25	77	86.5
27	49.25	85.75	93.5
28	98.25	99.75	100
29	89.5	99.25	100
30	73	85.75	95.5
31	63	88.5	96
32	66.75	82	89.5
33	58	79.25	86
34	86.75	91.25	97
35	46.5	77.5	90
36	84.25	97.25	99
37	88	97.25	98.5
38	59.75	82.5	90.25
39	76.25	77.25	81.5
40	55.5	76	82
41	70.75	85.75	93.75
42	68.25	85.25	95
43	79.25	92	96.75
44	82.5	90	91.5

Table 6.8: Accuracy(%) values obtained by prediction models with PPbF on individual characters (on HGCD-1)

Character Class	Naive Bayes	Linear SVM	Polynomial SVM
1	72.25	84.5	89.75
2	69	79.75	90.75
3	72.5	92	94.5
4	90.25	97.25	98
5	55	71.25	79
6	54	82	90.25
7	85.25	87.75	91.5
8	93.25	92.5	97
9	72.5	86.5	93.5
10	69.75	85.5	89.25
11	83.25	92.75	96.5
12	87	82.5	89.25
13	71	90.75	92
14	77.75	90.75	92.25
15	79.25	86.75	92.75
16	80	89	94.25
17	60	79.25	84.75
18	84	88.25	96
19	76.25	76.5	85.5
20	69.75	83.5	88.5
21	83	85	91.25
22	61.25	81.25	87.5
23	69.25	89	92.5
24	74.75	83.75	91.75
25	57	74.5	85
26	48.75	74.75	82.75
27	59	81	94
28	98.5	99.75	100
29	95.25	97	100
30	78.5	84.75	92.25
31	78.25	89.75	94
32	79.75	81.25	86
33	73.75	77.75	89.25
34	94	92	96.75
35	68	68.75	85.5
36	96.5	98.25	97.5
37	90.25	98	98
38	65.5	77	88.75
39	80	76	82.75
40	53	75	81.75
41	82.5	81.5	92.25
42	81.75	84.75	93.25
43	87.25	89.75	96.25
44	87.75	91.75	91.75

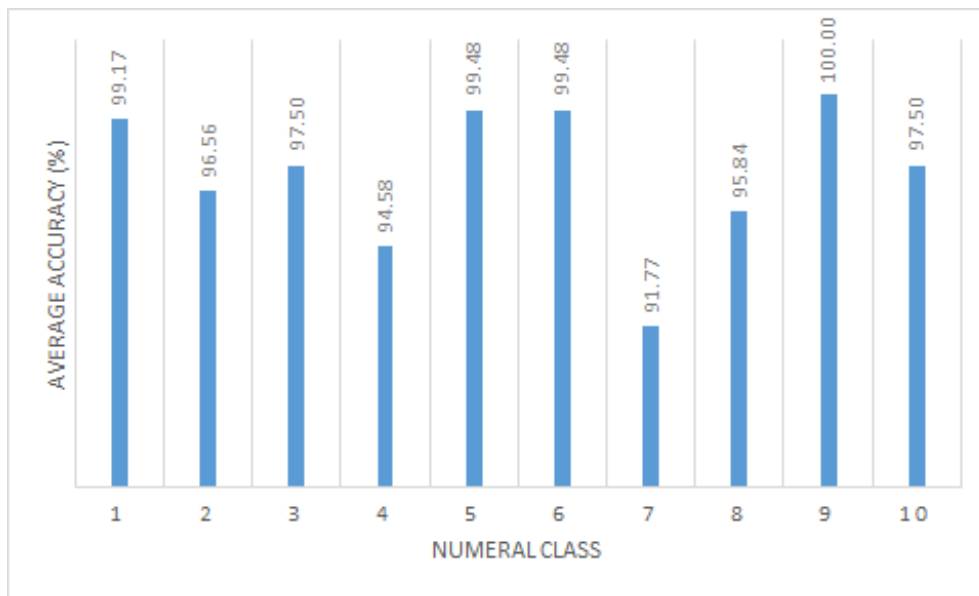


Figure 6.4: Accuracy values averaged over prediction models learnt using SoCC features for individual numerals (on HGND)



Figure 6.5: Accuracy values averaged over prediction models learnt using ZbF for individual numerals (on HGND)

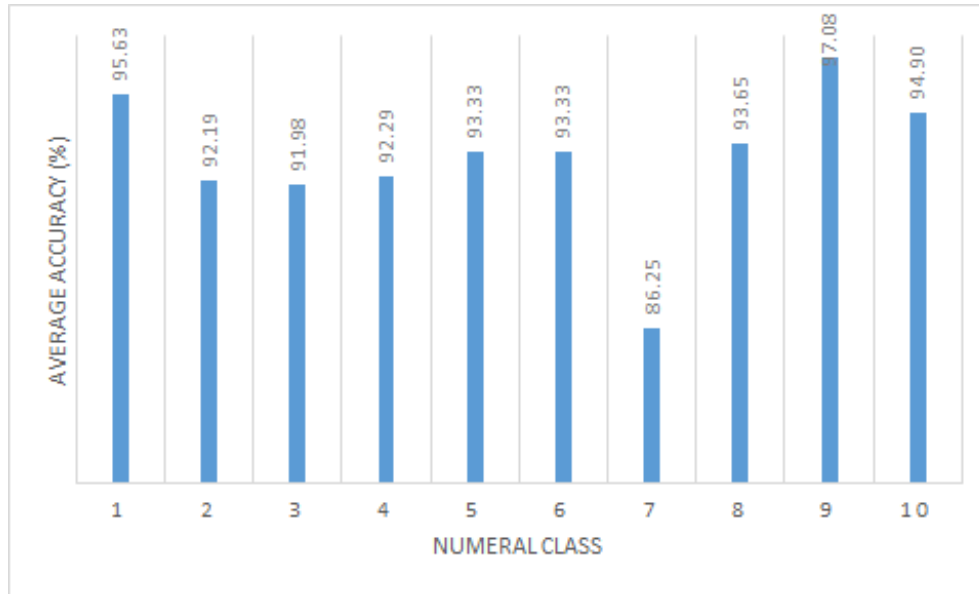


Figure 6.6: Accuracy values averaged over prediction models learnt using PPbF for individual numerals (on HGND)

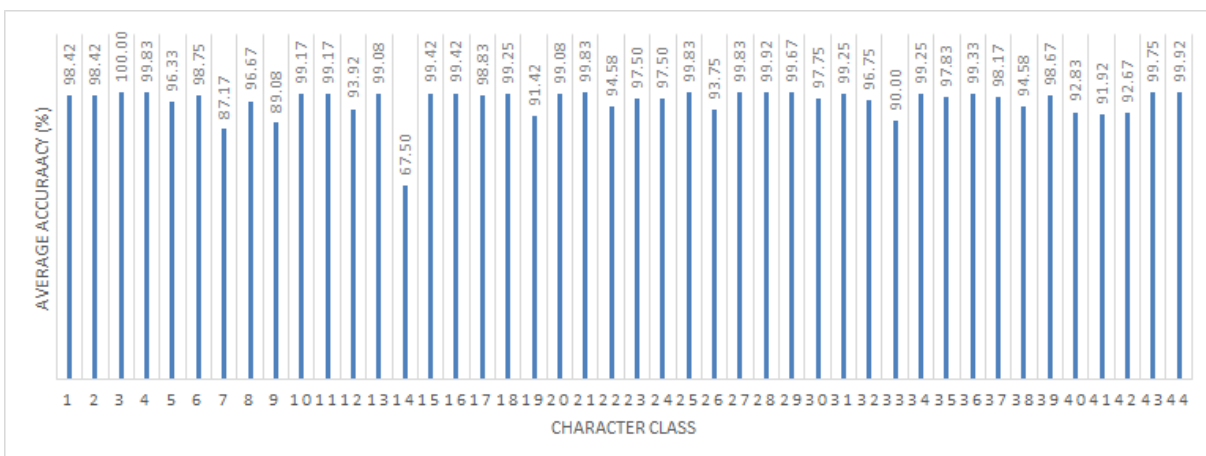


Figure 6.7: Accuracy values averaged over prediction models learnt using SoCC features for individual characters (on HGCD-1)



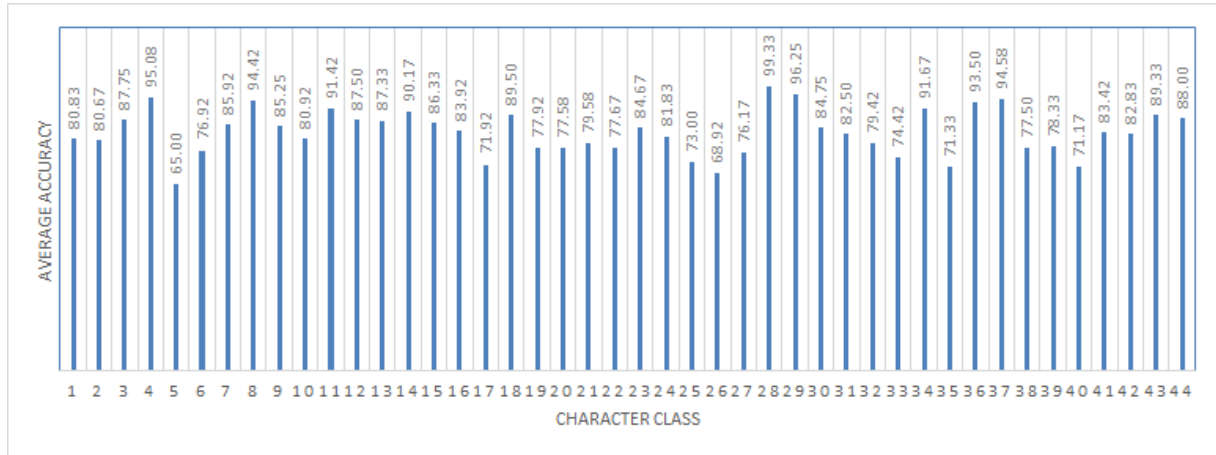


Figure 6.8: Accuracy values averaged over prediction models learnt using ZbF for individual characters (on HGCD-1)

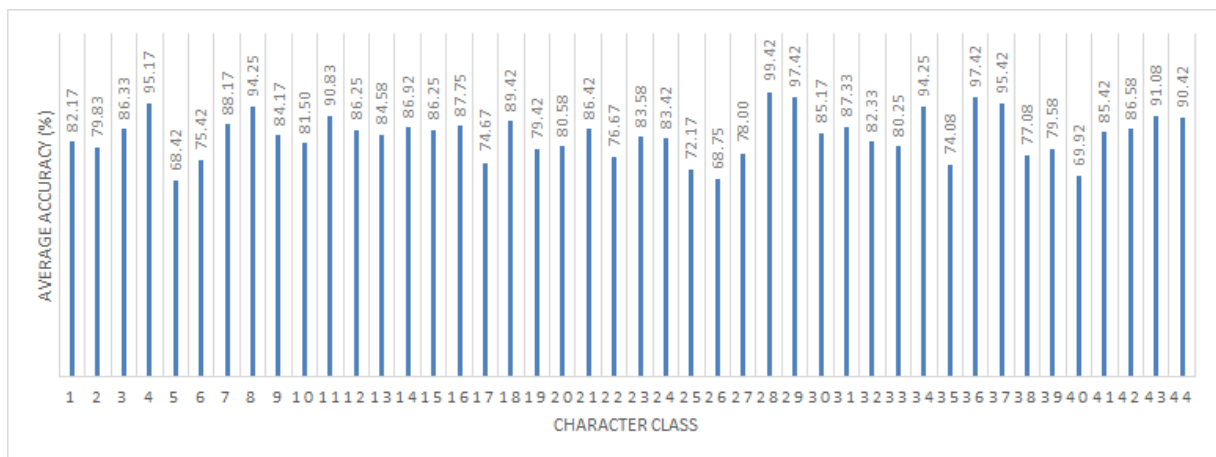


Figure 6.9: Accuracy values averaged over prediction models learnt using PPbF for individual characters (on HGCD-1)

accuracies of all predictions models when compared to their individual constituent counterparts.

Table 6.9: Performance of prediction models on HGND with individual features and fusion features

Representation	Prediction Models					
	Linear SVM			Polynomial SVM		
	Accuracy (%)	F-measure (%)	BVoDP	Accuracy (%)	F-measure (%)	BVoDP
SoCC	98.41	98.39	c=0.01	99.25	99.22	c=0.01, d=4
PPbF	93.41	93.51	c=0.01	94.87	94.80	c=1, d=2
ZbF	94.57	94.58	c=0.1	97.07	97.06	c=10, d=2
CP	99.67	99.67	c=0.01	99.85	99.85	c=0.01, d=3
CZ	99.46	99.46	c=0.01	99.92	99.92	c=0.1, d=3
PZ	94.39	94.38	c=0.01	96.50	96.50	c=1, d=3
CPZ	99.67	99.67	c=0.1	99.50	99.50	c=1, d=2

Representation	Prediction Models					
	Naive Bayes			ANN		
	Accuracy (%)	F-measure (%)	BVoDP	Accuracy (%)	F-measure (%)	BVoDP
SoCC	94.70	94.62	-	96.37	96.28	n=20, ep=8000
PPbF	90.29	90.27	-	93.66	93.51	n=90, ep=3000
ZbF	86.85	86.31	-	94.35	94.34	n=80, ep=5000
CP	97.57	97.58	-	97.85	97.84	n=30, ep=1000
CZ	97.50	97.52	-	97.67	97.66	n=30, ep=8000
PZ	89.14	89.10	-	93.78	93.77	n=90, ep=7000
CPZ	98.17	98.18	-	97.17	97.15	n=50, ep=8000

Results achieved with fusion features are further summarized and depicted in Figures 6.10 and 6.11. Figure 6.10 demonstrates the results on HGND while Figure 6.11 depicts the results on HGCD-1. These figures report accuracy values averaged over prediction models as was the case with Figures 6.2 and 6.3 for individual features. It is apparent from these figures that irrespective of the prediction models, all fusion features except PZ have a definite edge over their individual constituent counterparts. This is again noteworthy as it establishes these fusion features as robust features.

Table 6.10: Performance of prediction models on HGCD-1 with individual features and fusion features

Representation	Prediction Models					
	Linear SVM			Polynomial SVM		
	Accuracy (%)	F-measure (%)	BVoDP	Accuracy (%)	F-measure (%)	BVoDP
SoCC	98.98	98.98	c=0.1	99.47	99.47	c=1, d=2
PPbF	85.25	85.23	c=0.01	91.27	91.27	c=1, d=2
ZbF	86.69	86.68	c=0.1	92.78	92.78	c=10, d=2
CP	99.49	99.49	c=0.1	99.73	99.73	c=0.01, d=3
CZ	99.56	99.56	c=0.1	99.8	99.8	c=1, d=2
PZ	87.44	87.43	c=0.01	92.22	92.23	c=1, d=2
CPZ	99.63	99.63	c=0.01	99.73	99.73	c=1, d=2

Representation	Prediction Models		
	Naïve Bayes		
	Accuracy (%)	F-measure (%)	BVoDP
SoCC	90.76	89.88	-
PPbF	76.03	75.97	-
ZbF	69.82	69.9	-
CP	96.8	96.75	-
CZ	96.5	96.47	-
PZ	76.64	76.7	-
CPZ	96.43	96.43	-

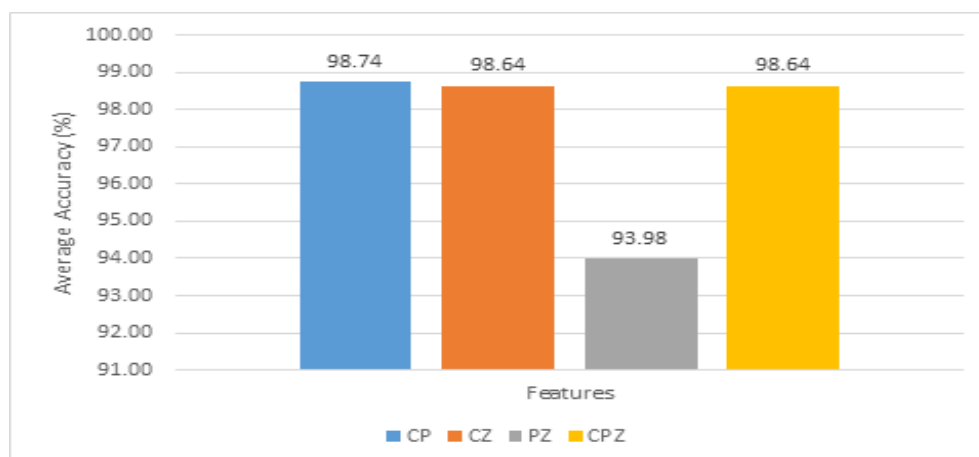


Figure 6.10: Accuracy values averaged over prediction models learnt using fusion features (on HGND)

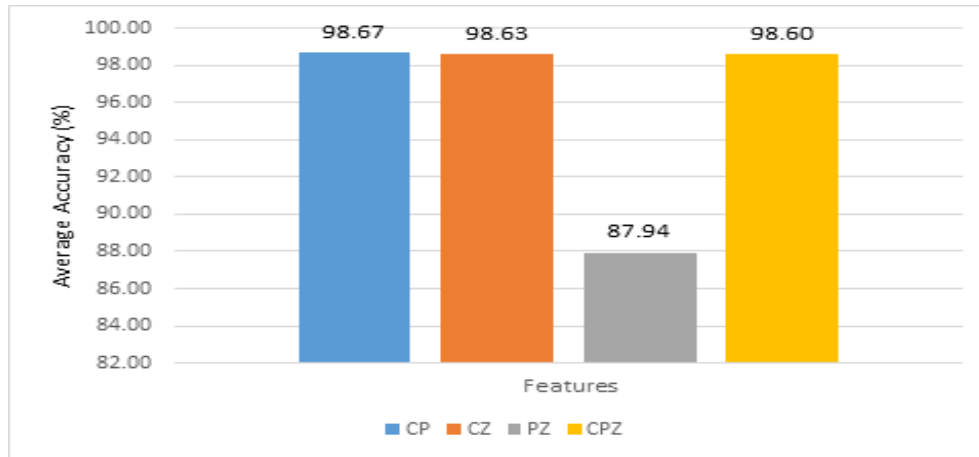


Figure 6.11: Accuracy values averaged over prediction models learnt using fusion features (on HGCD-1)

Performance of prediction models with CP, CZ, PZ and CPZ features on individual numerals is presented in Tables 6.11, 6.12, 6.13 and 6.14, and in the same way for individual characters is presented in Tables 6.15, 6.16, 6.17 and 6.18. Accuracy values averaged over prediction models learnt using fusion features for each numeral and character is depicted in Figures 6.12, 6.13 6.14 and 6.15 for numerals and in Figures 6.16, 6.17 6.18 and 6.19 for characters.

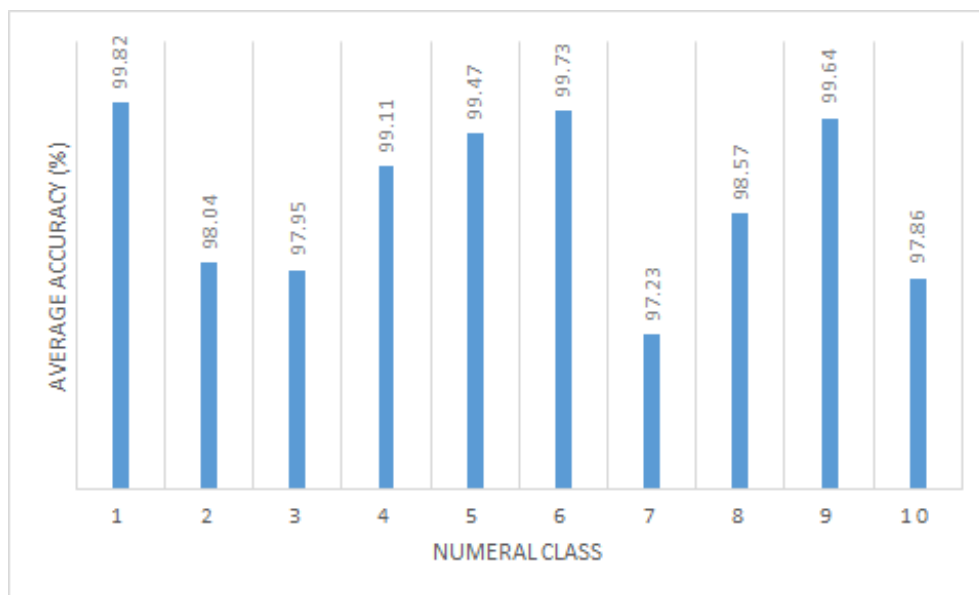


Figure 6.12: Accuracy values averaged over prediction models learnt using PC features for individual numerals (on HGND)

Table 6.11: Accuracy(%) values obtained by prediction models with CP on individual numerals (on HGND)

Numeral Class	Naive Bayes	ANN	Linear SVM	Polynomial SVM
1	99.29	100	100	100
2	93.93	98.93	99.64	99.64
3	95	98.21	99.29	99.29
4	96.79	99.64	100	100
5	99.29	98.57	100	100
6	99.64	99.64	99.64	100
7	98.21	90.71	100	100
8	98.57	96.07	99.64	100
9	99.29	99.64	99.64	100
10	95.71	97.14	98.93	99.64

Table 6.12: Accuracy(%) values obtained by prediction models with CZ on individual numerals (on HGND)

Numeral Class	Naive Bayes	ANN	Linear SVM	Polynomial SVM
1	100	100	100	100
2	96.43	96.43	98.21	99.64
3	98.21	97.14	98.93	100
4	95.71	99.29	100	100
5	97.5	99.29	100	100
6	97.5	99.29	100	100
7	97.14	92.14	100	100
8	97.5	96.79	98.21	100
9	98.57	99.64	100	100
10	96.43	96.79	99.29	99.64

Table 6.13: Accuracy(%) values obtained by prediction models with PZ on individual numerals (on HGND)

Numeral Class	Naive Bayes	ANN	Linear SVM	Polynomial SVM
1	93.21	97.86	98.57	97.86
2	91.07	93.93	95.36	95.71
3	85.36	95.71	95	96.07
4	87.5	92.86	94.64	95.36
5	94.64	93.93	95.36	97.5
6	94.29	94.29	94.64	98.57
7	72.14	87.5	87.5	94.29
8	87.14	93.57	92.5	95
9	92.5	97.14	98.21	98.93
10	93.57	91.07	92.14	95.71

Table 6.14: Accuracy(%) values obtained by prediction models with CPZ on individual numerals (on HGND)

Numeral Class	Naive Bayes	ANN	Linear SVM	Polynomial SVM
1	100	99.64	100	100
2	97.5	96.43	99.64	98.21
3	98.57	97.5	99.64	99.29
4	96.07	99.64	100	100
5	99.29	98.21	99.64	99.64
6	98.21	98.21	100	99.64
7	97.14	87.86	100	98.93
8	97.5	95.36	98.93	99.64
9	98.93	100	99.64	100
10	98.57	98.93	99.29	99.64

Table 6.15: Accuracy(%) values obtained by prediction models with CP on individual characters (on HGCD-1)

Character Class	Naive Bayes	Linear SVM	Polynomial SVM
1	97.75	99.5	100
2	98.75	100	99.5
3	99.5	100	100
4	99.5	100	100
5	98	98.75	99.75
6	93.75	99.75	99.75
7	87.25	99	99.5
8	97	99.75	100
9	94.25	99.75	99
10	99.5	99	99.75
11	98.5	99.5	100
12	98.75	98.25	99.75
13	97.5	99.5	100
14	67.25	98.75	99.5
15	97.25	100	100
16	98.75	100	100
17	98.75	100	100
18	98.25	100	99.75
19	97.25	99.5	99.25
20	97.5	99.5	100
21	99.5	100	100
22	97.75	100	99.75
23	96.5	99.5	100
24	97.25	99.5	100
25	99.5	100	100
26	97.5	99.75	100
27	97.25	100	100
28	99.5	100	100
29	99.25	100	100
30	98.75	99.25	99.25
31	98.5	98.75	99.75
32	98.5	100	100
33	97.25	99	99.75
34	99.25	99.75	99.75
35	98	99.75	99.25
36	98.75	99.5	100
37	97.5	99.5	99.75
38	96.75	99.5	99.75
39	99	96.25	99.75
40	87.5	97.5	97
41	97.5	100	99.5
42	94	99.75	99.75
43	99.75	100	100
44	99.25	100	100

Table 6.16: Accuracy(%) values obtained by prediction models with CZ on individual characters (on HGCD-1)

Character Class	Naive Bayes	Linear SVM	Polynomial SVM
1	99.25	100	99.75
2	96.5	99.5	100
3	100	100	100
4	99.75	99.75	99.75
5	96.75	99	99
6	95.75	99	99.5
7	88.25	98.5	100
8	97.25	100	100
9	86.25	99	99.5
10	99.25	99.75	99.75
11	99	100	100
12	98.25	99.5	99.5
13	98	100	100
14	70.75	99	99.5
15	98	100	100
16	99.25	99.75	100
17	100	100	100
18	97	99.25	100
19	96.5	99	100
20	98	100	100
21	99	100	100
22	98.5	99.25	100
23	96.75	99.5	99.75
24	94.75	99.75	100
25	98.25	99.75	100
26	98.5	99	100
27	99	99.75	100
28	99.75	100	100
29	96.5	100	100
30	96	99.5	99.75
31	99.5	99	99.25
32	98.5	100	99.75
33	95.25	99.75	100
34	98.5	100	99.75
35	98.25	99.75	100
36	98.75	99.75	100
37	98.5	100	99.5
38	96	100	99.75
39	98.25	99	98.75
40	86	96.5	99
41	97	99.5	100
42	91.75	100	100
43	100	100	100
44	99.25	100	100



Table 6.17: Accuracy(%) values obtained by prediction models with PZ on individual characters (on HGCD-1)

Character Class	Naive Bayes	Linear SVM	Polynomial SVM
1	72.75	88	90.25
2	76.75	85.5	93
3	80.75	92.5	95.25
4	92.5	97.5	98
5	51.5	74.25	80
6	62.75	84.25	93.5
7	82.25	89.75	92
8	90.5	94	96.75
9	77.25	90.25	92.75
10	69	84.5	89.75
11	88.25	93.25	97.25
12	85.25	87	93.25
13	78.25	93	92.75
14	81	91.75	93.5
15	82.75	88.25	92.75
16	78.25	90.5	95.5
17	56.25	80.25	87
18	87.5	90	96.25
19	75.75	82.25	86.5
20	72.5	84.75	88.75
21	79.75	86.25	91.5
22	58	85.25	88
23	72	89.25	94.5
24	71.5	85.5	94
25	61.5	78	87
26	49	78.5	85.5
27	63	83.5	93.75
28	98	100	100
29	95	97.25	100
30	77.25	86.75	93.5
31	76	90.5	95.5
32	78	83.5	88.5
33	79.5	81.75	90
34	91.25	93	97
35	56.75	74.75	87.75
36	92.75	98.25	98.25
37	89.5	98	97.5
38	70.25	82.75	91
39	76.5	78	82
40	60	78	83
41	80.75	85.25	93.5
42	78.25	88.5	93.5
43	89.25	92.25	96
44	87	91.25	92

Table 6.18: Accuracy(%) values obtained by prediction models with CPZ on individual characters (on HGCD-1)

Character Class	Naive Bayes	Linear SVM	Polynomial SVM
1	98.5	99.75	99.75
2	97.75	100	99.75
3	100	100	100
4	98.5	100	100
5	96.5	99	99
6	93	99.5	99.25
7	94	99.5	99.5
8	97.25	100	100
9	91.75	99	99
10	97.75	100	99.75
11	97.25	100	100
12	97.5	100	100
13	94.5	100	100
14	79.75	98.75	99.5
15	96.25	100	100
16	97.75	100	100
17	98.5	100	100
18	95.75	100	100
19	97.75	100	100
20	98.5	98.75	100
21	98.75	100	100
22	97.25	100	100
23	94.5	100	100
24	96	100	99.75
25	97.75	100	100
26	96.5	99.75	100
27	98.5	100	100
28	99.5	100	100
29	99	100	100
30	94.25	98.75	99.75
31	98.25	99.75	99.25
32	97	99.75	99
33	98.75	100	99.75
34	97.5	99.75	99.75
35	97.25	99.5	100
36	98.5	99.75	99.75
37	97	99.75	99.5
38	97.5	99	99.5
39	94.5	98.25	99.25
40	86.5	96	97.75
41	96.5	99.75	100
42	95.25	99.75	100
43	100	100	99.75
44	98	100	100

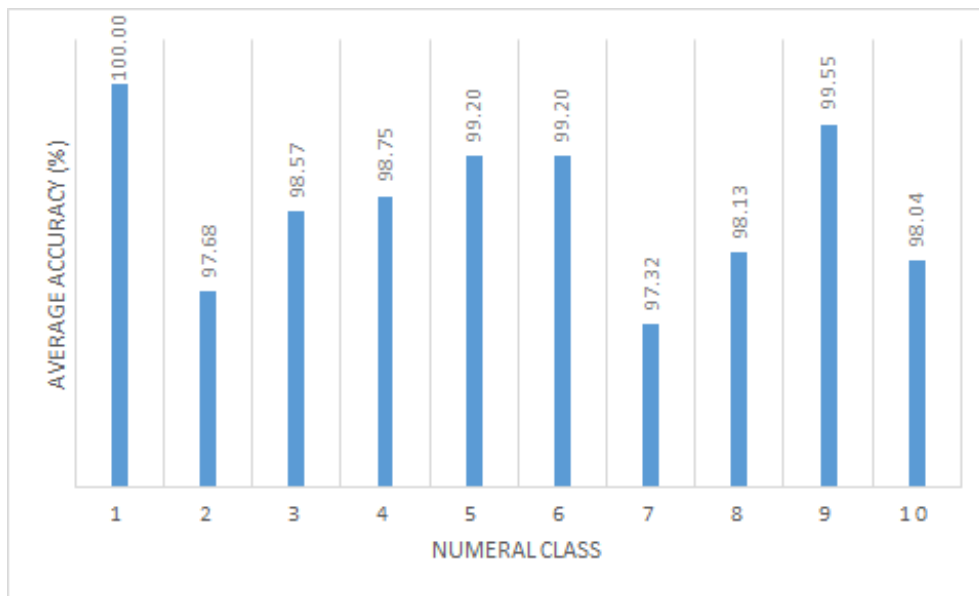


Figure 6.13: Accuracy values averaged over prediction models learnt using ZC features for individual numerals (on HGND)

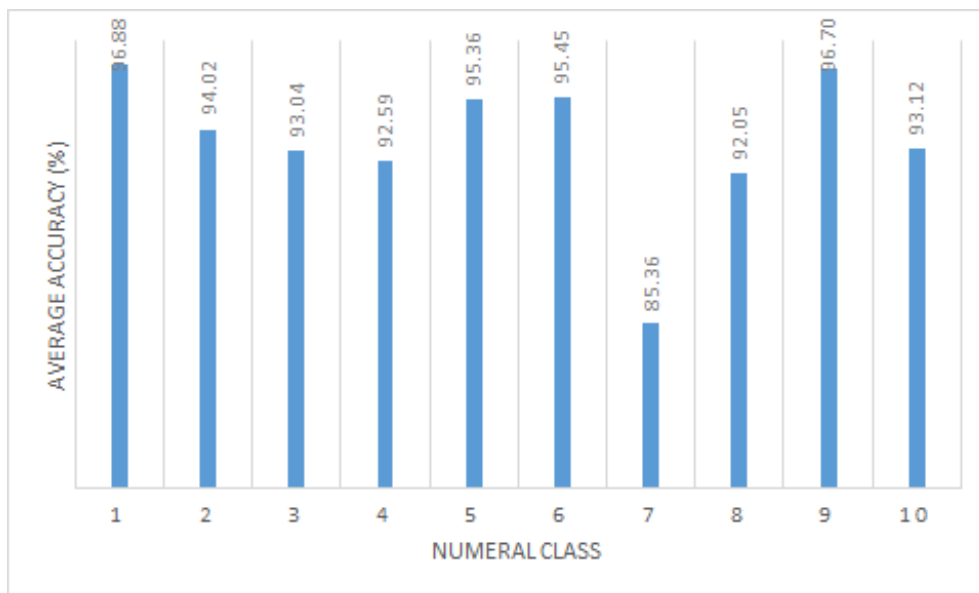


Figure 6.14: Accuracy values averaged over prediction models learnt using PZ features for individual numerals (on HGND)

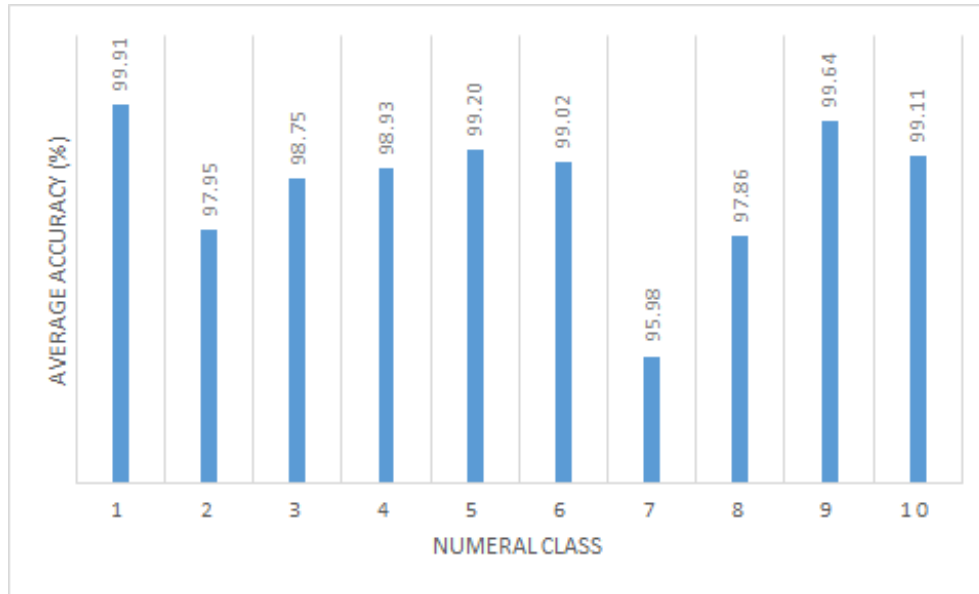


Figure 6.15: Accuracy values averaged over prediction models learnt using ZPC features for individual numerals (on HGND)

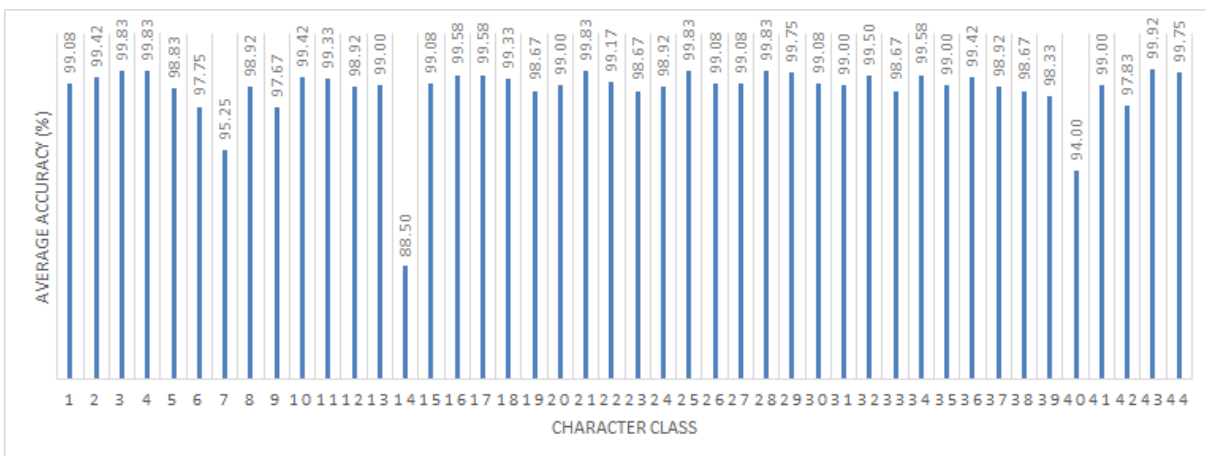


Figure 6.16: Accuracy values averaged over prediction models learnt using PC features for individual characters (on HGCD-1)

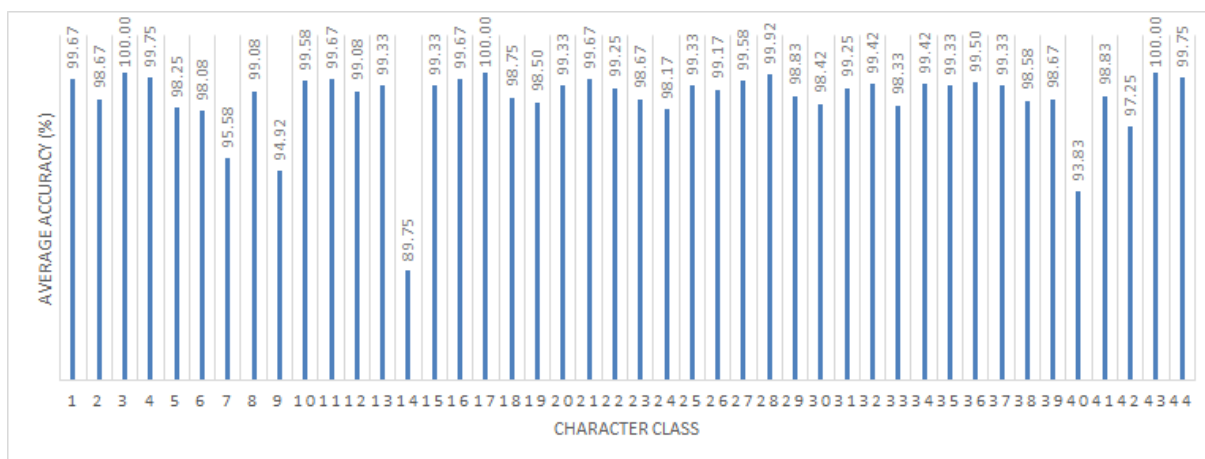


Figure 6.17: Accuracy values averaged over prediction models learnt using ZC features for individual characters (on HGCD-1)

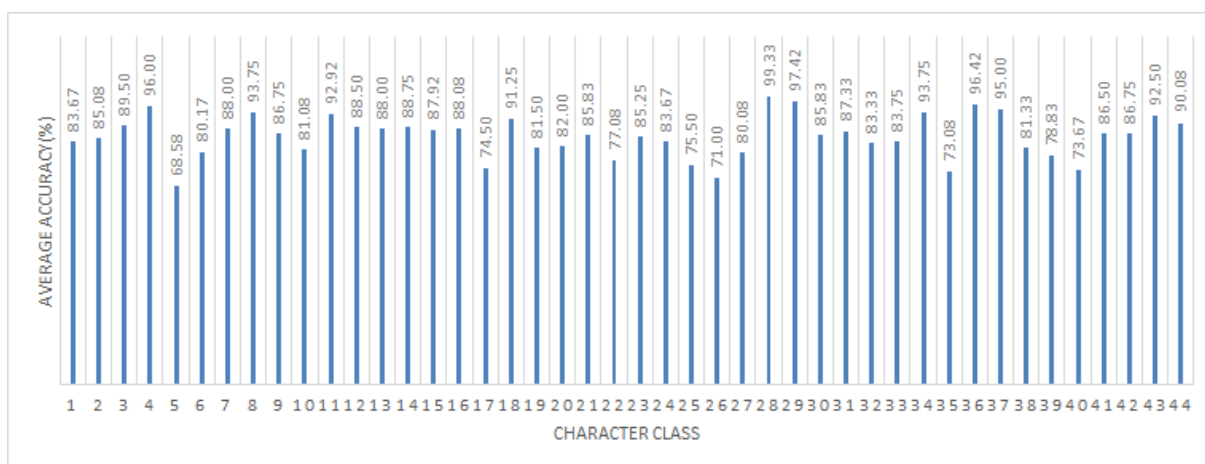


Figure 6.18: Accuracy values averaged over prediction models learnt using PZ features for individual characters (on HGCD-1)

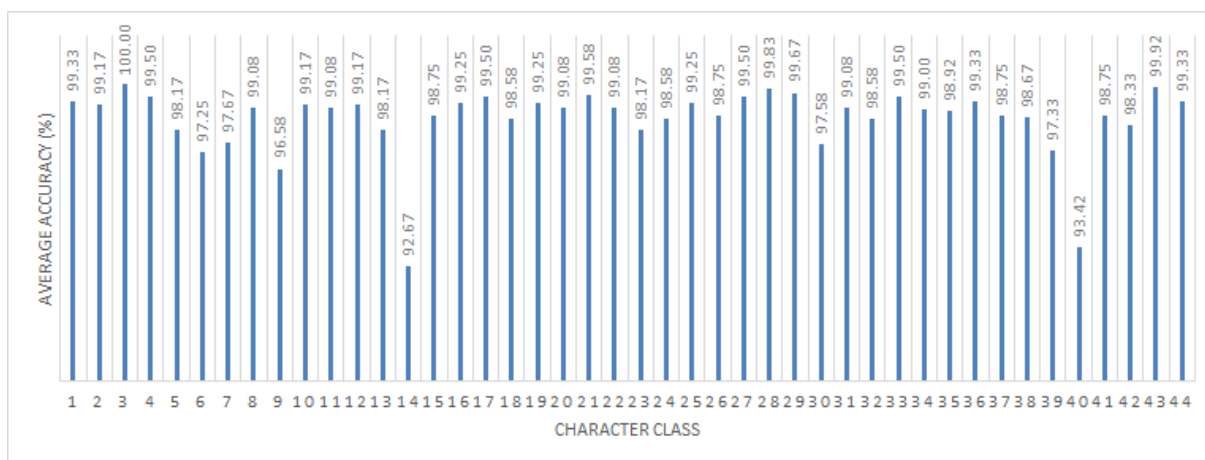


Figure 6.19: Accuracy values averaged over prediction models learnt using ZPC features for individual characters (on HGCD-1)

### 6.1.3 Performance Evaluation using SD, ZPM and NCC

For experimentation purpose datasets consist of 20,500 handwritten Gujarati characters (HGCD-2) was used. This dataset accommodates 41 distinct Gujarati characters as shown in Figure 3.3. Uniform distribution of 500 samples per class was provided for character dataset.

Table 6.19 shows performance of linear SVM, polynomial SVM and naive Bayes on handwritten Gujarati character dataset by using structure decomposition based approach (SD), normalised cross correlation based approach (NCC) and zone pattern matching based approach (ZPM).

Table 6.19: Performance of prediction models on HGCD-2 with SD, ZPM, NCC features

Representation	Prediction Models					
	Linear SVM			Polynomial SVM		
	Accuracy (%)	F-measure (%)	BVoDP	Accuracy (%)	F-measure (%)	BVoDP
SD	99.29	99.29	c=0.01	99.48	99.48	c=1, d=2
ZPM	84.63	84.65	c=0.01	89.02	89.05	c=0.1, d=2
NCC	68.53	68.44	c=10	66.43	66.20	c=100, d=2

Representation	Prediction Models		
	Naive Bayes		
	Accuracy (%)	F-measure (%)	BVoDP
SD	97.53	97.52	-
ZPM	65.87	65.06	-
NCC	53.12	52.11	-

These accuracy values are further averaged over prediction models and are depicted in Figure 6.20.

As per the Figure 6.20, it is clear that SD provides highest accuracy compared to other feature sets. It is evident from this figure that irrespective of the prediction models, SD has a definite edge over other features. This is remarkable as it establishes SD as a robust feature.

Accuracy values averaged over prediction models learnt using SD, ZPM and NCC features for individual characters are represented in Figures 6.21, 6.22 and 6.23.

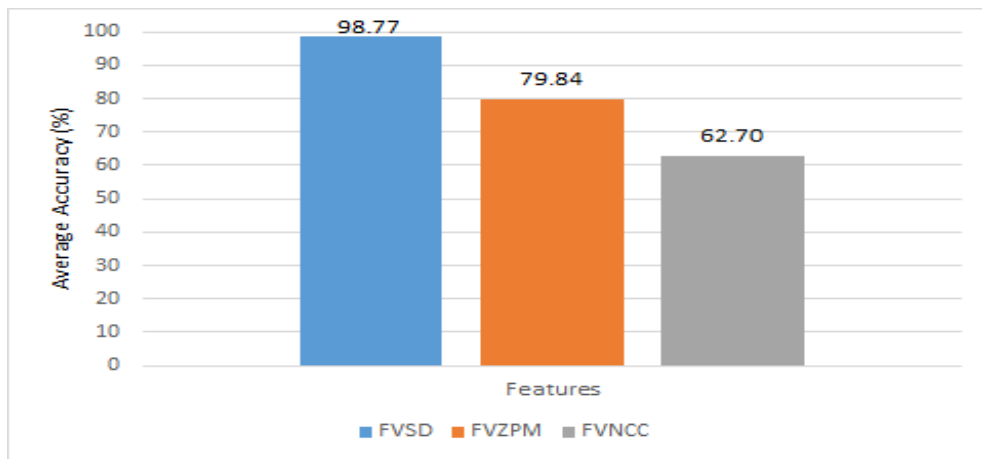


Figure 6.20: Accuracy values averaged over prediction models learnt using SD, ZPM and NCC features (on HGCD-2)

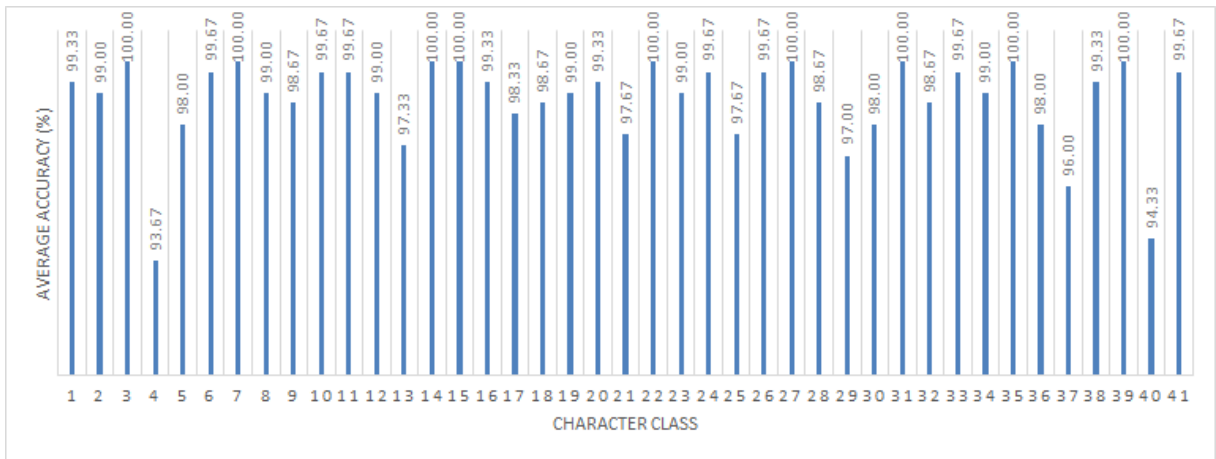


Figure 6.21: Accuracy values averaged over prediction models learnt using SD features for individual characters (on HGCD-2)

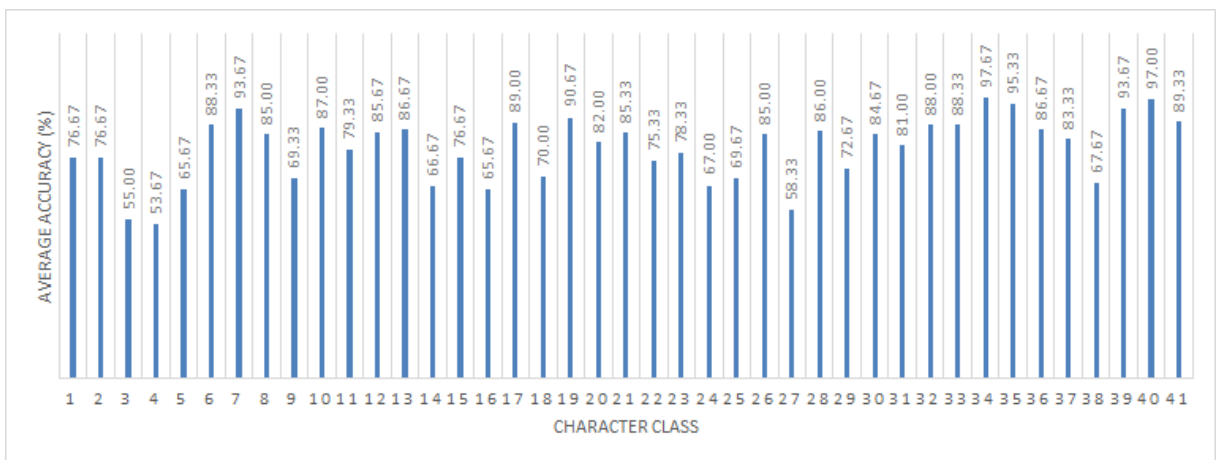


Figure 6.22: Accuracy values averaged over prediction models learnt using ZPM features for individual characters (on HGCD-2)

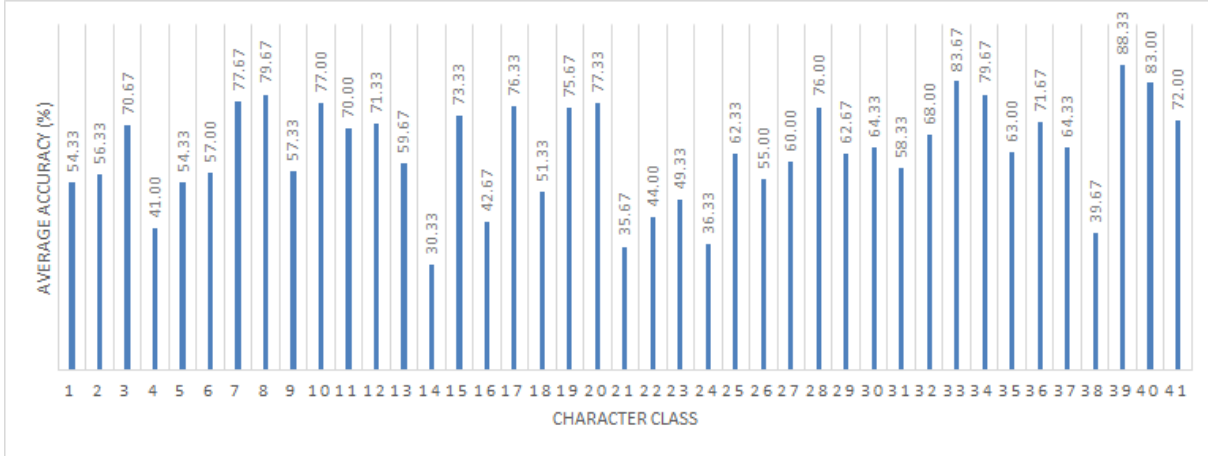


Figure 6.23: Accuracy values averaged over prediction models learnt using NCC features for individual characters (on HGCD-2)

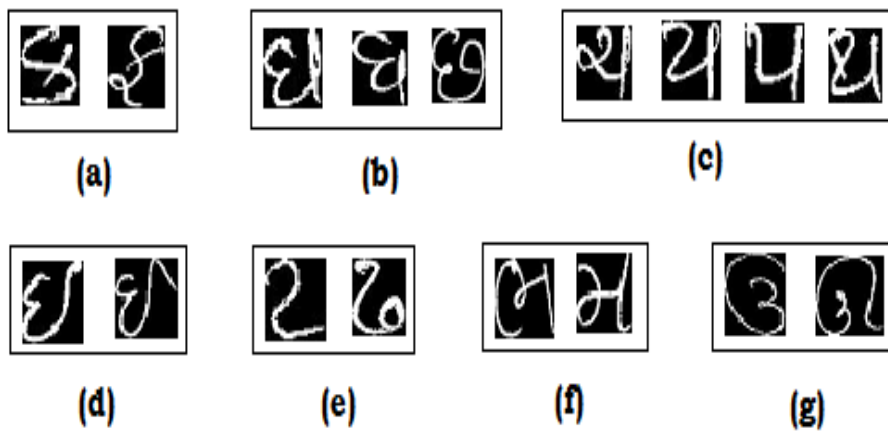


Figure 6.24: Sets of similar looking Gujarati characters



## 6.2 Summary

It can be realized from the results achieved that all prediction models performed better when these were learnt through SoCC compared to other feature sets comprising of only individual features. Highest accuracy values of 99.25% and 99.47% were achieved using SoCC with polynomial SVM for numeral dataset and character dataset respectively. In case of fusion features, fusion of SoCC and ZbF (CZ features) along with polynomial SVM provides the best results. Proposed feature SD provides highest accuracy of 99.48% for handwritten characters. In Gujarati, many characters have similar shapes. This provides a great challenge in recognition of handwritten characters. On the basis of analysis of performance achieved for individual handwritten Gujarati characters it can be concluded that most of the errors are due to confusion among similar looking character patterns. Several such pairs exist in Gujarati language. Some of them are displayed in Figure 6.24.

Table 6.20 shows few cases of misclassification between the predicted class and actual class for character images (Class for the characters are considered as per Figure 3.2). It can be observed that the images shown in Table 6.20 are even difficult for human being to classify. Major reasons behind misclassification are the similar looking characters in Gujarati script and significant variations in writing style of different persons.

Table 6.20: Misclassified character images

Input character images from testing dataset	Predicted Class	Actual Class
	5	7
	7	5
	30	35
	39	40
	40	39



# Chapter 7

## Conclusions and Future Work

This chapter summarizes the thesis and provides discussions on possible extensions for future work. Section 7.1 concludes the thesis and possible future extensions are discussed in Section 7.2.

### 7.1 Conclusions

In this thesis various methodologies are proposed and utilized for offline handwritten Gujarati character recognition. Some novel efficient algorithms are also proposed for the recognition of handwritten Gujarati characters. The peculiarities and challenges of handwritten Gujarati character recognition are discussed. A detailed literature review in the field of offline Gujarati and Devanagari character recognition is described on the basis of different feature extraction and classification methods. A benchmark dataset of handwritten Gujarati characters and numerals is developed for experimentation purpose. This work is the first exclusive work on such a large and representative dataset of handwritten characters in the Gujarati script. All the set of symbols/characters of Gujarati script are included for dataset collection.

Zone based, projection profiles based and chain code based features are employed as individual features. Fusion of these features is also proposed for learning prediction models. Even though each feature extraction method itself is quite efficient, there is more to be improved in classification accuracy which is achieved by features fusion based approach. Moreover, novel feature extraction methods based on structural decomposition, zone pattern matching and normalized cross correlation are proposed.

Performance evaluation is provided with classifiers such as Artificial Neural Net-

work (ANN), Support Vector Machine (SVM) and Naive Bayes (NB) classifier. Classifier design is based on extensive experimentation for fine-tuning several parameters that influence the performance. The created benchmark dataset of handwritten numerals and characters is used to evaluate the performance of these methodologies. The success of proposed methodologies is evident from the experimental results. Owing to this success, the proposed work can be considered as the noteworthy contribution to the research.

## 7.2 Future Scope

One important direction for the future is to extend the work by incorporating conjuncts and modifiers in handwritten Gujarati character dataset. A complete HCR system can be developed by including all the basic as well as frequently used composite characters from the Gujarati alphabet. In this regard, it will be required to develop suitable classifiers that can be customized to handle such a large set of characters. Increase in a number of classes can make the problem difficult and usefulness of deep learning may emerge as the potential solution.

This work opens up many interesting problems in this domain. One important direction for the future work is to extend the work beyond handwritten characters by incorporating recognition of complete words or sentences. Proposed approach can be implemented for the recognition of other Indian scripts that share similar structures and features. The proposed methodologies can also be utilized for identification of handwritten characters of other languages. Recognition of degraded handwritten characters is another area need to explore and it is needed to develop efficient algorithms for this purpose.

Efforts are needed in the domain of camera captured document recognition. Furthermore, a system can be developed that could recognize the text from the inscription inscribed on the surface of the rock or wall. To do so, it will be required to analyse the photographic images of the inscriptions, instead of analysing the scanned images of the manuscripts. Proposed approach can be implemented in different applications like automatic processing of postal codes and addresses, bank cheques, tax and admission forms, reservation forms. The benchmark dataset which is generated through proposed work can be utilized by research community for their experimen-

tation work. Future of this research could move forward with the recognition of handwritten documents leading to the ultimate goal of machine simulation of human reading.



# Works Cited

- Amin, Adnan. “Off-line Arabic character recognition: the state of the art.” *Pattern Recognition* 31.5 (1998): 517–530.
- Antani, Sameer and Lalitha Agnihotri. “Gujarati character recognition.” *Document Analysis and Recognition, 1999. ICDAR’99. Proceedings of the Fifth International Conference on*. IEEE. 1999. 418–421.
- Arica, Nafiz and Fatos T Yarman-Vural. “An overview of character recognition focused on off-line handwriting.” *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 31.2 (2001): 216–233.
- Arora, Sandhya, et al. “A two stage classification approach for handwritten Devnagari characters.” *Conference on Computational Intelligence and Multimedia Applications, 2007. International Conference on*. IEEE. 2007. 399–403.
- Arora, Sandhya, et al. “Recognition of non-compound handwritten Devnagari characters using a combination of mlp and minimum edit distance.” *arXiv preprint arXiv:1006.5908* (2010).
- Arora, Sandhya, et al. “Study of different features on handwritten Devnagari character.” *Emerging Trends in Engineering and Technology (ICETET), 2009 2nd International Conference on*. IEEE. 2009. 929–933.
- Baheti, MJ, KV Kale, and ME Jadhav. “Comparison of classifiers for gujarati numeral recognition.” *International Journal of Machine Intelligence* 3.3 (2011).
- Bansal, Veena and RMK Sinha. “A complete OCR for printed Hindi text in Devanagari script.” *Document Analysis and Recognition, 2001. Proceedings. Sixth International Conference on*. IEEE. 2001. 800–804.
- . “Integrating knowledge sources in Devanagari text recognition system.” *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans* 30.4 (2000): 500–505.

- Bellili, Abdel, Michel Gilloux, and Patrick Gallinari. "An MLP-SVM combination architecture for offline handwritten digit recognition." *Document Analysis and Recognition* 5.4 (2003): 244–252.
- Bhaskarabhatla, Ajay S and Sriganesh Madhvanath. "Experiences in Collection of Handwriting Data for Online Handwriting Recognition in Indic Scripts." *LREC*. 2004.
- Bhattacharya, Ujjwal and BB Chaudhuri. "Databases for research on recognition of handwritten characters of Indian scripts." *Document Analysis and Recognition, 2005. Proceedings. Eighth International Conference on*. IEEE. 2005. 789–793.
- Bhattacharya, Ujjwal, SK Ghosh, and S Parui. "A two stage recognition scheme for handwritten Tamil characters." *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*. IEEE. 2007. 511–515.
- Bishop, C. "Pattern Recognition and Machine Learning (Information Science and Statistics), 1st edn. 2006. corr. 2nd printing edn." *Springer, New York* (2007).
- Bortolozzi, Flávio, et al. "Recent advances in handwriting recognition." *Document Analysis* (2005): 1–31.
- Burges, Christopher JC. "A tutorial on support vector machines for pattern recognition." *Data Mining and Knowledge Discovery* 2.2 (1998): 121–167.
- Camastra, Francesco. "A SVM-based cursive character recognizer." *Pattern Recognition* 40.12 (2007): 3721–3727.
- Casey, Richard G. and George Nagy. *Advances in Pattern Recognition*. Vol. 224. Edited by Venu Govindaraju. 1971. 56–71.
- Casey, Richard G and George Nagy. "Advances in pattern recognition." *Scientific American* 224 (1971): 56–71.
- Chanda, Sukalpa, et al. "Two-stage approach for word-wise script identification." *Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on*. IEEE. 2009. 926–930.
- Chaudhary, Mandar, et al. "Similar looking Gujarati printed character recognition using Locality Preserving Projection and artificial neural networks." *Emerging Applications of Information Technology (EAIT), 2012 Third International Conference on*. IEEE. 2012. 153–156.



- Chaudhuri, BB and U Pal. “An OCR system to read two Indian language scripts: Bangla and Devnagari (Hindi).” *Document Analysis and Recognition, 1997., Proceedings of the Fourth International Conference on.* IEEE. 1997. 1011–1015.
- Chen, GY, Tien D Bui, and Adam Krzyzak. “Contour-based handwritten numeral recognition using multiwavelets and neural networks.” *Pattern Recognition* 36.7 (2003): 1597–1604.
- Cheriet, Mohamed, et al. *Character recognition systems: a guide for students and practitioners.* John Wiley & Sons, 2007.
- Cheriet, Mohamed, et al. “Handwriting recognition research: Twenty years of achievement and beyond.” *Pattern Recognition* 42.12 (2009): 3131–3135.
- Das, Nibaran, et al. “A statistical–topological feature combination for recognition of handwritten numerals.” *Applied Soft Computing* 12.8 (2012): 2486–2495.
- Desai, Apurva A. “Gujarati handwritten numeral optical character reorganization through neural network.” *Pattern Recognition* 43.7 (2010): 2582–2589.
- . “Support vector machine for identification of handwritten Gujarati alphabets using hybrid feature space.” *CSI Transactions on ICT* 2.4 (2015): 235–241.
- Deshpande, PS, Latesh Malik, and Sandhya Arora. “Fine Classification & Recognition of Hand Written Devnagari Characters with Regular Expressions & Minimum Edit Distance Method.” *JCP* 3.5 (2008): 11–17.
- Dhingra, Kapil Dev, Sudip Sanyal, and Pramod Kumar Sharma. “A robust OCR for degraded documents.” *Advances in Communication Systems and Electrical Engineering.* Springer, 2008. 497–509.
- Dholakia, Jignesh, Atul Negi, and S Rama Mohan. “Progress in Gujarati document processing and character recognition.” *Guide to OCR for Indic Scripts.* Springer, 2009. 73–95.
- . “Zone identification in the printed Gujarati text.” *Document Analysis and Recognition, 2005. Proceedings. Eighth International Conference on.* IEEE. 2005. 272–276.
- Dholakia, Jignesh, Archit Yajnik, and Atul Negi. “Wavelet feature based confusion character sets for Gujarati script.” *International conference on Computational Intelligence and Multimedia Applications.* IEEE. 2007. 366–370.

- Dhurandhar, Amit, Kartik Shankarnarayanan, and Rakesh Jawale. “Robust pattern recognition scheme for Devanagari script.” *International Conference on Computational and Information Science*. Springer. 2005. 1021–1026.
- Dong, Jian-xiong, Adam Krzyzak, and Ching Y Suen. “An improved handwritten Chinese character recognition system using support vector machine.” *Pattern Recognition Letters* 26.12 (2005): 1849–1856.
- Dwyer, Rachel. *The poetics of devotion: the Gujarati lyrics of Dayaram*. Psychology Press, 2001.
- Freeman, Herbert. “On the encoding of arbitrary geometric configurations.” *IRE Transactions on Electronic Computers* 2 (1961): 260–268.
- Gandhi, Mahatma. *Hind swaraj, or, Indian home rule*. Navajivan Publishing House Ahmedabad, 1939.
- Gandhi, MK. *Satya-na Prayogo - Atmakatha (My Experiments with Truth - Autobiography)*. Navjeevan Publishers.
- Ghosh, Debashis, Tulika Dube, and Adamane Shivaprasad. “Script recognition - a review.” *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32.12 (2010): 2142–2161.
- Goswami, Mukesh and Suman Mitra. “Structural feature based classification of printed Gujarati characters.” *International Conference on Pattern Recognition and Machine Intelligence*. Springer. 2013. 82–87.
- Goswami, Mukesh M and Suman K Mitra. “Classification of Printed Gujarati Characters Using Low-Level Stroke Features.” *ACM Transactions on Asian and Low-Resource Language Information Processing* 15.4 (2016): 25.
- . “Offline handwritten Gujarati numeral recognition using low-level strokes.” *International Journal of Applied Pattern Recognition* 2.4 (2015): 353–379.
- Goswami, Mukesh M, Harshad B Prajapati, and Vipul K Dabhi. “Classification of printed Gujarati characters using SOM based k-Nearest Neighbor Classifier.” *Image Information Processing (ICIIP), 2011 International Conference on*. IEEE. 2011. 1–5.

- Govindaraju, Venu, et al. "Tools for enabling digital access to multi-lingual Indic documents." *Document Image Analysis for Libraries, 2004. Proceedings. First International Workshop on*. IEEE. 2004. 122–133.
- Hanmandlu, Madasu, OV Ramana Murthy, and Vamsi Krishna Madasu. "Fuzzy Model based recognition of handwritten Hindi characters." *Digital Image Computing Techniques and Applications, 9th Biennial Conference of the Australian Pattern Recognition Society on*. IEEE. 2007. 454–461.
- Hassan, Ehtesham, Santanu Chaudhury, and Madan Gopal. "Feature combination for binary pattern classification." *International Journal on Document Analysis and Recognition (IJDAR)* 17.4 (2014): 375–392.
- Hull, Jonathan J. "A database for handwritten text recognition research." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 16.5 (1994): 550–554.
- Impedovo, Donato and Giuseppe Pirlo. "Zoning methods for handwritten character recognition: A survey." *Pattern Recognition* 47.3 (2014): 969–981.
- Impedovo, Sebastiano. "More than twenty years of advancements on Frontiers in handwriting recognition." *Pattern Recognition* 47.3 (2014): 916–928.
- Jawahar, CV, MNSSK Pavan Kumar, and SS Ravi Kiran. "A bilingual OCR for hindi-telugu documents." *Technical Report TR-CVIT-22, IIT* (2002).
- Jayanthi, Krishnamachari, et al. "Devanagari character recognition using structure analysis." *TENCON'89. Fourth IEEE Region 10 International Conference*. IEEE. 1989. 363–366.
- Jindal, MK, RK Sharma, and GS Lehal. "Segmentation of horizontally overlapping lines in printed gurmukhi script." *Advanced Computing and Communications, 2006. ADCOM 2006. International Conference on*. IEEE. 2006. 226–229.
- Jomy, John, KV Pramod, and Balakrishnan Kannan. "Handwritten character recognition of south Indian scripts: a review." *arXiv preprint arXiv:1106.0107* (2011).
- Kale, Karbhari V, et al. "Zernike moment feature extraction for handwritten Devanagari compound character recognition." *Science and Information Conference (SAI), 2013*. IEEE. 2013. 459–466.
- Kasturi, Rangachar, Lawrence O'gorman, and Venu Govindaraju. "Document image analysis: A primer." *Sadhana* 27.1 (2002): 3–22.

- Kavallieratou, Ergina, Nikos Fakotakis, and G Kokkinakis. "Handwritten character recognition based on structural characteristics." *Pattern Recognition, 2002. Proceedings. 16th International Conference on*. IEEE. 2002. 139–142.
- Khanduja, Deepti, Neeta Nain, and Subhash Panwar. "A Hybrid Feature Extraction Algorithm for Devanagari Script." *ACM Transactions on Asian and Low-Resource Language Information Processing* 15.1 (2016): 2.
- Kompalli, Suryaprakash, Srirangaraj Setlur, and Venu Govindaraju. "Design and comparison of segmentation driven and recognition driven Devanagari OCR." *Document Image Analysis for Libraries, 2006. DIAL'06. Second International Conference on*. IEEE. 2006. 7–pp.
- Kompalli, Suryaprakash, et al. "Challenges in OCR of Devanagari documents." *Document Analysis and Recognition, 2005. Proceedings. Eighth International Conference on*. IEEE. 2005. 327–331.
- Kumar, Satish. "Performance comparison of features on Devanagari handprinted dataset." *Int. J. Recent Trends* 1.2 (2009): 33–37.
- LeCun, Yann, et al. "Backpropagation applied to handwritten zip code recognition." *Neural Computation* 1.4 (1989): 541–551.
- LeCun, Yann, et al. "Gradient-based learning applied to document recognition." *Proceedings of the IEEE* 86.11 (1998): 2278–2324.
- Lee, Seong-Whan, et al. "Multiresolution recognition of unconstrained handwritten numerals with wavelet transform and multilayer cluster neural network." *Pattern Recognition* 29.12 (1996): 1953–1961.
- Lewis, JP. "Fast normalized cross-correlation." *Vision Interface*. 1995. 120–123.
- Liu, Cheng-Lin, et al. "CASIA online and offline Chinese handwriting databases." *Document Analysis and Recognition (ICDAR), 2011 International Conference on*. IEEE. 2011. 37–41.
- Liu, Cheng-Lin, et al. "Handwritten digit recognition: benchmarking of state-of-the-art techniques." *Pattern Recognition* 36.10 (2003): 2271–2285.
- Lorigo, Liana M and Venugopal Govindaraju. "Offline Arabic handwriting recognition: a survey." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28.5 (2006): 712–724.

- Ma, Huanfeng and David Doermann. "Adaptive Hindi OCR using generalized Hausdorff image comparison." *ACM Transactions on Asian Language Information Processing (TALIP)* 2.3 (2003): 193–218.
- Maloo, Mamta and KV Kale. "Support vector machine based Gujarati numeral recognition." *International Journal on Computer Science and Engineering* 3.7 (2011): 2595–2600.
- Mane, Vanita and Lena Ragha. "Handwritten character recognition using elastic matching and PCA." *Proceedings of the International Conference on Advances in Computing, Communication and Control*. ACM. 2009. 410–415.
- Mantas, J. "An overview of character recognition methodologies." *Pattern Recognition* 19.6 (1986): 425–430.
- Mori, Shunji, Ching Y Suen, and Kazuhiko Yamamoto. "Historical review of OCR research and development." *Proceedings of the IEEE* 80.7 (1992): 1029–1058.
- Nagar, Ravi and Suman K Mitra. "Feature extraction based on stroke orientation estimation technique for handwritten numeral." *Advances in Pattern Recognition (ICAPR), 2015 Eighth International Conference on*. IEEE. 2015. 1–6.
- Natarajan, Premkumar S, Ehry MacRostie, and Michael Decerbo. "The bbn byblos hindi ocr system." *Electronic Imaging 2005*. International Society for Optics and Photonics. 2005. 10–16.
- Otsu, Nobuyuki. "A threshold selection method from gray-level histograms." *Automatica* 11.285-296 (1975): 23–27.
- Pal, U and BB Chaudhuri. "Indian script character recognition: a survey." *pattern Recognition* 37.9 (2004): 1887–1899.
- Pal, Umapada, Ramachandran Jayadevan, and Nabin Sharma. "Handwriting recognition in indian regional scripts: a survey of offline techniques." *ACM Transactions on Asian Language Information Processing (TALIP)* 11.1 (2012): 1.
- Pal, Umapada, Tetsushi Wakabayashi, and Fumitaka Kimura. "Comparative study of Devnagari handwritten character recognition using different feature and classifiers." *Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on*. IEEE. 2009. 1111–1115.

- Pal, Umapada, et al. "Accuracy improvement of Devnagari character recognition combining SVM and MQDF." *Proc. 11th International Conference on Frontiers in Handwriting Recognition*. Citeseer. 2008. 367–372.
- Pal, Umapada, et al. "Off-line handwritten character recognition of Devnagari script." *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*. IEEE. 2007. 496–500.
- Patel, Chhaya and Apurva Desai. "Gujarati handwritten character recognition using hybrid method based on binary tree-classifier and k-nearest neighbour." *International Journal of Engineering Research and Technology*. ESRSA Publications. 2013.
- Plamondon, Réjean and Sargur N Srihari. "Online and off-line handwriting recognition: a comprehensive survey." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22.1 (2000): 63–84.
- Saito, T, H Yamada, and K Yamamoto. "An analysis of handprinted character database VIII: An estimation of the database ETL9 of handprinted characters in JIS Chinese characters by directional pattern matching approach." *Bul. Electrotech* 49.7 (1985): 487–525.
- Sanossian, Hermineh YY. "An Arabic character recognition system using neural network." *Neural Networks for Signal Processing [1996] VI. Proceedings of the 1996 IEEE Signal Processing Society Workshop*. IEEE. 1996. 340–348.
- Sezgin, Mehmet, et al. "Survey over image thresholding techniques and quantitative performance evaluation." *Journal of Electronic Imaging* 13.1 (2004): 146–168.
- Shah, Lipi, et al. "Handwritten Character Recognition using Radial Histogram." *International Journal of Research in Advent Technology* 2.4 (2014): 24–28.
- Shah, SK and A Sharma. "Design and implementation of optical character recognition system to recognize Gujarati script using template matching." *Journal of the Institution of Engineers (India) Part ET, Electronics and telecommunication engineering division* 86.N (2006): 44.
- Sharma, Ankit K, et al. "Comparative analysis of zoning based methods for Gujarati handwritten numeral recognition." *Engineering (NUiCONE), 2015 5th Nirma University International Conference on*. IEEE. 2015. 1–5.

- Sharma, Nabin, et al. "Recognition of off-line handwritten Devnagari characters using quadratic classifier." *Computer Vision, Graphics and Image Processing*. Springer, 2006. 805–816.
- Sinha, RMK and HN Mahabala. "Machine recognition of Devanagari script." *IEEE Transactions on Systems, Man, and Cybernetics* 9.8 (1979): 435–441.
- Soman, KP, Shyam Diwakar, and V Ajay. *Data Mining: Theory and Practice*. PHI Learning Pvt. Ltd., 2006.
- Srihari, Sargur N. "Recognition of handwritten and machine-printed text for postal address interpretation." *Pattern Recognition Letters* 14.4 (1993): 291–302.
- Suen, Ching Y, Marc Berthod, and Shunji Mori. "Automatic recognition of hand-printed characters-the state of the art." *Proceedings of the IEEE* 68.4 (1980): 469–487.
- Suen, Ching Y, et al. "Computer recognition of unconstrained handwritten numerals." *Proceedings of the IEEE* 80.7 (1992): 1162–1180.
- Thaker, H and C Kumbharana. "Analysis of Structural Features and Classification of Gujarati Consonant for Offline Character Recognition." *International Journal of Scientific and Research Publications* 4.8 (2014).
- Topiwala, C. *Gujarati Sahityakosh VOL.3*. Gujarati Sahitya Parishad, 1996.
- Trier, Oeivind Due and Torfinn Taxt. "Evaluation of binarization methods for document images." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 17.3 (1995): 312–315.
- Trier, Øivind Due, Anil K Jain, and Torfinn Taxt. "Feature extraction methods for character recognition-a survey." *Pattern Recognition* 29.4 (1996): 641–662.
- Tzanakou, Evangelia Miche. *Supervised and unsupervised pattern recognition: feature extraction and computational intelligence*. CRC Press, 1999.
- Vamvakas, Georgios, Basilis Gatos, and Stavros J Perantonis. "Handwritten character recognition through two-stage foreground sub-sampling." *Pattern Recognition* 43.8 (2010): 2807–2816.
- Wilkinson, R, et al. "The first census optical character recognition systems." *The US Bureau of Census and the National Institute of Standards and Technology, Gaithersburg, MD* (1992).

- Wunsch, Patrick and Andrew F Laine. “Wavelet descriptors for multiresolution recognition of handprinted characters.” *Pattern Recognition* 28.8 (1995): 1237–1249.
- Yamada, Hiromitsu, Kazuhiko Yamamoto, and Taiichi Saito. “A nonlinear normalization method for handprinted Kanji character recognition-line density equalization.” *Pattern Recognition* 23.9 (1990): 1023–1029.
- Zhang, Honggang, et al. “HCL2000-A large-scale handwritten Chinese character database for handwritten character recognition.” *Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on*. IEEE. 2009. 286–290.
- Zhang, Ping, Tien D Bui, and Ching Y Suen. “A novel cascade ensemble classifier system with a high recognition performance on handwritten digits.” *Pattern Recognition* 40.12 (2007): 3415–3429.