

Pattern Analysis Techniques for the Recognition of Unconstrained Handwritten Malayalam Character Images

Thesis submitted to
Cochin University of Science and Technology
in partial fulfillment for the award of the Degree of
DOCTOR OF PHILOSOPHY
Under the Faculty of Technology

By
JOMY JOHN
(Reg. No. 3539)

Under the guidance of
Dr. K. V. PRAMOD



Department of Computer Applications
Cochin University of Science and Technology
Cochin – 682 022, Kerala, India

January 2014



Dr. K. V. Pramod (Supervising Guide)

Associate Professor

Dept. of Computer Applications

Cochin University of Science and Technology

Certificate

Certified that the thesis entitled “Pattern Analysis Techniques for the Recognition of Unconstrained Handwritten Malayalam Character Images” is a bonafide record of research carried out by Jomy John under my guidance in the Department of Computer Applications, Cochin University of Science and Technology, Kochi-22. The work does not form part of any dissertation submitted for the award of any degree, diploma, associate ship or any other title or recognition from any University.

Kochi - 22

Date:30-1-2014

Dr. K. V. Pramod



Dr. K. V. Pramod (Supervising Guide)

Associate Professor

Dept. of Computer Applications

Cochin University of Science and Technology

Certificate

This is to certify that all the relevant corrections and modifications suggested by the audience during the Pre-synopsis seminar and recommendations by the Doctoral Committee of the candidate have been incorporated in the thesis.

Kochi - 22

Date: 30-1-2014

Dr. K. V. Pramod

Declaration

I hereby declare that the thesis entitled "Pattern Analysis Techniques for the Recognition of Unconstrained Handwritten Malayalam Character Images" is the outcome of the original work done by me under the guidance of Dr. K. V. Pramod, Associate Professor, Department of Computer Applications, Cochin University of Science and Technology, Kochi-22. The work does not form part of any dissertation submitted for the award of any degree, diploma, associate ship or any other title or recognition from any University.

Kochi - 22

Jomy John

Date:30-1-2014

Acknowledgement

This thesis is the outcome of the research that I began few years ago. First and foremost, I thank God Almighty who bestowed upon me, the courage, patience, strength and suitable circumstances to embark upon this work and carry it to final completion.

I take this opportunity to express my heartfelt gratitude to my supervising guide Dr. K. V. Pramod, Associate Professor, Department of Computer Applications, Cochin University of Science and Technology for his kind advice, dedication, encouragement, constant motivation and also for his inspiring guidance during these years. I wish to express my hearty thanks to him for helping me in achieving this goal successfully.

I am greatly indebted to Dr. B. Kannan, Head, Department of Computer Applications, Cochin University of Science and Technology for his generous support, encouragement, valuable suggestions and kindness to help me in any way possible.

I would also like to thank Prof. B. B. Chaudhuri, Head, Computer Vision & Pattern Recognition Unit, Indian Statistical Institute for many valuable discussions, continuous support and critical comments in addition to the offer of writing a joint paper. I am immensely benefited by his domain knowledge in the area of character recognition.

I acknowledge all the faculties of Cochin University of Science and Technology particularly Dr. M. Jathavedan, Dr. A. Sree Kumar, Ms. S. Malathi for

their cooperation and support. My sincere thanks are due to all non teaching staff of the department for their assistance and for providing facilities during these years.

I thank every one of the research scholars of the department of Computer Applications especially Simily Joseph, Bino Sebastian, Remya A. R., Ramkumar R., Sindhumol S., Binu V. P. and Santhoshkumar M. B. for their valuable ideas and suggestions. I am grateful to all my colleagues and friends for the support they had given me during this period.

I sincerely thank University Grants Commission for providing fellowship and the Director of Collegiate Education for sanctioning deputation under Faculty Improvement Programme.

It is beyond words to express my gratitude to my loving parents and my siblings for their support throughout my research period. I thank my father-in-law, mother-in-law and my relatives for their prayers and blessings. I express my gratitude to my cousin, Jissy Mathew for providing help in data collection.

My husband, Dr. A. Eldos, is the driving force behind this accomplishment. His inspiration, encouragement and loving support made me carry out this tremendous work. My children, Eby and Merin sacrificed many pleasures that they duly deserve, for my research work. Their moral support was immeasurable during my tough times. I dedicate my achievement to all of them for the everlasting support they have rendered throughout my life.

Jomy John

Preface

Character recognition is a special branch of pattern recognition, contributing enormously towards the improvement of automation process. Machine recognition of handwritings is challenging, as human writing varies from person to person. Even for the same person it varies depending on the speed, mood or the environment. This process involves the conversion of scanned images of handwriting data into digital format. Handwriting recognition is a subject of active research for decades due to its versatile range of applications including processing of bank cheques, mail addresses, white board reading and recognition of handwritten manuscripts. Handwritten character recognition is the integral part of handwriting recognition. Isolated handwritten characters appear largely in census forms, tax forms, application forms in banks, reservation counters etc. The automatic processing of such data itself is of great importance as they are collected in large volumes.

Many promising results were reported in the area of handwritten character recognition research for languages like English,

Chinese, Japanese and Arabic. In Indian scenario, the studies are insufficient and most of the pieces of work deal with Devanagari and Bangla script, the two most popular scripts in India. Malayalam is one among the twenty two scheduled languages of India, spoken by 33 million people, with official language status in the state of Kerala and union territories of Lakshadweep and Puducherry. Works on Malayalam started recently and all the existing studies are limited to just basic characters of the script. There is no work on compound characters or vowel-consonant signs. However, the usage of compound characters and vowel-consonant signs are common in written Malayalam. Therefore, the objective of this study is to develop a handwritten character recognition system that could recognize all the characters in the modern script of Malayalam language at a high recognition rate.

Chapter 1. Introduction describes the work presented in this thesis. The motivation of the work, challenges, objectives and major contributions are outlined.

Chapter 2: Literature Survey presents review to bring out the present status of character recognition research by discussing major character recognition methodologies evolved over times. The survey starts with discussion on the historical background of character recognition; followed by advancements in the global scenario and then Indian scenario with special focus to Malayalam language.

Chapter 3: Development of a Comprehensive Dataset and Preprocessing begins with the description of Malayalam language and script and derives the set of symbols needed for data collection. This chapter describes the data collection and data preparation methods. The description of generated data is also provided. The major preprocessing techniques used to improve the quality of images are described.

Chapter 4: Feature Extraction discusses all the feature extraction methods used in this thesis. Features from both spatial domain and transform domain are explored. A novel feature descriptor using gradient features is discussed. Feature dimensionality reduction

technique such as Principal Component Analysis is also covered in this chapter.

Chapter 5: Performance Evaluation Using Different Classifiers

discusses machine learning classifiers such as k-NN, SVM and ELM for classification. Performances of all the feature extraction methods described in the previous chapter are identified. The performance in the reduced dimension is also covered.

Chapter 6: Accuracy Improvement through Two-Stage Approach and

Class Specific Features presents the difficulty induced with single stage classification. In this chapter, the need for a two-stage classification approach is discussed. A novel attempt of designing class specific features is outlined.

Chapter 7: Conclusion and Future summarizes the thesis and mentions the possible extensions for future works.

List of Publications

International Journals

1. **Jomy John**, Kannan Balakrishnan, K. V. Pramod, "A system for offline recognition of handwritten characters in Malayalam script", **International Journal of Image, Graphics and Signal Processing (IJIGSP)**, MECS-Press, Volume 5, Issue 4, 2013, Pages 53-59, ISSN 2074-9074, DOI: 10.5815/ijigsp.2013.04.07
2. **Jomy John**, Kannan Balakrishnan, K. V. Pramod, "A Novel Feature Descriptor for Malayalam Handwritten Character Recognition Using Support Vector Machine and Extreme Learning Machine", **Central European Journal of Computer Science**, Versita, co-published with Springer Verlag (Communicated)
3. **Jomy John**, Pramod K. V., Kannan Balakrishnan, "Unconstrained Handwritten Malayalam Character Recognition using Wavelet Transform and Support Vector Machine Classifier", **Procedia**

Engineering, Elsevier, Volume 30, 2012, Pages 598–605, ISSN 1877–7058, 10.1016/j.proeng.2012.01.904.

(www.sciencedirect.com/science/article/pii/S18777058120091)

4. **Jomy John**, Kannan Balakrishnan, K. V. Pramod, “Malayalam Character Recognition System for Camera Enabled Mobile Devices”, **International Journal of Advanced Research in Computer Science**, Volume 3 No. 6 (Nov-Dec 2012)
5. **Jomy John**, Pramod K. V., Kannan Balakrishnan, “Topological Features for Malayalam Handwritten Character Recognition”, **Journal of Cybernetics and Systems, Taylor and Francis** (Communicated)

International Conferences

6. **Jomy John**, K. V. Pramod, Kannan Balakrishnan, Bidyut B. Chaudhuri, “A two stage approach for handwritten Malayalam character recognition”, **14th International Conference on**

Frontiers in Handwriting Recognition (ICFHR-2014), September 1-4, 2014, Crete, Greece.

7. **Jomy John**, Kannan Balakrishnan, Pramod K. V., "Grouping scheme for the recognition of handwritten Malayalam vowels, consonants, compound characters and vowel-consonant signs", **IEEE International Conference on Communication and Signal Processing - ICCSP13**, April 2-4, 2013, Melmaruvathur, India.
8. **Jomy John**, Pramod K.V, Balakrishnan, K., "Offline handwritten Malayalam Character Recognition based on chain code histogram", **IEEE International Conference on Emerging Trends in Electrical and Computer Technology**, pp.736-741, March 23-24, 2011, Kanyakumari, India.

URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5760215&isnumber=5760077>
9. Simily Joseph, **Jomy John**, Kannan Balakrishnan, Pramod K. V., "Content Based Image Retrieval System for Malayalam Handwritten Characters", **IEEE International Conference on**

Network and Computer Science, April 8-10, 2011, Kanyakumari,
India.

URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5942026&isnumber=5941942>

National Conference

10. **Jomy John**, Pramod K. V, Kannan Balakrishnan , "Handwritten Character Recognition of South Indian Scripts: A Review", **National Conference on Indian Language Computing**, Cochin, February 19-20, 2011, arXiv preprint arXiv:1106.0107

Contents

Chapter 1: Introduction.....	1
1.1 Preamble.....	1
1.2 Application Domain.....	3
1.3 Overview of Handwritten Character Recognition	3
1.4 Motivation.....	4
1.5 Problem Definition.....	6
1.6 Challenges.....	6
1.7 Objectives	7
1.8 Framework of the Proposed System	7
1.8.1 Digitization	9
1.8.2 Preprocessing.....	10
1.8.3 Feature Extraction	10
1.8.4 Feature Reduction	11
1.8.5 Classification	11
1.8.6 Classification in Two Stages.....	12
1.9 Thesis Contributions.....	13
1.10 Conclusion.....	14
Chapter 2: Literature Survey.....	17
2.1 Introduction.....	17
2.2 Historical Background	18
2.3 Developments of HCR in the Global Scenario	20
2.3.1 Benchmark Databases of HCR Research.....	21
2.3.2 Latin and Greek.....	22

2.3.3 Chinese.....	25
2.3.4 Japanese	27
2.3.5 Arabic.....	28
2.3.6 Korean.....	30
2.4 Developments of HCR in the Indian Scenario	31
2.4.1 Languages in India.....	32
2.4.2 Characteristics of Indian Scripts	33
2.4.3 Databases of Indian HCR.....	34
2.4.4 Developments on North Indian Scripts.....	35
2.4.4.1 Devnagari	35
2.4.4.2 Bangla.....	40
2.4.4.3 Gujarati.....	42
2.4.4.4 Oriya	43
2.4.4.5 Gurumukhi.....	45
2.4.4.6 Urdu.....	46
2.4.5 Developments on South Indian Scripts.....	48
2.4.5.1 Tamil.....	48
2.4.5.2 Telugu	51
2.4.5.3 Kannada	52
2.4.5.4 Malayalam.....	54
2.5 Summary of Developments in Indian Scripts.....	56
2.6 Commercial Character Recognition Softwares.....	61
2.6.1 Machine Printed.....	61
2.6.2 Handwritten.....	61
2.7 Conclusion	62

Chapter 3: Development of a Comprehensive Dataset and Preprocessing	65
3.1 Introduction.....	66
3.2 Malayalam Language.....	66
3.3 Malayalam Script Overview	67
3.4 Data Collection	73
3.5 Data Preparation	75
3.5.1 Skew Detection and Correction.....	77
3.5.2 Segmentation	78
3.5.2.1 Horizontal Projection Profile	78
3.5.2.2 Connected Component Labelling	78
3.5.2.3 Character Extraction Algorithm.....	79
3.6 Description of Generated Data.....	81
3.7 Document Acquisition and Enhancement through Mobile Phone Camera.....	87
3.8 Data Preprocessing.....	89
3.8.1 Smoothing and Noise Removal.....	91
3.8.2 Normalization of the Image	91
3.8.3 Binarization of the Image	92
3.8.4 Thinning of the Binary Image	94
3.8.5 Boundary (Contour) Extraction of the Binary Image	97
3.8.6 Gray Scale Normalization of the Image	97
3.9 Conclusion	99
Chapter 4: Feature Extraction	101
4.1 Introduction.....	101
4.2 Topological Features	104

4.3	<i>Distribution of Foreground Pixels</i>	107
4.4	<i>Transition Count Features</i>	109
4.5	<i>Local Binary Pattern Descriptors</i>	110
4.6	<i>Wavelet Features</i>	112
4.6.1	<i>Wavelet Theory</i>	113
4.6.2	<i>Haar Wavelets</i>	115
4.6.3	<i>Feature Extraction using Wavelet</i>	116
4.7	<i>Chain Code Features</i>	119
4.8	<i>Gradient Features</i>	123
4.8.1	<i>Definition of Gradient</i>	123
4.8.2	<i>Computation of Gradient</i>	124
4.8.3	<i>Feature Extraction using Gradient</i>	125
4.9	<i>Curvature Features</i>	128
4.9.1	<i>Definition of Curvature</i>	128
4.9.2	<i>Feature Extraction using Curvature</i>	130
4.10	<i>Design of Novel Feature Descriptor Based on Image Gradient</i>	132
4.11	<i>Reduction of Feature Dimension</i>	136
4.12	<i>Conclusion</i>	139
Chapter 5: Performance Evaluation Using Different Classifiers		141
5.1	<i>Introduction</i>	141
5.2	<i>Classification Algorithms</i>	142
5.2.1	<i>K-Nearest Neighbour</i>	143
5.2.2	<i>Support Vector Machines</i>	145
5.2.3	<i>Extreme Learning Machine</i>	149

5.2.3.1	<i>Constrained Optimization based ELM</i>	152
5.3	<i>Performance Evaluation Measures</i>	154
5.3.1	<i>Accuracy</i>	155
5.3.2	<i>Precision</i>	155
5.3.3	<i>Recall</i>	155
5.3.4	<i>F-measure</i>	156
5.4	<i>Performances Obtained Using Different Feature Sets and Classifiers</i>	156
5.4.1	<i>Experimental Setup</i>	157
5.4.2	<i>Performance Evaluation Using Topological Features, Distribution Features, Transition Count Features and LBP Features</i>	160
5.4.2.1	<i>Feature Combination</i>	162
5.4.3	<i>Performance Evaluation Using Wavelet Features</i>	164
5.4.4	<i>Performance Evaluation Using Chain Code Features</i>	166
5.4.5	<i>Performance Evaluation Using Gradient Features</i>	167
5.4.6	<i>Performance Evaluation Using Curvature Features</i>	168
5.4.7	<i>Performance Evaluation Using SSG Features</i>	169
5.4.8	<i>Analysis of Results</i>	170
5.4.9	<i>Performance Evaluation on the Dataset Created Using Mobile Phone Camera</i>	172
5.5	<i>Conclusion</i>	173
Chapter 6:	<i>Accuracy Improvement through Two-Stage Approach and Class Specific Features</i>	177
6.1	<i>Introduction</i>	177
6.2	<i>Design of a Two-Stage Recognizer</i>	178

6.2.1 Architecture of the Two-Stage Recognizer.....	179
6.2.2 Creation of Groups.....	181
6.2.3 Design of Second Stage Classifier.....	183
6.3 Results and Discussions.....	188
6.4 Conclusion.....	190
Chapter 7: Conclusion and Future Scope.....	193
7.1 Conclusion and Major Contributions.....	193
7.2 Future Scope.....	196
References.....	197

List of Tables

<i>Table</i>	<i>Title</i>	<i>Page No</i>
2.1	<i>Summary of developments in Indian scripts.....</i>	<i>57</i>
3.1	<i>Malayalam vowels</i>	<i>68</i>
3.2	<i>Dependent vowel signs.....</i>	<i>68</i>
3.3	<i>Malayalam consonants.....</i>	<i>68</i>
3.4	<i>Pure consonants</i>	<i>69</i>
3.5	<i>Script Revision: Usage of separate symbols for vowel-consonant signs</i>	<i>70</i>
3.6	<i>Compound characters.....</i>	<i>71</i>
3.7	<i>Script Revision: Usage of ് (chandrakkala).....</i>	<i>71</i>
3.8	<i>Frequency of occurrence of top 20 symbols.....</i>	<i>72</i>
3.9	<i>Malayalam numerals.....</i>	<i>73</i>
5.1	<i>Training parameters of ELM network.....</i>	<i>151</i>
5.2	<i>Confusion matrix for two classes</i>	<i>154</i>
5.3	<i>Classification Result of Topological Features, Distribution Features, Transition Count Features and LBP Features</i>	<i>162</i>
5.4	<i>Classification Result of Feature Combination in PCA Feature Space.....</i>	<i>164</i>
5.5	<i>Classification Result of Wavelet Features.....</i>	<i>165</i>
5.6	<i>Classification Result of Wavelet Features in PCA Feature Space.....</i>	<i>166</i>
5.7	<i>Classification Result of Chain Code Features.....</i>	<i>166</i>
5.8	<i>Classification Result of Chain Code Features in PCA Feature Space.....</i>	<i>167</i>

5.9	<i>Classification Result of Gradient Features.....</i>	168
5.10	<i>Classification Result of Gradient Features in PCA Feature Space.....</i>	168
5.11	<i>Classification Result of Curvature Features.....</i>	169
5.12	<i>Classification Result of Curvature Features in PCA Feature Space.....</i>	169
5.13	<i>Classification Result of SSG Features in PCA Feature Space.....</i>	170
5.14	<i>Classification Result on the Dataset Created through Mobile Phone Camera.....</i>	172
6.1	<i>Group Classification Accuracy with SVM.....</i>	189
6.2	<i>Second Stage Classification Results.....</i>	190

List of Figures

Figure No	Caption	Page No
1.1	Overall architecture of the proposed system.....	8
2.1	Classification of writing systems and languages of the present world.....	20
2.2	The Brahmic family of scripts used in India	34
3.1	Statistics of contributors used for data collection.....	75
3.2	A sample filled in data collection sheet	76
3.3	Part of skewed and skew corrected image.....	77
3.4	Extraction of characters from the filled in data collection sheet.....	80
3.5	Manual refinement of samples.....	81
3.6	Samples of handwritten characters along with their class reference.....	85
3.7	Invalid characters deleted from the database.....	85
3.8	A glimpse of handwritten samples in folder 1 and folder 58.....	86
3.9	System framework for acquisition of data through mobile phone camera.....	87
3.10	Processing of camera captured document	88
3.11	Some of the samples from the database	89
3.12	Steps in preprocessing	90
3.13	3×3 averaging filter mask.....	91
3.14	Eight Neighbours of p	95
3.15	Samples of normalized thinned binary image.....	96
3.16	Summary of preprocessing steps	98
4.1	Categorization of used feature extraction methods.....	103

4.2	<i>Samples of loops in handwritten pattern</i>	105
4.3	<i>Character skeleton Ω (/da/)</i>	106
4.4	<i>Character pattern Θ (/ra/) divided into 4×4 zones</i>	108
4.5	<i>Illustration of \mathcal{LBP} operator</i>	111
4.6	<i>Decomposition using analysis filter bank_s</i>	115
4.7	<i>Second-level decomposition of the input image</i>	115
4.8	<i>Original image and decomposition at level 3 of character Θ (/a/)</i>	117
4.9	<i>Types of chain codes</i>	120
4.10	<i>Schematic diagram of chain code based feature extraction method</i>	122
4.11	<i>Sobel horizontal and vertical operators</i>	125
4.12	<i>Direction and strength of gradient of character pattern Θ (/a/)</i>	125
4.13	<i>Decomposition of gradient direction</i>	126
4.14	<i>8-neighbours of x_0 and pixel values in the neighbourhood</i>	129
4.15	<i>Curvature image of handwritten pattern Θ (/a/) and its division into three regions</i>	131
4.16	<i>Sobel diagonal operators</i>	133
4.17	<i>SSG Feature vectors of two typical samples of Θ (/a/) and Ω (/vva/)</i>	135
4.18	<i>Principal Component Analysis: Y_1 and Y_2 are the first two principal components for the given data</i>	137
5.1	<i>The 1-, 2- and 3- nearest neighbours of an unknown pattern x</i>	145
5.2	<i>Optimal separating hyper plane in a two-dimensional space</i>	146
5.3	<i>Inseparable case in two-dimensional space</i>	147
5.4	<i>Architecture of single hidden layer neural network</i>	150

5.5	<i>Parameter Tuning for SVM</i>	158
5.6	<i>Parameter tuning for ELM with sigmoid activation function.....</i>	159
5.7	<i>Parameter tuning for ELM with Gaussian activation function.....</i>	159
5.8	<i>Variations in recognition accuracy based on number of distribution features.....</i>	161
5.9	<i>Scree Plot: Variance Explained by Principal Components.....</i>	163
5.10	<i>Some similar characters in Malayalam.....</i>	171
6.1	<i>Architecture of two stage recognizer.....</i>	180
6.2	<i>Groups.....</i>	182
6.3	<i>Samples of the characters ഇ, ഉ, ഊ.....</i>	184
6.4	<i>Samples of ആ, ഘ.....</i>	186
6.5	<i>Samples of റ, റ്റ, റ്റ, ശ.....</i>	187

List of Algorithms

<i>Algorithm</i>	<i>Title</i>	<i>Page No</i>
3.1	<i>Character Extraction Algorithm</i>	<i>79</i>
4.1	<i>Extraction of Topological Features.....</i>	<i>106</i>
4.2	<i>Extraction of LBP Features.....</i>	<i>111</i>
4.3	<i>Extraction of Wavelet Features</i>	<i>118</i>
4.4	<i>Extraction of Chain Code Features.....</i>	<i>122</i>
4.5	<i>Extraction of Gradient Features.....</i>	<i>126</i>
4.6	<i>Extraction of Curvature Features.....</i>	<i>131</i>
4.7	<i>Creation of SSG Features.....</i>	<i>135</i>

List of Acronyms

2D	Two dimensional
Acc	Accuracy
AMD	Asymmetric Mahalanobis Distance
ANN	Artificial Neural Network
BPNN	Back Propagation Neural Network
CBDD	City Block Distance with Deviation
CCA	Connected Component Analysis
CMF	Compound Mahalanobis Function
CMNN	Class Modular Neural Network
CNN	Convolutional Neural Network
DEF	Directional Element Feature
DLQDF	Discriminative Learning Quadratic Discriminant Function
DWT	Discrete Wavelet Transform
ELM	Extreme Learning Machine
ELM-noReg	ELM without regularization factor
ELM-opt	Optimization based ELM
EM	Expectation Maximization
FLDA	Fisher Linear Discriminant Analysis
FN	False Negative
FP	False Positive
FZ	Fuzzy Zoning
GA	Genetic Algorithm
HCR	Handwritten Character Recognition
HMCR	Handwritten Malayalam Character Recognition
HMM	Hidden Markov Model
HNN	Hierarchical Neural Network
HRG	Hierarchical Random Graph

HWT	Haar Wavelet Transform
ICA	Independent Component Analysis
k-NN	k-Nearest Neighbour
LBP	Local Binary Pattern
LDA	Linear Discriminant Analysis
LVQ	Learning Vector Quantization
MLP	Multi-Layer Perceptron
MQDF	Modified Quadratic Discriminant Function
NN	Neural Network
OCR	Optical Character Recognition
PCA	Principal Component Analysis
PNN	Probabilistic Neural Network
QDF	Quadratic Discriminant Function
RBF	Radial Basis Function
RLC	Run Length Count
SIFT	Scale Invariant Feature Transform
SLFN	Single Hidden Layer Feed Forward Neural Network
SSG	Sobel-Sobel Gradient
SSM	Scale Space Map
SSPD	Scale Space Point Distribution
SVM	Support Vector Machine
TN	True Negative
TP	True Positive
TPID	Transformation based Partial Inclination Detection

Chapter 1

INTRODUCTION

<i>Contents</i>	1.1	Preamble
	1.2	Application Domain
	1.3	Overview of Handwritten Character Recognition
	1.4	Motivation
	1.5	Problem Definition
	1.6	Challenges
	1.7	Objectives
	1.8	Framework of the Proposed System
	1.9	Thesis Contributions
	1.10	Conclusion

1.1 Preamble

Character recognition is a special branch of pattern recognition, contributing enormously towards the improvement of automation process. It refers to the translation of handwritten or printed text into machine readable text. Machine printed character recognition system analyses layout of the document and interprets textual content while handwriting recognition address the variability in writing styles of individual. Machine recognition of handwritings is challenging, as human writing varies from person to person and even for the same person depending on the speed, mood or the environment. Based on the way in which handwritings are drawn, there are two approaches, namely, online and offline. The former

involves recognition of writings on an electronic surface such as a digitizer with a special pen, where the writer's pen movements, velocity, acceleration and stroke order can be traced at the time of writing. Online recognition is mainly dedicated to security domains such as signature verification and author authentication. The latter involves the conversion of scanned images of handwriting data into digital form. Offline recognition is comparatively a difficult problem as the only information available is a set of pixels and consequently, the recognition rates reported are lower compared to online [1]. However, the ease of writing with pen and paper is not available with keyboard or digitizer and therefore, handwriting persists to continue as means of communication and recording information in day-to-day life even with the introduction of new technologies. In spite of enormous efforts by a large number of scientists, engineers and environmentalists to promote a paperless society, to date we are still faced with all sorts of documents at home and at work [2]. Moreover, handwriting appears to be the most direct, natural way of expressing the ideas of our brain and at the same time, the use of the hand to write words and draw illustrations seems to simulate the brain more than the use of electronic means, as well as to enhance the memory [3]. Accordingly, our research is focused on the area of offline recognition of handwritten characters.

1.2 Application Domain

Offline handwriting recognition system has versatile range of applications including processing of bank cheques, mail addresses, white board reading, recognition of handwritten manuscripts etc. When coupled with speech processing, handwriting recognition system can provide an interface to the visually impaired [4]. Consequently, handwriting recognition is a subject of active research for decades [5-8]. Handwritten character recognition is an integral part of handwriting recognition. Isolated handwritten characters appear largely in census forms, tax forms, application forms in banks, reservation counters etc. The automatic processing of such data itself is of great importance as they are collected in large volumes.

1.3 Overview of Handwritten Character Recognition

By handwritten character recognition (HCR) one means the recognition of single and unconstrained hand drawn characters. HCR is not as simple task as it might appear, since even the eyes of human beings make some 4% of mistakes when reading in the absence of context [9]. Errors in reading handwritten characters are caused by infinite variations of shapes resulting from the writing habit, style, education, region of origin, social environment, mood, health and other conditions of the writer as well as other factors such as the writing instrument, writing surface, scanning methods and most prominently, the machine's character

recognition algorithms [9]. In order to cope up with the variability of handwriting, hand print models are designed in earlier systems, which allow people to write in boxes with a guideline of how to write each alphabet. In unconstrained system people could write the way they normally did and characters need not have to be written as in specified models. The research on offline handwriting recognition aims at processing of images of paper documents written in different scripts and varying writing styles. Hence HCR problem can be considered as the first step toward the solution of handwriting recognition. It is a challenging task to develop a HCR system that maintain a very high recognition accuracy considering the variability of human writing, even though this is a task that human perform easily and reliably.

1.4 Motivation

It is always fascinating to make computer do preliminary functions of humans such as reading, writing, seeing things etc. But in spite of intensive research for more than five decades, the reading skill of the computer is still far behind that of human beings and most character recognition systems cannot read degraded documents and handwritten characters or words [8]. Demands on handwriting recognition have increased because large amounts of data were written by hand and they had to be entered into the computer for processing.

Many promising research results were reported in the area of handwritten character recognition for languages like English [10, 11], Chinese [12, 13], Japanese [14] and Arabic [15-17]. In Indian scenario, only a few works could be traced and most of the pieces of work deal with Devanagari and Bangla, the two most popular scripts in India [18]. Earlier and isolated attempts on handwritten character recognition in Indian scripts were made by Sethi and Chatterji [19] in Devanagari and Chinnuswamy and Krishnamoorthy [20] in Tamil characters. But, in recent times, research toward the recognition in Indian scripts is getting increased attention and approaches have been proposed toward the recognition of Indian numerals, characters and words in major Indian scripts [21]. Automatic recognition of handwritten characters is getting more importance nowadays. It is more significant in the context of e-governance. Furthermore, the government policy is supporting regional languages and it is instructed that official transactions should be in regional languages.

Malayalam is one among the twenty two scheduled languages of India with official language status in the state of Kerala and union territories of Lakshadweep and Puducherry. It is a classical language with rich literary tradition spoken by 33 million people. Therefore, the automatic interpretation of written Malayalam would have widespread benefits. Works on Malayalam started recently [22] and existing studies are limited to just basic characters of the language. A reliable system that

could recognize all the symbols in the modern Malayalam script is not attempted yet. Despite the huge efforts dedicated to handwriting recognition, it is still hard to find out the best feature extraction method available today. Moreover, recognition of unconstrained characters is more complex because the writing style is not known in advance. The problem is very relevant because the real application environment does not give any clue about the style of writing and the recognition system has to find out and manage different writing styles.

1.5 Problem Definition

To develop an efficient offline recognition system for unconstrained isolated handwritten Malayalam characters consisting of vowels, consonants, pure consonants, vowel signs, consonant signs and compound characters in a writer independent environment.

1.6 Challenges

- Enormously large character set: Large numbers of character symbols are present in Indian scripts compared to the symbols used to write European languages.
- Large number of similar or confusing characters: Several similar characters exist in Malayalam script. It is difficult to recognize them when isolated from context.
- Complex character shapes of compound characters in Malayalam script: The shapes of the compound characters are more complex

than basic characters. The inherent curved shape imposes another challenge.

- Non availability of benchmark databases are an additional hindrance for effective research.
- Extreme variability in collected samples due to writing style of each individual.

1.7 Objectives

- To set up a benchmark database for handwritten Malayalam characters that represents all the symbols in the modern script.
- To develop salient features suitable to accommodate the variability and complexity of handwritten characters.
- To identify the best classifier for the current problem that contains a large number of character classes.
- To find out an efficient method for separating similar characters in the script.
- To set up a very high recognition accuracy using two-stage classification strategy.

1.8 Framework of the Proposed System

The proposed HCR system addresses the variability of handwritten patterns by designing effective feature descriptors and powerful

classifiers. Besides the variability in handwriting, the variability in shape, size and complexity of characters need to be kept in mind while designing a recognition system. Fig. 1.1 gives the overall architecture of the proposed system. The proposed system involves tasks such as digitization, preprocessing, feature extraction, feature reduction and classification. Most of the steps in this system need to be optimized to obtain the best possible performance. The choice of preprocessing method affects feature extraction method and the subsequent classification process. We will take a quick look at each module and explore further details in the coming chapters.

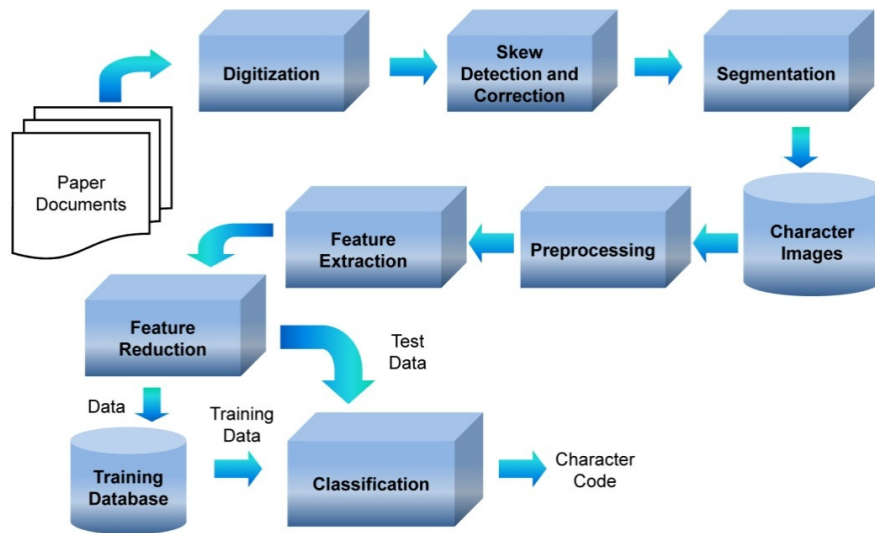


Fig. 1.1 Overall architecture of the proposed system

1.8.1 Digitization

One of the most challenging aspects of offline handwritten character recognition is finding a good database that well represents a wide variety of handwriting styles which contains the most important classes in the language. For this purpose, we have gathered data from different persons of various age groups and educational levels without imposing any constraints such as the style of writing, type of pen, colour of ink, thickness of lines, way of writing etc.

The paper documents containing unconstrained handwritten characters are digitized through flat-bed scanner by setting 300 dpi. Using these digitized documents, a benchmark database of totally unconstrained Malayalam handwritten samples is created for research purpose. The database contains 18,000 handwritten character images belonging to 90 character classes.

Due to the evolutionary changes in both hardware and software, today, even cell phones are equipped with camera that can capture various handwritten documents[2]. To cover this change, a smaller subset of the paper documents is also digitized through cell phone camera. This database contains 741 handwritten character images belonging to 25 character classes.

1.8.2 Preprocessing

Once the character samples are created, it undergoes preprocessing steps to enhance the quality of the character patterns. A series of preprocessing steps such as smoothing and noise removal, normalization, binarization, thinning, contour extraction etc. are required to make the character image ready for feature extraction. Some feature extraction algorithm can work directly on grayscale images while some work only on binary images. The binary image can also be represented in its contour form or skeleton form. Using appropriate preprocessing method, we have represented a character image, suitable for feature extraction.

1.8.3 Feature Extraction

Feature extraction method is important for any character recognition system to achieve high recognition performance. Character recognition techniques are generally classified as template based or feature based approach [18]. In the template based approach, the test pattern is directly superimposed on the ideal template pattern and the degree of correlation between the two is used as the decision factor. Due to the variability of human writing, template based approaches are not suitable for handwriting recognition and instead feature based approaches are used [23]. In the feature based approach, we have extracted features from spatial domain and from transform domain. Some feature extraction algorithms work directly on gray scale images, while some work only on

binary images. So we have identified which representation is more suitable for extracting features in each domain. For efficient discrimination of character patterns, we have developed a new feature descriptor based on image gradients.

1.8.4 Feature Reduction

With high dimensionality of the feature, the process of character classification becomes cumbersome. Hence there is a need to reduce feature dimension to improve performance as well as computational efficiency without loss of relevant information. Considering the above mentioned factors, we are able to reduce the dimensionality of the proposed feature without affecting the performance using Principal Component Analysis (PCA) technique. PCA reduces the dimensionality by transforming the data set to a new set of variables, called principal components. These components are uncorrelated and ordered so that the first few retain most of the variation present in the original set of variables.

1.8.5 Classification

The final step of classification is the process of finding the unknown class labels of the test data. The classifiers such as k -Nearest Neighbour, Support Vector Machines (SVM) and Extreme Learning Machines (ELM) are used for this purpose. K-Nearest Neighbour classification is one of the most fundamental and simple classification

methods and should be one of the first choices for a classification study when there is little or no prior knowledge about the distribution of the data. We have experimented with values of $k = 1, 3, 5, 7, 9$ and 15 .

Support Vector Machines (SVMs) are one of the most robust and powerful classifier for pattern recognition and are considered to be the state-of-the-art tool for linear and non-linear classification. For our classification model, as our data is non-linearly separable, two different kernels such as Radial Basis Function (RBF) kernel or Polynomial kernel is used to map the input space to a higher dimensional feature space so as to construct a linear decision boundary in the transformed space.

More recently, a new classifier, namely, Extreme Learning Machine (ELM) [24], is available for training of single layer feed forward neural network. ELM randomly chooses input weights and analytically determines the output weights of the network. In theory, this algorithm tends to provide a good generalization performance with much shorter learning time. Two variants of ELM such as optimization based ELM and ELM without any regularization factor are used in this work.

1.8.6 Classification in Two Stages

During analysis of results, we have identified that most of the errors are due to the presence of similar character shapes. Several such pairs exist in Malayalam. The frequency of misclassification among these similar characters is very high. The problem increases when the number

of classes increases. To cope up with this problem, we have designed an efficient two-stage classification approach. A two-stage classification approach has several advantages: First, it makes the number of classes in each stage small and it significantly reduces the error of misclassification of similarly shaped characters at the first stage. During the first stage of the two-stage classification approach, we have used the feature extraction algorithm proven to perform the best in the single stage classification. In the second stage, any type of features or any type of classifiers could be used. To resolve classification ambiguities, the possibilities of classification decisions are made within a group of commonly misclassified classes. Considering these facts, we have designed new classifiers for each group with specific features to handle each class separately. This approach improves recognition accuracy by reducing misclassification.

1.9 Thesis Contributions

- Created a benchmark database of 18,000 unconstrained handwritten Malayalam character images belonging to 90 character symbols.
- A novel attempt is made to create a database of Malayalam handwritten characters through mobile phone camera.
- First exclusive work on Malayalam compound characters, pure-consonants and vowel-consonant signs.

- Introduced Local Binary Pattern (LBP) based features for character recognition.
- Developed a novel feature descriptor based on image gradients.
- Introduced optimization based Extreme Learning Machine classifier for the present problem.
- Given a comparative study of performance of different algorithms with different classifiers over the created benchmark database.
- Introduced a grouping scheme coupled with two stage classification and obtained a novel benchmark recognition accuracy using specific features for discriminating similar handwritten patterns.

1.10 Conclusion

HCR is a special branch of pattern recognition devoted to the recognition of single isolated hand drawn characters. A HCR system involves tasks such as preprocessing, feature extraction and classification. The purpose of preprocessing is to discard irrelevant information from the input image. Preprocessing consists of smoothing and noise removal, normalization, binarization, thinning and contour extraction. The second step is feature extraction. The purpose of this step is to extract important information from images and represent this information in terms of

features. The last step is classification. In this step, the extracted features are mapped to different classes for identifying the characters.

In the proposed work, we have developed an efficient system for the recognition of unconstrained handwritten Malayalam characters consisting of vowels, consonants, pure consonants, vowel signs, consonant signs and compound characters. Machine recognition of handwritten characters are itself challenging as extreme variability is observed in human writing. Besides this, the shapes of compound characters are complex than its basic constituents. The number of classes used in this system is 90. This large character set with similar and confusing characters along with the inbuilt variability of handwritten patterns poses further challenges to the recognition system.

Overcoming these challenges, we have developed a recognition system with excellent classification performance by extracting salient features suitable to accommodate the variability and complexity of handwritten characters and also by designing a two-stage classifier for separating similar and confusing patterns.

Chapter 2

LITERATURE SURVEY

Contents

- 2.1 Introduction
- 2.2 Historical Background
- 2.3 Developments of HCR in the Global Scenario
- 2.4 Developments of HCR in the Indian Scenario
- 2.6 Summary of Developments of HCR in Indian Scenario
- 2.7 Commercial Character Recognition Softwares
- 2.8 Conclusion

2.1 Introduction

Writing, which has been the most natural way of collecting, storing and sending information through decades, now serves not only for communication among humans but also aims to serve for communication between humans and machines [25]. The character recognition system evolves through generations with the ultimate goal of making computer read the text with the same fluency as that of humans. The general framework of character recognition involves tasks such as digitization, preprocessing and segmentation, representation of character pattern, feature extraction, feature reduction and classification. The steps required depend on the techniques incorporated in the recognition.

This survey is focused on offline handwritten character recognition research and is conducted to bring out the present status by discussing major character recognition methodologies evolved over times. Given the vast number of papers published on character recognition every year, it is not possible to include all the available works in this survey. Instead, we tried to include a representative selection to illustrate the works starting from older classic papers from 1970s to the most relevant papers recently published up to the year 2013. The survey starts with discussion on the historical background of character recognition as depicted in Section 2.2; followed by advancements in the global scenario in Section 2.3 and then in Indian scenario with special focus to Malayalam in Section 2.4. Summary of developments in Indian scenario is provided in Section 2.5. This review also covers commercially available character recognition softwares in printed and handwritten domain in various languages in Section 2.6 and Section 2.7 concludes the chapter.

2.2 Historical Background

The origin of character recognition can be found in 1870 when Casey invented the retina scanner [9], which is an image transmission system using a mosaic of photocells. The first successful attempts were made by the Russian scientist Tyurin in 1900 to develop an aid for visually handicapped [26]. The modern version of optical character recognition appeared in the middle of 1940s with the development of the digital computer. The earlier work mainly concentrated upon machine

printed characters. Later, hand print models are designed so that people could write within boxes in specified shapes and recognition of hand printed characters or numerals were initiated [6]. These approaches generally used template matching in which the image of an unknown character is matched with a set of previously stored images. Matching techniques are based on the similarity degree between two vectors in feature space. These techniques can be categorized in three classes: direct matching [27], elastic and deformable matching [28] and matching by relaxation. Later, structural methods [29] were employed in character recognition along with statistical methods. These methods are concerned with statistical decision functions and a set of optimal criteria. With the advancement of information technology such as digital computers, scanners, cameras, the real progress of character recognition systems is achieved after 1990. Modern methodologies such as Neural Networks (NN) [30, 31], Hidden Markov Models (HMM) [32, 33], fuzzy set reasoning have developed. Fuzzy set reasoning employs fuzzy set elements like fuzzy graphs [34] and fuzzy rules [35] to describe the similarities between features. Since then intensive research has been carried out in this field and vast number of papers and books are being published [7]. The methodologies of character recognition have also changed from basic techniques for recognition of machine printed numerals and limited number of characters as in Latin to wide variety of hand printed characters of scripts like Chinese and Japanese.

2.3 Developments of HCR in the Global Scenario

Most character recognition systems are script specific and are designed to read characters written in one particular script only. Script is defined as the graphic form of a writing system used to write statements expressible in the language. A script may be used by only one language or may be shared by many languages [36]. According to Ghosh et al. [37], the major scripts of the world are Chinese, Japanese, Korean, Arabic, Hebrew, Latin, Cyrillic and the Brahmic family of Indian scripts as described in Fig. 2.1.

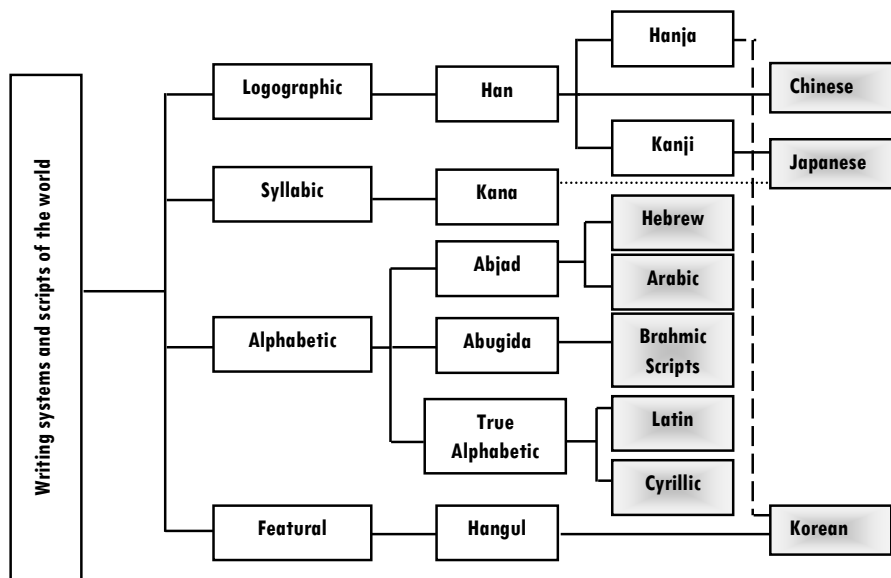


Fig. 2.1 Classification of writing systems and languages of the present world

2.3.1 Benchmark Databases for HCR Research

To encourage research in offline handwritings, a number of benchmark databases are created by various research groups. MNIST database [38] consists of 70000 isolated and labelled handwritten digits. It is divided into a training set of 60000 and a test set of 10000 digits. CEDAR database created by Centre of Excellence for Document Analysis and Recognition, SUNY, Buffalo, contains handwritten words and ZIP codes in high resolution gray scale as well as binary samples of 52 English handwritten characters and 10 handwritten digits. The CENPARMI digit database was released by Centre for Pattern Recognition and Machine Intelligence, Concordia University. It contains 6000 digit images where 4000 images are specified for training and the remaining 2000 images are specified for testing.

ETL9B database was created by Electro Technical Laboratory of Japan. It contains 200 samples for each of 3036 categories, 2965 Chinese and 71 Japanese characters. HCL2000 database [39] was collected by Beijing University of Posts and Telecommunications for China-863 project and it contains 3755 frequently used simplified Chinese characters. CASIA database [40] were built by the National Laboratory of Pattern Recognition, Institute of Automation of Chinese Academy of Sciences (CASIA) for unconstrained online and offline Chinese handwriting recognition. The handwritten samples were produced by 1,020 writers using Anoto pen on papers, such that both online and offline

data were obtained. The samples include both isolated characters and handwritten texts.

2.3.2 Latin and Greek

Latin is an alphabetic script where an alphabet is a set of characters representing phonemes of a spoken language. Latin script, also known as, Roman script, is used to write languages such as English, Italian, French, German, Portuguese, Spanish and some other European languages [37]. Other scripts following alphabetic system include Greek, Cyrillic, and Armenian etc.

A comprehensive survey on offline and online handwriting recognition in Latin script was available [1]. Developments in offline handwritings along with the strengths and weaknesses of each techniques up to the year 2000 was covered in the survey of Arica and Yarman-Vural [25]. Advances in handwriting recognition using MNIST, CEDAR and CENPARMI databases up to the year 2004 was described in [41]. Some of the notable works in this domain are provided in the following paragraphs:

Wunsch and Laine [42] used wavelet features extracted from contour of the handwritten characters for classification using neural networks. Lee et al. [43] used wavelet features extracted from handwritten numerals and classified it using multilayer cluster neural network. Chen et al. [44] developed multi wavelet descriptor from the contour of

handwritten numerals using neural network. Bellili et al. [45] introduced a hybrid Multilayer Perceptron (MLP) and Support Vector Machine (SVM) classifiers for handwritten digit recognition. Liu et al. [46] presented the results of handwritten digit recognition on well-known image databases such as CENPARMI, CEDAR, and MNIST using chain code feature, gradient feature, profile structure feature, and peripheral direction contributivity. The gradient feature is extracted from either binary image or gray scale image. The classifiers include the k -NN classifier, three neural classifiers, a Learning Vector Quantization (LVQ) classifier, a discriminative learning quadratic discriminant function (DLQDF) classifier, and two SVMs. The chain code feature and the gradient feature show advantage over other features and SVM with radial basis function kernel gives the highest accuracy in most cases.

Zhang et al. [47] presented a novel cascade ensemble classifier system for the recognition of handwritten digits. This system aims at attaining a very high recognition rate and a very high reliability at the same time. Seven sets of discriminative features and three sets of random hybrid features are extracted and used in the different layers of the cascade recognition system. The novel Gating Networks (GNs) are used to congregate the confidence values of three parallel artificial neural network classifiers. The weights of the GNs are trained by the Genetic Algorithms (GAs) to achieve the overall optimal performance. Experiments conducted on the MNIST handwritten numeral database are

shown with encouraging results: a high reliability of 99.96% with minimal rejection, or a 99.59% correct recognition rate without rejection in the last cascade layer.

Recognition accuracy is always an important factor in evaluating the methodology and has reached beyond 99% in benchmark numeral databases such as MNIST and CENPARMI but it is still impossible to obtain 100%. Hence, research in this domain is focusing towards the recognition of more reliable systems.

Kavallieratou et al. [48] presented a handwritten Greek character recognition system based on structural characteristics, histograms and profiles. The horizontal and vertical histograms are used, in combination with radial histogram, out-in radial and in-out radial profiles for representing 32×32 matrices of characters, as 280 dimensional feature vectors. The K-means algorithm is used for the classification of these vectors. Detailed experiments performed in NIST and Greek databases gave accuracy results that vary from 72.8% to 98.8% depending on the difficulty of the database and the character category.

Vamvakas et al. [49] proposed a feature extraction method which is based on recursive subdivisions of the character image so as to result in sub-images of equal foreground pixels and calculation of division point. This recursive subdivision is done so that the sub-images have equal foreground pixels. Two handwritten character database such as CEDAR

and Greek database as well as two handwritten digit databases (MNIST and CEDAR) were used to demonstrate the effectiveness of the proposed technique. A two stage hierarchical approach is used to classify characters using SVM classifier. The recognition result achieved in CEDAR database is 94.73% and MNIST database is 99.03% and Greek database is 95.63%

2.3.3 Chinese

Chinese is spoken by about 1.3 billion people mainly in China, Taiwan, Singapore and other parts of Southeast Asia [36]. It is written in logographic script, where a logogram refers to a symbol that graphically represents a complete word.

In 1966, Casey and Nagy [50] presented one of the first attempts at machine printed Chinese character recognition. In late 1970s, Agui and Nagahashi [51] suggested a description method for hand printed Chinese character recognition. Directional features are widely used and found to be effective in Chinese character recognition since it catch the stroke direction pattern which is an important characteristic of Chinese character. Among directional features, gradient feature outperforms various other directional features [52]. In the work by Dong et al. [12], the authors describe several techniques such as enhanced nonlinear normalization, feature extraction and tuning kernel parameters of SVM for improving the classification accuracy. The recognition system has achieved a high recognition rate of 99% on

ETL9B, a handwritten Chinese character database. Ding et al. [53] proposed a method using Gabor features as it is suitable for extracting the joint information in two-dimensional spatial and frequency domain. Gao and Liu [54] proposed Linear Discriminant Analysis (LDA) based compound distances for discriminating similar characters. The LDA-based method is an extension of Compound Mahalanobis Function (CMF), which calculates a complementary distance on a one-dimensional subspace (discriminant vector) for discriminating two classes and combines this complementary distance with a baseline quadratic classifier. For evaluation the ETL9B and CASIA databases are used with the Modified Quadratic Discriminant Function (MQDF) as baseline classifier. The results demonstrate the superiority of LDA-based method over the CMF and the superiority of discriminant vector learning from high-dimensional feature spaces. Compared to the MQDF, the proposed method reduces the error rates by factors of over 26%.

Zhang et al. [55] proposed a modified Scale Invariant Feature Transform (SIFT) based feature for offline handwritten Chinese character recognition. In their approach, global elastic meshing was first constructed and then the related gradient code of each sub-region was classified using MQDF classifier. The recognition rate of 97.87% was obtained on HCL2000 database. Leung and Leung [56] proposed a “critical region analysis” technique which highlights the critical regions that distinguish one character from another similar character. The critical

regions are identified automatically based on the output of the Fisher's discriminant. Additional features are extracted from these regions and contribute to the recognition process. By incorporating this technique into the character recognition system, a high recognition rate of 99.53% on the ETL-9B database was obtained.

Recently, Ni, Jiang et al. [57] presented a method of radical extraction by using radical cascade classifier, which detects radicals within characters. Haar-like features are applied in the cascade classifier, as they can absorb radical distortion and guarantee the high speed of radical detection. Two methods of radical detection are proposed according to the characteristics of Chinese characters. He Zhong et al. [58] proposed a new method for handwritten Chinese character recognition based on a combination of Independent Component Analysis (ICA) and SVM. They extracted independent basis images of handwritten Chinese characters and the projection vector by using fast ICA algorithm. For recognition, a two stage classification strategy is used. Evaluation is done on HCL2000 database and achieved a recognition accuracy of 99.87%

2.3.4 Japanese

Japanese script uses a mix of logographic Kanji and syllabic Kana [37]. In a syllable system, every written symbol represents a phonetic sound or syllable. Japanese writing system has three different character sets, namely, Hiragana, Katakana and Kanji.

In the work by Kimura et al. [59], three types of nonlinear normalization for the preprocessing, the discriminant analysis and the principal component analysis for the feature extraction, the minimum distance classifiers and the linear classifier for the high speed pre-classification, and modified Bayes classifier and subspace method for the robust main classification was experimentally compared. The performance of the recognition algorithm is fully tested using the ETL-9B character database and the recognition accuracy obtained was 99.15%

Kato et al. [60] presents a precise system for handwritten Chinese and Japanese character recognition. Before extracting Directional Element Feature (DEF) from each character image, Transformation based on Partial Inclination Detection (TPID) is used to reduce undesired effects of degraded images. In the recognition process, City Block Distance with Deviation (CBDD) and Asymmetric Mahalanobis Distance (AMD) are proposed for rough classification and fine classification. With this recognition system, the experimental result of the database ETL9B reaches to 99.42%

2.3.5 Arabic

Arabic script is used to write Arabic, Persian (Farsi) or Urdu. Persian is a writing script based on Arabic script and Urdu is a modification of Persian script.

In 1996, Amin [15] proposed a technique for the recognition of hand-printed Arabic characters using ANNs. For this, skeleton of the character image is traced from right to left in order to build a graph and primitives such as straight lines, curves and loops are extracted from the graph. Finally, a five layer ANN is used for the character classification and the correct recognition rate obtained was 92%. Later he also conducted a survey on this script [16].

Mowlaei et al. [61] developed a system for recognition of handwritten Farsi/Arabic characters and numerals. They used Haar wavelet for feature extraction in this system. The extracted features are used as training inputs to a feed forward neural network using the back propagation learning rule. They categorize 32 characters in Farsi language to 8 different classes in which characters of each class are very similar to each others. Along with these 8 character classes, eight digit classes are also considered. This system yields the classification rates of 92.33% and 91.81% for these 8 classes of handwritten Farsi characters and numerals respectively.

Mozaffari et al. [62] proposed fractal codes and Haar Wavelet Transform for the recognition isolated handwritten Farsi/Arabic characters and numerals. Fractal codes represent affine transformations which when iteratively applied to the range-domain pairs in an arbitrary initial image, the result is close to the given image. Each fractal code consists of six parameters such as corresponding domain coordinates for

each range block, brightness offset and an affine transformation. This method is robust to scale and frame size changes. For classification, the discriminating power of SVM is utilized.

Liu and Suen [63] presented a recognition scheme for handwritten Bangla and Farsi numerals of binary and gray scale images. For recognition on gray scale images, they proposed a process with proper image preprocessing and feature extraction. In experiments on three databases, ISI Bangla numerals, CENPARMI Farsi numerals, and IFHCDB Farsi numerals, the highest test accuracies were 99.40%, 99.16% and 99.73% respectively using DLQDF classifier.

Recently, Shayegan and Chan [64] extracted a set of features, by employing one and two-dimensional spectrum diagrams. In the experiment, they obtained 95.70% recognition accuracy.

2.3.6 Korean

Korean script is formed by mixing logographic Hanja with featural Hangul. Consequently, Korean script is relatively less complex compared to Chinese and Japanese [37]. Korean is a language spoken by about 63 million people in South Korea, North Korea, China, Japan, Uzbekistan, Kazakhstan and Russia.

Kim and Kim [65] presented a recognition system for handwritten Hangul (Korean) characters using Hierarchical Random Graph (HRG). In the HRG, the bottom layer is constructed with extended random graphs to

describe various strokes, while the next upper layers are constructed with random graphs to model spatial and structural relationships between strokes and between sub-characters. Model parameters of the hierarchical graph have been estimated automatically from the training data by Expectation Maximization (EM) algorithm and embedded training.

Kang and Kim [66] proposed a stochastic modelling scheme by which strokes as well as relationships are represented by utilizing the hierarchical characteristics of target characters. Based on the proposed scheme, a handwritten Hangul (Korean) character recognition system is developed. The effectiveness of the proposed scheme is shown through experimental results conducted on a public database. They proposed another system [67] in which the difficulties of the learning due to the high order of the probability distribution are overcome by factorizing and approximating the probability distribution by a set of lower-order probability distributions.

Park et al. [68], performed a comprehensive evaluation of different statistical methods. They implemented 15 character normalization methods, five feature extraction methods and four classification methods and evaluated their performance on two public Hangul databases.

2.4 Developments of HCR in the Indian Scenario

Ministry of Communication and Information Technology, Government of India, has initiated Technology Development for Indian

Languages (TDIL) programme and thirteen Resource Centres for Indian Language Technology Solutions (RCILTS) have been established under this project. Initiatives have been taken for long term research for development of Machine Translation System, Optical Character Recognition, On-line Handwriting Recognition System, Cross-lingual Information Access and Speech Processing in Indian languages by this programme. Commercial systems for machine printed characters are developed for some Indian scripts namely Assamese, Bangla, Devnagari, Malayalam, Oriya, Tamil and Telugu, but no system is available for recognizing handwritten Indian manuscript till date. A short description about the works on the recognition of machine printed and handwritten Indian scripts including Bangla, Tamil, Telugu, Gurumukhi, Oriya, Gujarati, Kannada and Devnagari up to 2002 is provided in the survey of Pal and Chaudhuri [18].

2.4.1 Languages in India

India is a multi-lingual, multi-script country with twenty two scheduled languages, namely, Assamese, Bengali, Bodo, Dogri, Gujarati, Hindi, Kannada, Kashmiri, Konkani, Maithili, Malayalam, Manipuri (Meithei), Marathi, Nepali, Oriya, Punjabi, Sanskrit, Santali, Sindhi, Tamil, Telugu and Urdu [69]. Some of these languages have common scripts and some have their own script. Devnagari script used to write Hindi, Konkani, Kashmiri, Marathi, Nepali, Sanskrit, Bodo, Dogri, Maithili and Sindhi. Among this, Kashmiri is also written using Sharada

script and Perso-Arabic script. Assamese, Manipuri and Bengali languages are written using Bangla script while Punjabi language is written using Gurumukhi script. Santali is the language spoken by Santal, the largest tribal community in India. Even though it owns a script named Ol Chiki, created in 1920, Santali is also written using Oriya, Bangla, Devnagari and Latin scripts. Other languages such as Malayalam, Tamil, Telugu and Kannada have their own script. There are nine scripts which are considered as basic beside the script for Urdu and these are Devnagari, Bangla, Gurumukhi, Gujarati, Oriya, Kannada, Telugu, Tamil and Malayalam.

2.4.2 Characteristics of Indian Scripts

Most of the Indian scripts are originated from ancient Brahmi script through various transformations. They are phonetic in nature and hence writing maps sounds of alphabets to specific shapes. All these languages, except Urdu, are written from left to right. The basic characters comprises of vowels and consonants. Two or more basic characters are combined to form compound characters. The Indic scripts do not have a prominent writing style and the notion of upper case and lower case characters are not present. Urdu is an Indo-Aryan language and it is related to Arabic, Persian and Hindi. A speaker of Hindi can understand spoken Urdu but may not be able to read written Urdu because Urdu is written in Nastaliq script from right to left and uses a modified set of Persian alphabets and Arabic alphabets. Like Arabic, the shapes of

characters in Urdu are determined by their position in the word. The same letter can have different shapes when written in isolation, in the initial part, in the middle, or as the final letter of a word [70]. Evolution of Indian scripts from ancient Brahmi script is depicted in Fig. 2.2 [37].

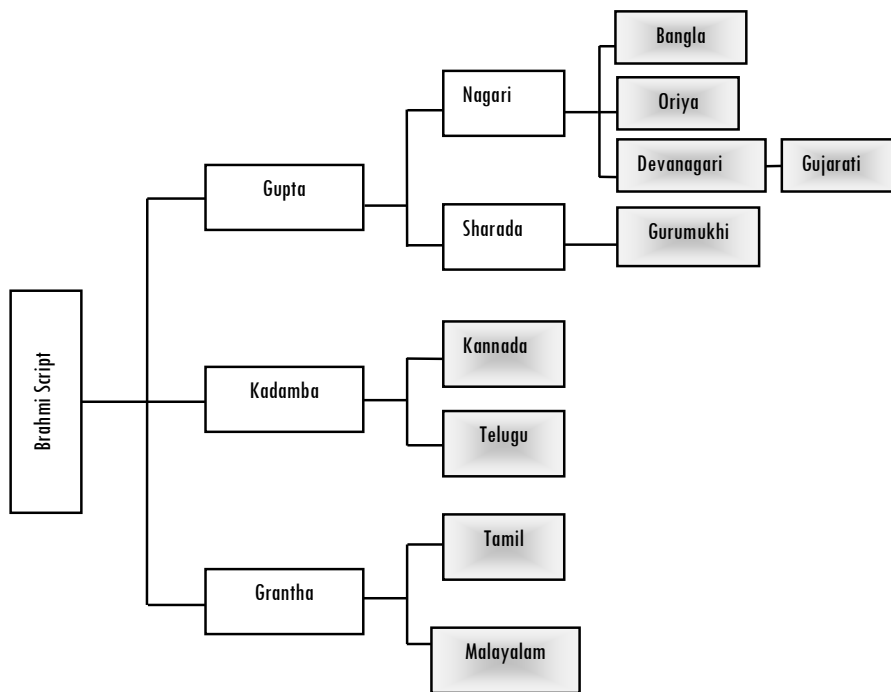


Fig. 2.2 The Brahmic family of scripts used in India

2.4.3 Databases for Indian HCR

For successful research, creation of benchmark database is essential. Only few such databases exist for Indian scripts. One such

database was developed by Indian Statistical Institute, Kolkata [71] which contains offline handwritten database of numerals, basic characters, vowel modifiers and compound characters of Bangla, numerals and basic characters of Devnagari and numerals of Oriya script. HP Labs, India developed a database, called hpl-tamil-iso-char, of handwritten samples of 156 different Tamil characters. The data was collected using HP Tablet PCs and is in standard UNIPEN format. However, an offline version of the same set is available for the research developments [72]. For Urdu, a comprehensive database developed by CENPARMI which contains dates, isolated digits, numerical strings, isolated letters, a collection of 57 words and a collection of special symbols is available for research.

2.4.4 Developments on North Indian Scripts

North Indian languages, except Urdu are derived from Nagari and Sharada scripts of ancient Brahmi through various transformations.

2.4.4.1 Devnagari

Devnagari script has 13 vowels and 34 consonants. Apart from the vowels and consonants, there are compound characters in Devnagari, which are formed by combining two or more basic characters. The shape of a compound character is usually more complex than its constituent characters. A vowel following a consonant may take a modified shape depending on whether the vowel is placed to the left, right, top or bottom of the consonant. A survey on offline recognition of Devnagari script has

been conducted by Jayadevan et al. [73]. A set of notable and recent works on handwritten numeral/character recognition are described in this section.

The first research report on Devnagari characters was reported by Sethi and Chatterjee [19] in 1977. The authors have presented a constrained hand printed Devnagari character recognition in which the presence or absence of four basic primitives, namely, horizontal and vertical line segment, left and right slant and their interconnections are used and recognition is done with the help of decision trees. Even though, the works started earlier, the research became active only recently. Sharma et al. [74] proposed a quadratic classifier based scheme for the recognition of offline Devnagari handwritten characters. The features used in the classifier are obtained from the directional chain code information of the contour points of the characters. They obtained 98.86% and 80.36% recognition accuracy on Devnagari numerals and characters, respectively. Hanmandlu and Murthy [75] presented a recognition scheme for handwritten Hindi and English numerals by representing them in the form of exponential membership functions which serve as a fuzzy model. These exponential membership functions fitted to the fuzzy sets derived from normalized distances obtained using the box approach. The overall recognition rate is found to be 95% for Hindi numerals and 98.4% for English numerals from CEDAR database.

Arora et al. [76] presented a two stage classification approach for handwritten Devnagari characters. The first stage is using structural properties like ‘shirorekha’, ‘spine’ in character and second stage exploits some intersection features of characters which are fed to a feed forward neural network. They designed a differential distance based technique to find a near straight line for ‘shirorekha’ and ‘spine’. The recognition accuracy obtained 89.12%. The same authors [77] presented a system for recognition of basic characters using four feature extraction techniques namely, intersection, shadow feature, chain code histogram and straight line fitting features. Weighted majority voting technique is used for combining the classification decision and obtained a recognition rate of 92.80%.

Pal et al. [78] proposed recognition of offline handwritten numerals of six popular Indian scripts such as Devnagari, Bangla, Telugu, Oriya, Kannada and Tamil. The features used in the classifier are obtained from the directional information of the numerals. For feature computation, the bounding box of a numeral is segmented into blocks and the directional features are computed in each of the blocks. These blocks are then down sampled by a Gaussian filter and the features obtained from the down sampled blocks are fed to a modified quadratic classifier for recognition. A five-fold cross validation technique has been used for result computation and obtained 99.56%, 98.99%, 99.37%, 98.40%, 98.71% and 98.51% accuracy for Devnagari, Bangla, Telugu, Oriya, Kannada and Tamil scripts, respectively.

In [79], Pal et al. presented a system for Devnagari handwritten character recognition in which the features used are based on the direction of the gradient obtained using Roberts filter. The direction of the gradient is quantized into 32 directions and the strength of the gradient accumulated in each of the quantized direction is down sampled using Gaussian filter. A MQDF classifier is applied on these features for recognition and obtained a recognition accuracy of 94.24%. In [80], the same authors presented a comparative study using different classifiers and different sets of features. Projection distance, subspace method, linear discriminant function, SVMs, modified quadratic discriminant function, mirror image learning, Euclidean distance, k-NN, modified projection distance, compound projection distance and compound modified quadratic discriminant function are used as different classifiers. Feature sets used in the classifiers are computed based on curvature and gradient information obtained from binary as well as gray scale images.

Mukherji and Rege [81] proposed a new shape based technique for recognition of isolated handwritten Devnagari characters. The thinned character is segmented into segments (strokes), using basic structural features like endpoint, cross point, junction points and adaptive thinning algorithm. The segments of characters are coded using average compressed direction code algorithm. The segment shapes are classified as left curve, right curve, horizontal stroke, vertical stroke, slanted lines etc. The knowledge of script grammar is applied to identify the character

using shapes of strokes, mean row and column co-ordinates, relative strength, straightness and circularity. Their location in the image frame is based on fuzzy classification. Characters are pre-classified using a tree classifier. Subsequently unordered stroke classification based on mean stroke features is used for final classification and recognition of characters. The average accuracy of recognition of the proposed system is 86.4%. Recently, Jangid [82] proposed a methodology which relies on a three feature extraction techniques. The first technique is based on recursive subdivisions of the character image so that the resulting sub-images have approximately equal number of foreground pixels. Second technique is based on the zone density of the pixel and third is based on the directional distribution of neighbouring background pixels to foreground pixels.

Recognition of handwritten compound characters could be traced only in the work of Shelke and Apte [83] in Marathi language. Although, Marathi is written in Devnagari script, compound characters are more frequent in Marathi language. They have employed a two-stage scheme in which the initial stage is based upon the structural features and the final stage is based on wavelet transform. The average recognition rate was found to be 96.23%.

2.4.4.2 Bangla

Bangla is the third most popular language in India and the national language of Bangladesh and is the fifth popular language in the world. Bangla script is evolved from the ancient Brahmi script through various transformations [36]. The script is syllabic. The text is written using consonants and vowels. The direction of writing is from left to right in horizontal lines.

For the recognition of handwritten Bangla numerals, Bhattacharya et al. [84] proposed a modified topology adaptive self-organizing neural network to extract a vector skeleton from the binary numeral image. Certain topological and structural features are used along with a hierarchical tree classifier to classify handwritten numerals into smaller subgroups. MLP is then employed to uniquely classify the numerals belonging to each subgroup. Recognition of Bangla numerals are also addressed in [63]. Rahman et al. [85] proposed a sub grouping scheme for basic characters considering ‘matra’, upper part of the character, disjoint section of character, vertical line and double vertical line. This multi-stage approach is applied to a small database collected in laboratory environment. Bhowmik et al. [86] proposed an MLP based recognition scheme using stroke features of Bangla basic characters. Bhattacharya et al. [87] proposed a scheme to recognize Bangla basic characters using shape based features. Preprocessing is done with restricted mean filter approach and classification is done with MLP.

Recognition of compound characters is addressed only in few works. Pal et al. [88] proposed gradient features for the recognition of Bangla compound characters using MQDF. Using 5-fold cross validation technique they obtained 85.90% accuracy. Bhowmik et al. [89] proposed an SVM based hierarchical classification scheme for recognition of handwritten Bangla basic characters. The number of classes evaluated is 45. A comparative study is made among MLP, RBF and SVM classifiers and SVM is found to outperform the other classifiers. A fusion scheme is proposed using these three classifiers. Three different two-stage hierarchical learning architecture are proposed using three grouping scheme. These groups are determined in two different ways based on the confusion matrix obtained from SVM classifier and neural gas algorithm. Disjoint and overlapped grouping schemes were evaluated and overlapped groups outperform other two hierarchical learning architecture.

Das et al. [90] proposed a recognition scheme for handwritten Bangla basic and compound characters using MLP and SVM. The proposed scheme recognizes handwritten characters of 93 classes; among them 50 are basic and the rest 43 are compound characters using features such as shadow and longest run. The recognition accuracy obtained was 79.25% and 80.51% using MLP and SVM respectively using three fold cross validation. The same authors [91] proposed a novel Genetic Algorithm (GA) and SVM based multistage recognition strategy to recognize handwritten Bangla compound characters. The developed

algorithm identifies optimal local discriminating regions in the second pass of the multistage approach, within each group of pattern classes identified by the first pass classifier. They have obtained an accuracy of 78.93% which is 2.83% more than the result achieved by single pass approach. Reza and Khan [92] combined Bangla basic characters, numerals and vowel modifiers together and proposed a recognition method using chain code and grouping with SVM. They obtained a recognition accuracy of 89.9% using 69 classes.

2.4.4.3 Gujarati

Gujarati script was adapted from Devnagari script. The script first appeared in printed form in an advertisement in 1797. Until the 19th century it was mainly used for writing letters and keeping accounts [36]. The script is syllabic in nature. Vowels can be written independently or using a variety of modifiers, which are written above, below, before or after a consonant. When two consonants are combined, special conjunct letters are formed. The mode of writing is from left to right.

We could trace only few works in Gujarati HCR. Prasad et al. [93] proposed a template-matching technique for the recognition of handwritten Gujarati characters, where a character is identified by analyzing its shape that distinguishes each character. Recognition accuracy specified was 71.66%. Desai [94] proposed a multi layered feed forward neural network for Gujarati handwritten numeral recognition.

Based on the structural behaviour of Gujarati characters, four different profile features are extracted from each numeral. Thinning and skew-correction are also done for preprocessing of handwritten numerals before their classification. This work has achieved approximately 82% of success rate for Gujarati handwritten digit identification.

2.4.4.4 Oriya

The Oriya script is being used to write Oriya language. Most of the characters in Oriya script are round shapes and it is due to the habit of writing on palm leaves [36]. Oriya is a syllabic alphabet. When vowels appear at the beginning of word, they are written independently. Other vowels are indicated with diacritics capable of appearing above, below or after the consonants. When two consonants come together in groups, special composite shapes are used. The direction of writing is from left to right in horizontal lines.

Roy et al. [95] proposed a scheme in which each Oriya handwritten numeral is segmented into a few blocks. The features used for recognition are based on the direction chain code histogram of the contour points of these blocks. Neural Network (NN) classifier and quadratic classifier are used separately for recognition and the results obtained from these two classifiers are compared. The authors obtained 90.38% and 94.81% recognition accuracy from NN and quadratic classifier with a rejection rate of about 1.84% and 1.31%, respectively.

A Hidden Markov Model (HMM) was proposed by Bhowmik et al. [96] for the recognition of handwritten Oriya numerals. A handwritten numeral is assumed to be a string of several shape primitives. One HMM is constructed for each numeral where the states of HMM are determined automatically based on a database of handwritten numeral images. To classify an unknown numeral image, its class conditional probability for each HMM is computed. The classification scheme has been tested on a large handwritten Oriya numeral database. The classification accuracy obtained was 95.89% and 90.50% for training and test sets respectively.

Most Oriya characters have rounded curve shapes at the upper part of the characters. Because of this shape, Pal et al. [97] presented a system for the recognition of Oriya handwritten characters using curvature features. To get the feature, at first, the input image is size normalized and segmented into 49×49 blocks. Curvature is then computed using bi-quadratic interpolation method and quantized into 3 levels according to concave, linear and convex regions. Next direction of gradient is quantized into 32 levels with $\pi/16$ intervals, and strength of the gradient is accumulated in each of the 32 directions and in each of the 3 curvature levels of every block. A spatial resolution is reduced to get 7×7 blocks and a directional resolution is reduced to get 8×8 directions. The dimension of feature was 1176, which is reduced 392 and obtained 94.60% accuracy.

Padhi and Senapati [98] designed a two stage recognition system with standard deviation and zone centroid average distance based feature. The characters are classified into four groups according to similarity of their shapes and features. For recognition, feed forward BPNN in two stages is used where the first stage classifies the characters into similar groups and in the second stage classifies individual characters.

2.4.4.5 Gurumukhi

Gurumukhi script is used to write Punjabi language. Guru Nanak, the first Sikh guru developed the Gurumukhi alphabet during the 16th century [36]. The name Gurumukhi means “from the mouth of the Guru”. Gurumukhi is a syllabic alphabet. The script is similar to Devnagari but simpler since compound characters are absent here [18]. When vowels appear at the beginning of a word, they are written independently. Other vowels are indicated with diacritics capable of appearing above, below or after the consonants. The direction of writing is from left to right in horizontal lines.

Garg and Verma [99] used structural features along with feed forward back propagation neural network. In [100], Siddharth et al. have used some statistical features like zonal density, projection histograms (horizontal, vertical and both diagonal) and distance profiles (from left, right, top and bottom sides). In addition, they have used background directional distribution features in a database of 200 samples of 35 basic

characters of Gurmukhi script collected from different writers. SVM, k-NN and PNN classifiers are used for classification. The performance comparison of features used in different combination with different classifiers is presented and analysed. The highest accuracy obtained is 95.04% using zone density and background distribution features with SVM classifier. Singh et al. [101] have used Gabor Filter based method for feature extraction. The highest accuracy obtained is 94.29% using 5-fold cross validation with SVM classifier for 35 basic characters.

2.4.4.6 Urdu

Urdu is an Indo-European language which originated in India. Written Urdu has been derived from the Persian alphabet, which itself has been derived from the Arabic alphabet. However, Urdu has more isolated letters than Arabic and Persian [102].

Yusuf and Haider [103] proposed recognition of handwritten Urdu digits using the novel descriptor for shape matching. Each digit is represented using a discrete set of n points sampled along its border. For each of these points, the shape context is a histogram of relative positions of the $n-1$ remaining points. They have taken the criterion of similarity between two instances as the weighted sum of cost of matching shape contexts and bending energy, which is the amount of work it takes to transform one instance to another and found that the technique is effective with zero percent error on the 28 test digits.

Haider and Yusuf [104] used prototype based object recognition which require measuring similarities between the test object and the prototype categories. When multiple instances per object are stored in the prototype set, this task becomes computationally expensive. To increase the efficiency, pruning techniques are used. In the work, the authors have presented a gradual pruning approach based on the dissimilarities between the test object and the objects in the prototype set. A mathematical expression has been derived analytically to save computational time. This approach is applied to the task of handwritten Urdu digit recognition using shape context. Compared to the classical and step pruning approaches, gradual pruning based method was found to be faster without compromising accuracy. Liu and Suen [63] presented a recognition scheme for handwritten IFHCDB Farsi (Urdu) numerals of binary and gray scale images as mentioned in Section 2.3.5 and obtained an accuracy of 99.73% using DLQDF classifier.

Pathan et al. [105] presents an approach for recognition of offline handwritten isolated Urdu character based on invariant moments. The Urdu letters were grouped into single component and multi-component characters. If letter is multi-component then secondary component were separated from primary component. SVM is adopted for classification for 46 character classes and position of secondary component (above, below and middle) is considered for recognition and overall performance rate was found to be 93.59%

2.4.5 Developments on South Indian Scripts

South Indian languages are derived from Kadamba and Grantha scripts of ancient Brahmi through various transformations. We have conducted a survey on offline recognition of south Indian scripts [69].

2.4.5.1 Tamil

Tamil is one of the oldest languages in India. The Tamil script has 10 numerals, 12 vowels, 18 consonants and five Grantha letters. The script, however, is syllabic and not alphabetic. The complete script, therefore, consists of 31 letters in their independent form, and an additional 216 combining letters representing every possible combination of a vowel and a consonant.

One of the early attempts in HCR was by Chinnuswamy and Krishnamoorthy in the year 1980 [20]. In that work, authors proposed a method to recognize hand printed Tamil characters using curves and strokes of characters as features. The input image is converted to labelled graph before extracting features and correlation coefficients are computed for recognition of the character. Later, in 1993, Paulpandian and Ganapathy [106] proposed a Hierarchical Neural Network (HNN) which can recognize handwritten Tamil characters independently of their position and size. Twelve character classes are used in the experiment. HNN is compared with and found to be superior to single neural network approach and the method of moments in conjunction with neural network.

Suresh et al. [107] attempts to use the fuzzy concept on handwritten Tamil characters to classify them as one among the prototype characters using distance from the frame and a suitable membership function. The prototype characters are categorized into two classes: one is considered as line characters/patterns and the other is arc patterns. The unknown input character is classified into one of these two classes first and then recognized to be one of the characters in that class. The algorithm is tested for about 250 samples for seven chosen Tamil characters and the success rate obtained varies from 88% to 100%. Hewavitharana and Fernando [108] recognizes 26 Tamil characters through a two-stage classification approach, which is a hybrid of structural and statistical techniques. In the first stage, an unknown character is pre-classified into one of the three groups: core, ascending and descending characters. Then, in the second stage, members of the pre-classified group are further analyzed using a statistical classifier for final recognition.

Pal et al. [109] proposed a quadratic classifier based scheme for the recognition of offline handwritten characters of three popular south Indian scripts: Kannada, Telugu, and Tamil. The features used are mainly obtained from gradient directional information. For feature computation, the bounding box of a character is segmented into blocks, and the directional features are computed in each block. These blocks are then down-sampled by a Gaussian filter. A five-fold cross validation technique was used for result computation, and obtained 90.34%, 90.90% and

96.73% accuracy rates from Kannada, Telugu, and Tamil characters, respectively, using 400 dimensional features.

Sudha and Ramaraj [110] proposed a scheme in which Fourier descriptor based features are extracted from character images. The system was trained using several different forms of handwriting provided by both male and female participants of different age groups using MLP with one hidden layer. Bhattacharya et al. [111] proposed a recognition system for Tamil characters using hpl-tamil-iso-char, the database of HP Labs, India. The system used a two stage recognition approach. The training samples are first grouped using K-means clustering using a count of transition from one pixel position into other. During the second stage MLP is used to classify each group using chain code histogram features of samples. The recognition accuracy obtained is 92.77% and 89.66% for training and testing sets. Shanthi and Duraiswamy [112] evaluated pixel density features in zones of sizes 2×2 to 8×8 pixels. The best result of 82.04% is obtained with 4×4 zone with 64 features. Subashini and Kodikara [113] investigated the effect of local Scale Invariant Feature Transform (SIFT) to classify twenty selected Tamil characters. In the proposed method each preprocessed character is represented by a set of local SIFT feature vectors. From a large set of SIFT descriptors, the key idea was to create a codebook for each character using K-means clustering algorithm and each character was recognized using k-NN with an average recognition accuracy of 87%.

2.4.5.2 Telugu

Telugu is the official language of the state of Andhra Pradesh. It is a syllabic language. Officially, there are 10 numerals, 18 vowels, 36 consonants and three dual symbols. When two consonants join together, compound characters are formed.

In 2009, Rajashekararadhya and Ranjan [114] proposed an offline handwritten numeral recognition technique for four south Indian languages such as Kannada, Telugu, Tamil and Malayalam. In this work they suggested a feature extraction technique based on zone and image centroid. They used two different classifiers, k-NN and BPNN, to achieve 99% accuracy for Kannada and Telugu, 96% for Tamil and 95% for Malayalam.

Pradhan and Negi [115] used approximate matching of the string for classification of 43 Telugu characters from the basic Telugu characters. During the preprocessing an input character image is transformed into a skeletonized image and discrete curves are found using a 3×3 pixel region. This discrete curve patterns are encoded. The encoding of several such sequence numbers for the thinned character constructs a pattern string. Approximate string matching is used to compare the encoded pattern string from a template character with the pattern string obtained from the input character. The proposed approach has recognised all the test characters correctly.

In a recent work, Soman et al. [116] combined the strengths of four different pattern analysis techniques such as CNN, PCA, SVMs and Multi-classifier systems to develop a powerful and efficient system for handwritten character recognition. They have used 36 Telugu consonant classes and 15 vowel modifiers in that study. The proposed system gave a performance of 92.26% on consonants and 92.0% on vowel modifiers.

2.4.5.3 Kannada

Kannada is the official language of the state of Karnataka and is spoken by about 44 million people. The Kannada script is evolved from the Kadamba script, a descendent of Brahmi. The script is alpha syllabic and is phonetic. There are 13 Vowels (Swara), 2 part vowel, part consonants (Yogavaha) and 34 Consonants (Vangana). When two consonants join together, compound characters are formed. The script also includes 10 different Kannada numerals.

The first research paper on Kannada HCR appeared in the year 2006. Sharma et al. [117] deals with a quadratic classifier based scheme for the recognition handwritten numerals belonging to 10 classes. The features used in the classifier are obtained from the directional chain code information of the contour points of the characters. Their scheme is tested with 2300 data samples and obtained 97.87% and 98.45% recognition accuracy using 64 dimensional and 100 dimensional features respectively.

For the recognition of numerals, Rajput and Hangarge [118] proposed image fusion scheme. Several digital images corresponding to each handwritten numeral are fused to generate patterns, which are stored in 8×8 matrices, irrespective of the size of images and the numerals to be recognized are matched using k-NN classifier. The recognition result of 91% was obtained for 250 test numerals and a recognition result of 89% was obtained using 4-fold cross validation. Manjunath Aradhya et al. [119] reported a work on handwritten digit recognition based on Radon Transform. Radon transform represents an image as a collection of projections along various directions. k-NN classifier is used for recognition purpose. The test was performed on the MNIST handwritten numeral database and on Kannada handwritten numerals.

Rajashekararadhya et al. [120] proposed the projection distance metric and zoning based scheme for numeral recognition and tested the method for Kannada and Tamil numerals using k-NN classifier with a recognition accuracy of 93% and 90% respectively. Niranjan et al. [121] proposed an unconstrained handwritten Kannada character recognition system based on Fisher Linear Discriminant Analysis (FLDA). The proposed system extracts features from FLD such as two dimensional FLD and diagonal FLD and classifies using different distance measures.

Ragha and Sasikumar [122], extracted moment features from the Gabor wavelets of preprocessed images of 49 characters. The comparison of moments features of four directional images with original images were

tested using back propagation MLP. The average performance of the system with these two features together was 92%.

Rajput et al. [123] presented a system for recognition of printed and handwritten mixed Kannada numerals using multiclass SVM with a recognition accuracy of 97.76%. The same authors [124] discusses the implementation of shape based features, namely, Fourier descriptors and chain codes. Invariant Fourier descriptors and normalized chain codes are obtained as features and experiments are conducted on handwritten Kannada numerals and vowels. The recognition result with SVM classifier using two shape based features together is 98.45% and 93.92%, for numerals and vowels, respectively.

2.4.5.4 Malayalam

Malayalam is the official language of the state of Kerala and is spoken by about 33 million people. The Malayalam script is evolved from ancient Brahmi. The details about the script are described in the next Chapter in Section 3.3. HCR on Malayalam script was started only recently and studies are conducted only in the basic characters of the language.

Lajish [22] presented a feature extraction method based on fuzzy-zoning and normalized vector distances. In that work, recognition of characters was done using Class Modular Neural Network (CMNN) and accuracy obtained was 78.87%. Lajish [125] also tried to extract features

from gray scale images using State Space Map (SSM) and State Space Point Distribution (SSPD) parameters. A CMNN was employed for recognition and the accuracy obtained was 73.03%. The numbers of distinct classes used in these experiments were forty four. This work pointed out the need for more efficient features for this character recognition problem.

Raju [126] used zero-crossing of wavelet coefficients for the recognition of unconstrained handwritten Malayalam characters using 30 classes. In [127], performance analysis of wavelet features using twelve different wavelet filters were done with MLP network and the average recognition accuracy reported was 76.8%. Chacko and Babu [128] deals with the recognition of handwritten Malayalam characters using discrete features extracted from skeleton of images. Pruning of skeleton is done by contour portioning with discrete curve evolution. Classification is done using MLP and the recognition accuracy obtained was 90.18%. The number of classes used in the experiment was 33. The same authors [129] presented a method for recognition of 30 classes of characters based on edge detection method. Canny edge detector is used to produce thinned edges of the character. To avoid, spurious branches, nonlinear anisotropic diffusion via partial differential equations was applied. Finally, the broken edges are linked using ant colony optimization method. For classification MLP is used and the result obtained was 95.16%. Chacko et al. [130] classified 30 classes using wavelet energy and Extreme Learning Machine

(ELM) and the accuracy reported is 95.59%. Moni and Raju [131] implemented a feature extraction method based on Run Length Count (RLC) where RLC is the count of contiguous group of 1's encountered in a left to right or top to bottom scan of a character image or block of an image. Using MQDF, the recognition accuracy obtained is 94.18%. The same authors [132] proposed directional features for recognition using MQDF and obtained an accuracy of 95.42%. In [133], the authors have created feature vector by traversing each zone diagonally. For the classification, simplified quadratic classifier is used. The study was carried out with a database containing isolated handwritten characters pertaining to 44 classes and obtained a recognition accuracy of 97.6%.

All of these works deal with 44 or lesser number of classes. There is no work on the recognition of compound characters in Malayalam.

2.5 Summary of Developments of HCR in Indian Scenario

Majority of works reported in the Indian scenario is based on either numerals or basic characters of the script. The work on compound characters is reported only in Bangla and Devnagari script. These works are summarized in Table 2.1.

Table 2.1 Summary of developments in Indian scripts

Authors	Year	Script	Type	Features	Classifiers	Acc (%)
Sharma et al. [74]	2006	Devnagari	Basic Characters	Chain Code	Quadratic	80.36
Sharma et al. [74]	2006	Devnagari	Numerals	Chain Code	Quadratic	98.86
Hanmandlu and Murthy [75]	2007	Devnagari	Numerals	Box Approach	Fuzzy Model	95.0
Arora et al. [76]	2007	Devnagari	Basic Characters	Shirorekha, Spine	NN	89.12
Pal et al. [78]	2007	Devnagari	Numerals	Directional	MQDF	99.56
Pal et al. [79]	2007	Devnagari	Basic Characters	Gradient	MQDF	94.24
Arora et al. [77]	2008	Devnagari	Basic Characters	Combined	Hybrid	92.80
Mukherji and Rege [81]	2009	Devnagari	Basic Characters	Shape Features	Fuzzy Logic	86.4
Jangid [82]	2011	Devnagari	Basic Characters	Statistical	SVM	94.89
Shelke and Apte [83]	2011	Devnagari	40 Compound Characters	Structural and Wavelet	Two-Stage NN	96.23
Bhattacharya et al. [84]	2002	Bangla	Numerals	Topological and Structural	Tree classifier, MLP	93.26
Rahiman et al. [85]	2002	Bangla	Basic Characters	Structural	Multi Stage	88.38
Bhowmik et al. [86]	2004	Bangla	Basic Characters	Stroke	MLP	84.33
Bhattacharya et al. [87]	2006	Bangla	Basic Characters	Shape	MLP	92.14
Pal et al. [78]	2007	Bangla	Numerals	Directional	MQDF	98.99
Pal et al. [88]	2007	Bangla	Compound Characters	Gradient	MQDF	85.90
Liu and Suen [63]	2009	Bangla	Numerals	Gradient	SVM	99.40
Bhowmik et al. [89]	2009	Bangla	Basic Characters	Wavelet	SVM	89.22
Das et al. [90]	2010	Bangla	Basic and Compound Characters	Shadow, Longest Run	MLP SVM	79.25 80.51
Das et al. [91]	2012	Bangla	Basic and Compound Characters	Genetic Algorithm	SVM	78.93
Reza and Khan [92]	2012	Bangla	Numerals, Basic Characters and Vowel Modifier	Chain Code	SVM	89.9
Prasad et al. [93]	2009	Gujarati	Basic Characters	None	Template Matching	71.66
Desai [94]	2010	Gujarati	Numerals	Profile	MLP	82.0

Literature Survey

Authors	Year	Script	Type	Features	Classifiers	Acc (%)
Roy et al. [95]	2005	Oriya	Numerals	Chain Code	NN QDF	90.38 94.81
Bhowmik et al. [96]	2006	Oriya	Numerals	Shape	HMM	90.50
Pal et al. [78]	2007	Oriya	Numerals	Directional	MQDF	98.40
Garg and Verma [99]	2009	Gurumukhi	Basic Characters	Structural	NN	69 to 96
Siddarth [100]	2011	Gurumukhi	Basic Characters	Statistical	SVM	95.04
Singh et al. [101]	2012	Gurumukhi	Basic Characters	Gabor Filter	SVM	94.29
Yusuf and Haider [103]	2004	Urdu	Numerals	Discrete Points	Shape Matching	-
Haider and Yusuf [104]	2007	Urdu	Numerals	Shape	SVM	92.6
Liu and Suen [63]	2009	Urdu	Numerals	Gradient	DLQDF	99.73
Sagheer et al. [102]	2009	Urdu	Numerals	Gradient	SVM	98.61
Pathan et al. [105]	2012	Urdu	46 Basic Characters	Invariant Moments	SVM	93.59
Paulpandian and Ganapathy [106]	1997	Tamil	12 Basic Characters	Invariant Moments	HNN	94.4
Suresh et al. [107]	1999	Tamil	7 Basic Characters	Fuzzy Distance	Distance Measure	88
Hewavitharana and Fernando [108]	2002	Tamil	26 Basic Characters	Pixel density	Statistical	80
Pal et al. [78]	2007	Tamil	Numerals	Directional	MQDF	98.51
Sudha and Ramaraj [110]	2007	Tamil	Basic Characters	Fourier Descriptors	MLP	97.0
Bhattacharya et al. [111]	2007	Tamil	Characters (hpl-tamil-iso-char)	Chain Code	K-Means and MLP	89.66
Rajashekaradhyha et al. [120]	2008	Tamil	Numerals	Projection Distance and Zoning	NN	90.0
Pal et al. [109]	2008	Tamil	Basic Characters	Gradient	Quadratic Classifier	96.73
Rajashekaradhyha and Ranjan [114]	2009	Tamil	Numerals	Zone and Centroid	k-NN and BPNN	96.0
Shanti and Duraiswamy [112]	2010	Tamil	Basic Characters	Pixel Density	SVM	82.04
Subashini and Kodikara [113]	2011	Tamil	Basic Characters	SIFT features	K-Means and k-NN	87
Pal et al. [78]	2007	Telugu	Numerals	Directional	MQDF	99.37
Pal et al. [109]	2008	Telugu	Basic Characters	Gradient	Quadratic Classifier	90.90
Rajashekaradhyha and Ranjan [114]	2009	Telugu	Numerals	Zone and Centroid	k-NN and BPNN	99.0

Authors	Year	Script	Type	Features	Classifiers	Acc (%)
Pradhan and Negi [115]	2012	Telugu	Basic Characters	Discrete Curves	String Matching	-
Soman et al. [116]	2013	Telugu	Consonants Vowel Modifiers	-	Multi-classifier	92.26 92.0
Sharma et al. [117]	2006	Kannada	Numerals	Directional	Quadratic	98.45
Pal et al. [78]	2007	Kannada	Numerals	Directional	MQDF	98.71
Manjunath Aradhya et al. [119]	2007	Kannada	Numerals	Radon Transform	k-NN	91.2
Rajashekararadhya et al. [120]	2008	Kannada	Numerals	Projection distance, Zoning	k-NN	93.0
Pal et al. [109]	2008	Kannada	Basic Characters	Gradient	Quadratic Classifier	90.34
Niranjan et al. [121]	2008	Kannada	Basic Characters	FLDA	Distance Measure	68.0
Rajashekararadhya and Ranjan [114]	2009	Kannada	Numerals	Zone and Centroid	k-NN and BPNN	99.0
Ragha and Sasikumar [122]	2010	Kannada	Basic Characters	Moment	MLP	92.0
Rajput et al. [123]	2010	Kannada	Numerals	Fourier	SVM	97.76
Rajput et al. [124]	2011	Kannada	Numerals	Fourier and Chain Code	SVM	98.45
Rajput et al. [124]	2011	Kannada	Basic Characters	Fourier and Chain Code	SVM	93.92
Lajish [22]	2007	Malayalam	44 Basic Characters	FZ and NVD	CMNN	78.87
Lajish [125]	2008	Malayalam	44 Basic Characters	SSM and SSPD	CMNN	73.03
Raju [127]	2008	Malayalam	30 Basic Characters	Wavelet	MLP	76.8
Chacko and Babu [128]	2009	Malayalam	33 Basic Characters	Discrete Features	MLP	90.18
Rajashekararadhya and Ranjan [114]	2009	Malayalam	Numerals	Zone and Centroid	k-NN and BPNN	95.0
Chacko and Babu [129]	2010	Malayalam	30 Basic Characters	Ant Colony Optimization	MLP	95.16
Chacko and Babu [130]	2011	Malayalam	30 Basic Characters	Wavelet Energy	ELM	95.59
Moni and Raju [131]	2011	Malayalam	44 Basic Characters	RLC	MQDF	94.18
Moni and Raju [132]	2011	Malayalam	44 Basic Characters	Directional	MQDF	95.42
Moni and Raju [133]	2012	Malayalam	44 Basic Characters	Diagonal Features	Simplified Quadratic	97.60

The above mentioned HCR systems have proposed many feature extraction methods and classification techniques for different languages.

Performance of HCR greatly depends on the characteristics of the script and the approaches found suitable for one script may not give the similar performance for another script. Most of these works, deal with handwritten numerals where the number of classes is only 10. The works on handwritten characters is limited to a sub set of basic characters in most of the scripts. Recognition of compound characters is still to be explored in many Indian scripts including Malayalam script. The recognition accuracy of a system is affected by a number of factors such as number of classes considered; number of samples used per class; the way in which samples are collected; and off course, the set of features extracted and classifiers used. From the above summary, it is clear that the recognition accuracy obtained has reached a significant level only for numerals where the number of classes is limited to 10. When the recognition system deals with more number of classes including compound characters in the Indian script, the accuracy goes below an adequate level. We could trace only few works that deals with HCR on compound characters (Table 2.1). The HCR system that considers basic characters together with compound characters is reported only in [91]. From the above results, it is obvious that recognition of basic characters together with compound characters and vowel modifiers is more difficult, yet necessary and under studied problem in HCR research.

2.6 Commercial Character Recognition Softwares

2.6.1 Machine Printed

Tesseract is one of the earlier character recognition software which recognizes printed characters in various languages. The latest release (2012) of this software can recognize characters in Arabic, Bulgarian, Catalan, Czech, Danish, Dutch, English, Finnish, French, German, Greek, Hindi, Hungarian, Indonesian, Italian, Latvian, Lithuanian, Norwegian, Polish, Portuguese, Romanian, Russian, Serbian, Slovak, Slovenian, Spanish, Swedish, Thai, Turkish, Ukrainian and Vietnamese.

For the recognition of Indian languages, Technology Development for Indian Languages (TDIL) developed softwares for Bangla, Devanagari, Gurmukhi, Kannada, Malayalam and Tamil. NAYANA is the character recognition software developed by C-DAC, Thiruvananthapuram for printed characters in Malayalam.

2.6.2 Handwritten

In handwriting domain, there are various tools available in foreign languages. LEADTools is an intelligent character recognition system for Spanish, English, French, German and Italian. A2iA-DocumentReader, CheckReader and AddressReader stand for recognition of English, French, German, Italian, Portuguese, Spanish and Arabic languages.

There is no commercial system is available for recognizing handwritten manuscript in any Indian languages.

2.7 Conclusion

Overview of a character recognition system is demonstrated and a detailed survey has been conducted on languages of the world. Different feature extraction and classification methods are also covered. Developments in Latin, Chinese and Japanese scripts are matured and research is progressing toward the recognition of more reliable system. But in Indian scenario, there is no full-fledged system available for recognizing handwritten manuscripts. But developments in this area are active and there is much more way to go ahead. One of the major difficulties in this field is the lack of benchmark database of handwritten characters for most of the Indian languages for research. Combination of classifiers is not attempted much in Indian HCR research. Compound character recognition is addressed only in Bangla and Devnagari (Marathi) scripts. Recognition of degraded character is another issue not addressed so far.

In the Indian context, the recognition accuracy has reached a substantial level only in the recognition of numerals. When the recognition system deals with more number of classes including compound characters in the Indian script, the accuracy goes below an adequate level.

From the literature, it is noteworthy that, research work on Malayalam is limited to basic characters of the language. In the case of Malayalam, occurrence of compound characters is more common in the script. In this context, we propose a system for the recognition of unconstrained handwritten Malayalam characters consisting of vowels, consonants, pure consonants, vowel signs, consonant signs and compound characters in a writer independent environment.

DEVELOPMENT OF A COMPREHENSIVE DATASET AND PREPROCESSING

3.1 Introduction
3.2 Malayalam Language
3.3 Malayalam Script Overview
3.4 Data Collection
3.5 Data Preparation
3.6 Description of the Generated Data
3.7 Document Acquisition, Enhancement and Database Creation using Mobile Phone Camera
3.8 Data Preprocessing
3.9 Conclusion

3.1 Introduction

One of the most challenging aspects of offline handwritten character recognition is finding a good database that well represents a wide variety of handwriting styles. The non-availability of standard/benchmark datasets that contain the most important classes of the language is an obstruction to the effective research work. An alphabetic character set has components whose shapes reflect the culture in which the character set was born [7]. All the characters are rich in shapes and they are different from each other though they share general

similarities. A single character from the alphabet is subject to many variations while in writing.

In this chapter, we first discuss the peculiarities of Malayalam language and its script. Based on this information, we derive the symbols needed in the data collection sheet. Data collection methods are described in Section 3.4. To create a benchmark database, the characters need to be segmented from the handwritten page. We have designed an algorithm for this task based on projection profiles and connected component labelling. The details of this method are described in Section 3.5. Description about the generated data can be found in Section 3.6. Procedure for creating dataset from document captured using mobile phone camera is discussed in Section 3.7. Data preprocessing methods adopted in this work such as smoothing and noise removal, normalization, binarization, thinning and contour extraction are described in Section 3.8. We conclude the chapter in Section 3.9.

3.2 Malayalam Language

Malayalam is one among the twenty two scheduled languages of India and was declared as a classical language by the Government of India in 2013. It is one among the four major Dravidian languages of India and one among the ten major Indian scripts and has official language status in the state of Kerala and union territories of Lakshadweep and Puducherry. It is spoken by 33 million people and is ranked eighth in India in terms of number of speakers. As Malayalam

speakers are itinerant, the language is heard widely all over Indian as well as in Gulf countries, Europe, Australia and North America [134]. Malayalam is closely related to Tamil and has indelible impression of Sanskrit. Consequently, Malayalam language is enriched with largest number of characters among all Indian languages and many characters are distinct just with a small variation in appearance. This language is capable of representing all sounds in Sanskrit and Dravidian languages.

3.3 Malayalam Script Overview

Malayalam script is derived from the Grantha script, an inheritor of ancient Brahmi script. Like many other Indian scripts, it is an abugida, or writing system that is partially “alphabetic” and partially syllable-based. Malayalam alphabet is unicase, or does not have a case distinction. Mode of writing is from left to right. The characters are isolated in nature and there is no concept of cursive writing. The script does not have a ‘Sirorekha’ as observed in north Indian scripts.

The alphabet consists of vowels (swarangaḷ, സ്വരങ്ങൾ) and consonants (vyanganagaḷ, വ്യഞ്ജനങ്ങൾ) called basic characters. Each vowel has two forms, an independent form and a dependent form. An independent vowel is used as the first letter of a word beginning with a vowel (Table 3.1). Dependent vowel sign is a diacritic attached to a consonant when the consonant is followed by a vowel other than the first vowel അ (/a/). These vowel signs have glyph pieces which do not exist

on their own and they appear either to the left, right or both sides of consonant (Table 3.2). The dotted circles are not a part of the symbol, but depict the position of each vowel signs. All the 36 consonants are displayed in Table 3.3.

Table 3.1 Malayalam vowels

അ	ആ	ഇ	ഈ	ഉ	ഊ	ഋ	എ
ഏ	ഐ	ഒ	ഓ	ഔ	അം	അഃ	

Table 3.2 Dependent vowel signs

ാ	ി	ീ	ു	ൂ	്യ	െ
േ	ൈ	ൊ	ോ	ൗ	ം	ഃ

Table 3.3 Malayalam consonants

ക	ഖ	ഗ	ഘ	ങ
ച	ഛ	ജ	ഝ	ഞ
ട	ഠ	ഡ	ഢ	ണ
ത	ഥ	ദ	ധ	ന
പ	ഫ	ബ	ഭ	മ
യ	ര	ല	വ	
ശ	ഷ	സ	ഹ	
ള	ഴ	റ		

There are three special consonant signs in the script. When the consonant യ (/ya/) is placed at the end of consonant, a special sign is placed after consonant as in ക്യ. When consonants ര (/ra/) or റ (/ra/) is placed at the end of a consonant, a special sign is pre-posed to the consonant as in ക്ര. When the consonant ല (/la/) is at the end of a consonant, a special sign is put at the bottom of the consonant as in ള്. Similarly, when the consonant വ (/va/) is at the end of a consonant, the special sign is post-posed to consonant as in ക്വ [135].

Apart from basic characters, there are pure consonants (chillaksharangal, ചില്ലക്ഷരങ്ങൾ) and compound characters (koottaksharangal, കൂട്ടക്ഷരങ്ങൾ) in the script. Pure consonants (Table 3.4) are a unique property of Malayalam among Indian languages, which is derived from basic consonant units: ണ (/ṇa/), ന (/na/), റ (/ra/), ല (/la/), ള (/ḷa/).

Table 3.4 Pure consonants

ൺ	ൻ	ർ	ൽ	ൾ
---	---	---	---	---

Compound characters are special type of characters formed as combination of two or more (pure) consonants. The shapes of the compound characters are different from the shapes of the constituent characters. These shapes are normally has complex orthographic structure

and some of them are difficult to recognize when isolated from the context. Two types of compound characters occur in Malayalam: one vertically compounded (൧, ഋ, ഡ) and the other horizontally compounded (കര, തര, മര). There exist many similar characters with little difference in their shape.

In 1969–1971 and in 1981, the Government of Kerala reformed the orthography of Malayalam script [136]. When consonants are followed by vowels such as ഉ (/u/), ഊ (/ū/), ഋ (/r/) or consonants such as ര (/ra/) or റ (/ra/), the character was represented by irregular glyph. Such irregularity is simplified in the script revision by writing the consonant and vowel-consonant signs separately as depicted in Table 3.5 rather than as a complex character and therefore any radical change in the shape of the characters are not observed.

Table 3.5 Script Revision: Usage of separate symbols for vowel-consonant signs

Old script	ക	കു	കൃ	ക്ര	ര	രു	രൃ	ര്ര	പ	പു	ര	രൂ
New script	കു	കൂ	കൃ	ക്ര	രു	രൂ	രൃ	ര്ര	പു	പൂ	രു	രൂ

The second major change is the considerable reduction in the number of special shapes representing large number of compound characters. The compound characters that occur regularly in Malayalam documents are written without any change (Table 3.6). Most of the other

letters are borrowed from Sanskrit. Instead of writing them in combined form, a crescent mark ്, called ‘chandrakkala’ (ചന്ദ്രക്കല) is used to separate a compound character into constituent consonants. This symbol is placed at the right top of the first consonant as demonstrated in Table 3.7. Due to these simplified approach, the new script needs only few symbols instead of 500 different letters required for handling old script.

Table 3.6 Compound characters

ക	ക	കേ
ച	ച	ചേ
ട	ട	ടേ
ത	ത	തേ
പ്പ	മ്പ	മ്മ
ല	ല	ലേ

Table 3.7 Script Revision: Usage of ് (chandrakkala)

Old script	ഹ	ര	ക	ച	ശ	ജ
New script	ഹ്	ര്	ക്	ച്	ശ്	ജ്

Though the script has been reformed, one could find a mixture of both old script and new script while writing. Some of the most commonly

used compound characters in the old script include: ഇള, ക്ഷ, റ, ൽ, ണ, ഡ, ള and ല.

Usage of basic characters, vowel signs and compound characters are common in a document as we can observe from Table 3.8 describing frequency of occurrence of top 20 symbols in Malayalam document, which was published on the basis of a frequency analysis of occurrence pattern in printed Malayalam literature [137].

Table 3.8 Frequency of occurrence of top 20 symbols

Sl. No.	Symbol	Freq.	Sl. No.	Symbol	Freq.
1	ി	0.0812	11	ം	0.0271
2	ാ	0.0777	12	േ	0.0262
3	ു	0.0746	13	മ	0.0242
4	െ	0.0399	14	്	0.0233
5	യ	0.0339	15	പ	0.0211
6	ര	0.0323	16	ന	0.0193
7	ക	0.0305	17	ട	0.0177
8	വ	0.0297	18	ക്ക	0.0172
9	ന	0.0295	19	ത്ത	0.0164
10	ത	0.0292	20	സ	0.0151

The script also consists of 10 numerals (Table 3.9), but it is seldom in use and instead Arabic numerals are in practice.

Table 3.9 Malayalam numerals

0	1	2	3	4	5	6	7	8	9
൦	൧	൨	൩	൪	൫	൬	൭	൮	൯

From the above knowledge, we adopted the basic graphical shapes in the script such as independent vowels, dependent vowel signs, consonants, pure consonants, consonant signs and the entire compound characters in the modern script along with the frequently used symbols already mentioned, as the data set.

3.4 Data Collection

A data collection sheet was provided to the contributors and requested to write Malayalam vowels, consonants, pure consonants, compound characters and vowel – consonant signs along with an inherent consonant. The contributors of data are native speakers of Malayalam language as the writing styles of native writers are different from those who are using this script as a second language. The style of the native writer is evolved from the learning which starts with elementary school where as the writing of non-native writer is derived from the structural pattern of the symbol or its similarity with his/her own language. Writers included are of different educational levels as handwriting skills are

correlated with education. The data is collected from school children, college students, university graduates and persons from various professions such as office staff, teachers, doctors etc. A small percentage of data are also collected from senior citizens. Each contributor is requested to write with his/her own writing instrument so as to reflect their natural handwriting style. In the case pen was not in hand with the writer, it was provided at random from a set of different types of pens. No constraints are imposed on the writer such as the style of writing, type of pen, colour of ink, thickness of lines, way of writing etc. and no special boxes or lines are provided for writing each character. Printed version of the characters is provided at the top of each section, so that people will not ignore any character while writing. The purpose of data collection was not disclosed and the option of disclosing personal information was left to the contributor. Proportions of male and female writers were approximately equal and writers belong to different districts of Kerala such as Ernakulam, Idukki, Thrissur, Palakkad. Around 450 pages of samples have been collected over a span of more than two years. Statistics of the contributors are provided in Fig. 3.1.

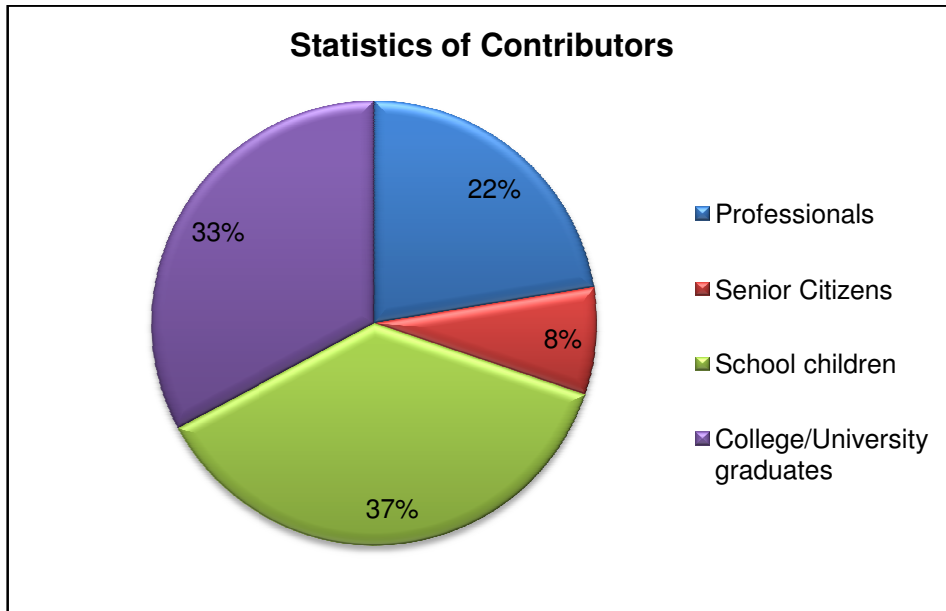


Fig. 3.1 Statistics of contributors used for data collection

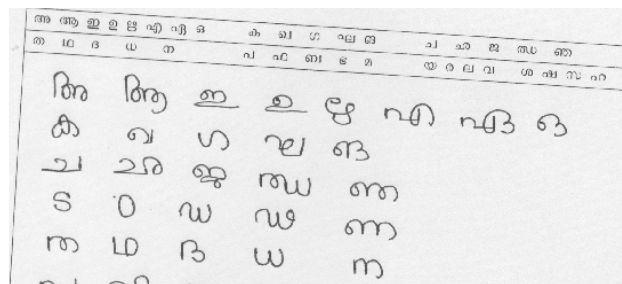
3.5 Data Preparation

Digitization of collected samples are done by a Flat-bed scanner (HP, Model Name: Scanjet 2400), by setting dpi to 300. The scanned images of the original data sheet are stored in TIFF format for future use. Fig. 3.2 shows sample data collection sheet written by a writer. It is tedious and time consuming task to isolate character symbols from the handwritten page. Hence, algorithms were developed to separate symbols from the handwritten page to create database. The overall procedure includes, skew correction and segmentation of isolated characters.

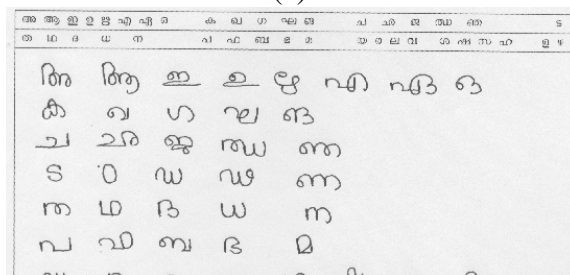
3.5.1 Skew Detection and Correction

Digitization using scanner may lead to skew in the image. To align handwritten page before further processing, the digitized image is checked for skewing. The initial true coloured image is converted to gray scale and thresholded using Otsu's global thresholding algorithm as explained in Section 3.8.3.

The angle made by the border line of the page with horizontal axis is calculated. This skew, if detected, is corrected by rotating the image by the skew angle in the opposite direction. Part of the handwritten page which undergoes skew correction of angle 3.2° is depicted in Fig. 3.3



(a)



(b)

Fig. 3.3 Part of skewed and skew corrected image

3.5.2 Segmentation

Segmentation step is required to isolate characters automatically from the handwritten page and we have designed an algorithm for this purpose based on horizontal projection profiles and connected component labelling.

3.5.2.1 Horizontal Projection Profile

Horizontal projection profile $H(x)$ is defined as the running count of black pixels in each row and it is calculated using Eq. 3.1. Zero or small values in the profile indicate white spaces in between the characters and it is considered as the line separation points.

$$H(x) = \sum_{i=1}^N \mathbf{IMG}_{M \times N}(x, i) \quad 3.1$$

3.5.2.2 Connected Component Labelling

The labelling of connected component in an image is central to automated segmentation of patterns. Connected component labelling scans the image and groups its pixels into components based on pixel connectivity. All pixels in the connected component share similar pixel values and are connected with each other. Once all groups have been determined, each pixel is labeled and serially numbered according to the component to which it was assigned.

3.5.2.3 Extraction of Characters

Extraction of characters from handwritten page is explained in the following algorithm.

Algorithm 3.1: Character Extraction Algorithm

Input: Filled in data collection sheet

Output: Segmented character images

- Step 1: The boundary around the handwritten page is removed as it is the first and biggest connected component.
- Step 2: Let $IMG_{M \times N}(x, y)$ be the image obtained from Step 1 with M rows and N columns, where $1 \leq x \leq M, 1 \leq y \leq N$.
- Step 3: Compute the horizontal projection profile $H(x)$ of $IMG_{M \times N}$ as specified in Sub Section 3.5.2.1
- Step 4: From $H(x)$, the entire valley points (zero or small values) are identified. These points are used to segment the page into rows. Each row is separately processed. The characters in each segmented line are labeled using connected component labelling algorithm as mentioned in Sub Section 3.5.2.2. Isolated pixels are treated as noise and hence removed.
- Step 5: The minimum bounded rectangles (the rectangles in Fig. 3.4) containing the component are detected and cropped one by

one from the input image. These segmented character images are automatically stored in separate files in uncompressed TIFF format and placed in proper folders. 90 folders are created for storing 8 vowel symbols, 36 consonant symbols, 5 pure consonant symbols, 26 compound character symbols, 11 vowel signs and 4 consonant signs.

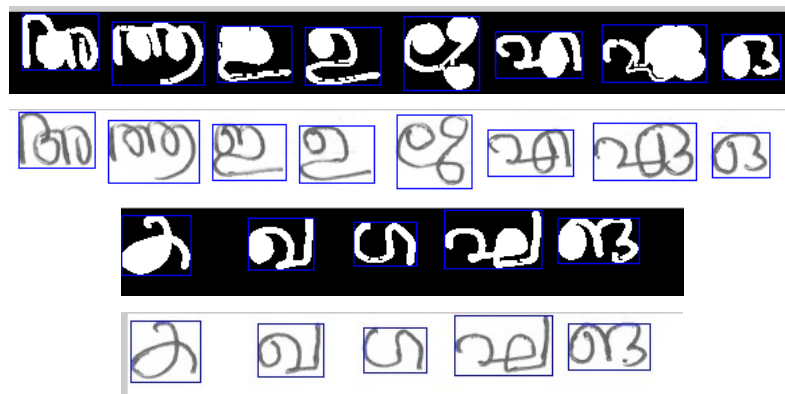


Fig. 3.4 Extraction of characters from the filled in data collection sheet

All files are visually checked for errors and these errors are corrected manually. The verification process displays samples of same class on the screen so that errors can be corrected (Fig. 3.5).

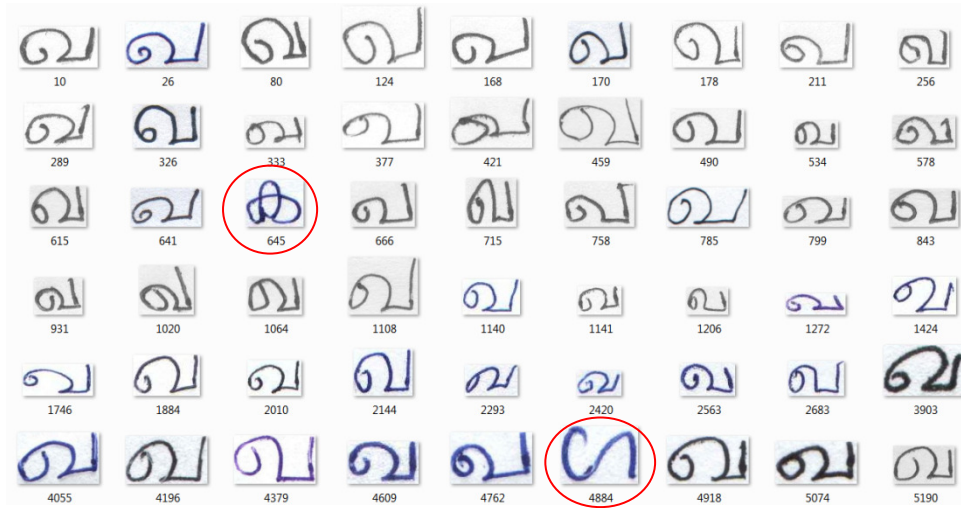








Fig. 3.5 Manual refinement of samples

3.6 Description of the Generated Data

All the character samples along with their corresponding class reference are displayed in Fig. 3.6. Some of the characters are written more than one way, for example,  and ,  and ,  and . The database mostly contains the first form of pure consonants as it is written by most people. Second form is written by few people is also preserved in the database. Distorted or broken samples are also conserved in the database. Some of the characters are wrongly written while some are omitted by the writers. Some of them are incorrectly segmented due to the limitations of the segmentation algorithm. Such invalid or incorrectly segmented character images (Fig. 3.7) are manually removed from the folders. Due to the above mentioned reasons, the frequency of one

character pattern varies from another pattern. This database contains 1991 samples of vowels, 9045 samples of consonants, 1498 samples of pure consonants, 6100 samples compound characters, 2200 samples of vowel signs and 800 samples of consonant signs. For uniform distribution of samples per class, 200 samples per 90 classes are randomly chosen to create a comprehensive database of 18000 samples.

Class No.	Character	Handwritten Samples	Class No.	Character	Handwritten Samples
1	അ		2	ആ	
3	ഇ		4	ഉ	
5	ഋ		6	എ	
7	ഘ		8	ഒ	
9	ക		10	ഖ	
11	ഗ		12	ച	
13	ങ		14	പ	
15	മ		16	ജ	
17	ഡ		18	ഞ	
19	ട		20	ഠ	
21	ഡ		22	ഡ	

Class No.	Character	Handwritten Samples	Class No.	Character	Handwritten Samples
23	ണ		24	ത	
25	ഥ		26	ദ	
27	ധ		28	ന	
29	പ		30	ഫ	
31	ബ		32	ഭ	
33	മ		34	യ	
35	ര		36	ല	
37	വ		38	ശ	
39	ഷ		40	സ	
41	ഹ		42	ള	
43	ഴ		44	റ	
45	ൺ		46	ൻ	
47	ൽ		48	ർ	
49	ൾ		50	ൺ	

Development of Comprehensive Dataset and Preprocessing

Class No.	Character	Handwritten Samples	Class No.	Character	Handwritten Samples
51	ക		52	ങ	
53	ച		54	ബ	
55	ത		56	ദ	
57	ന		58	ണ	
59	ര		60	ല	
61	മ		62	പ്പ	
63	വ		64	മ്മ	
65	ശ		66	ല്ല	
67	ഝ		68	ള	
69	ജ		70	റ്റ	
71	ഞ		72	ത	
73	ഡ		74	ദ	
75	ല		76	ഃ	

Class No.	Character	Handwritten Samples	Class No.	Character	Handwritten Samples
77	ി		78	ീ	
79	ു		80	ൂ	
81	്യ		82	െ	
83	േ		84	ൗ	
85	ോ		86	ഃ	
87	േ		88	്ല	
89	്യ		90	്	

Fig. 3.6 Samples of handwritten characters along with their class reference

Fig. 3.7 Invalid characters deleted from the database

In pattern recognition literature, we are concerned more about shape and size of the patterns, not the RGB colour details. So we have converted the true coloured RGB image to the gray scale intensity image by using a weighted sum of the R, G and B components: $0.2989 \times R + 0.5870 \times G + 0.1140 \times B$. A glimpse of characters stored in the database is depicted in Fig. 3.8.



Fig. 3.8 A glimpse of handwritten samples in folder 1 and folder 58

3.7 Document Acquisition, Enhancement and Database Creation using Mobile Phone Camera

The usage of mobile phone cameras and digital cameras are rapidly increasing, but handwritten character recognition applications are in the beginning stage mainly because the acquisition environments of camera images are different from scanner images. Owing to the increased usage of mobile phones, a subset of documents is captured through mobile phone camera (Nokia E5–5MP), keeping it parallel so as to avoid perspective distortion. The proposed system allows the user to snap image of handwriting and send it to the server; a server side program invokes character recognition procedure to recognize the characters. Fig. 3.9 displays the system framework for acquisition of data through mobile phone camera.

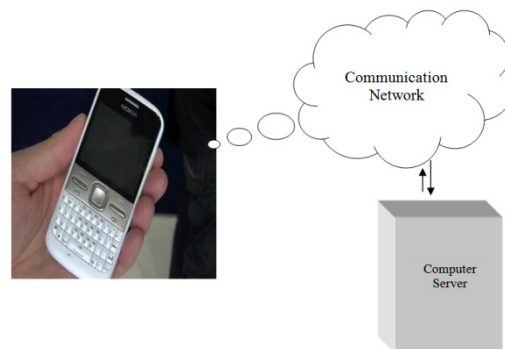


Fig. 3.9 System framework for acquisition of data through mobile phone camera

Global thresholding method by Otsu [138] is one of the most widely used binarization algorithm. Even though, this algorithm is very

fast and simple to implement, it works well only for images with uniform illumination. However, images captured by camera are not uniformly illuminated. The illumination variation in the document images makes it difficult to extract character areas from the document through binarization. To conquer this problem, after converting the colour image to gray scale, image enhancement is performed by contrast stretching as in Eq. 3.2. where min is the minimum intensity and max is the maximum intensity of $f(x,y)$, the input image. Fig. 3.10 shows the effect of Otsu's global thresholding method on this image and the effect of processing after contrast enhancement.

$$g(x,y) = 255 * (f(x,y) - min) / (max - min) \quad 3.2$$

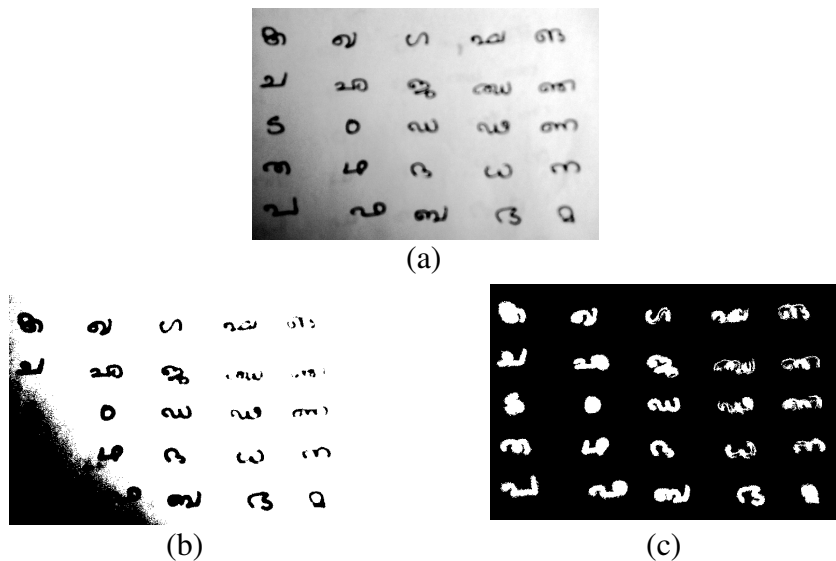


Fig 3.10 Processing of camera captured document (a) Image loaded from mobile memory (b) Effect of binarization using Otsu's method (c) Dilated and filled image after contrast enhancement

The segmentation procedure is similar to the one described in Section 3.5.2. As the captured images are of low quality, some of the characters became distorted during binarization, leading to incorrect segmentation. Each segmented character images are stored in the corresponding folders of the database in gray scale format. This database contains a total of 741 samples of first 25 consonants. Samples of each consonant from the database are displayed in Fig. 3.11.

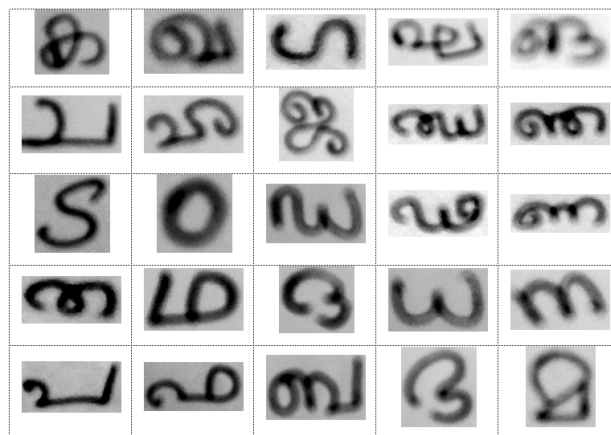


Fig. 3.11 Some of the samples from the database

3.8 Data Preprocessing

Preprocessing step enhances the quality of the character samples. A series of preprocessing steps are required to make the character image ready for feature extraction. In this section, all preprocessing operations

which are applied to one or more of the feature extraction algorithms are described. Fig. 3.12 demonstrates the preprocessing steps applied to the input image. The steps are summarized below:

- Smoothing and Noise Removal
- Size Normalization of the character image
- Binarization of the character image
- Thinning of binary image
- Contour Extraction of binary image
- Gray Scale Normalization of the image

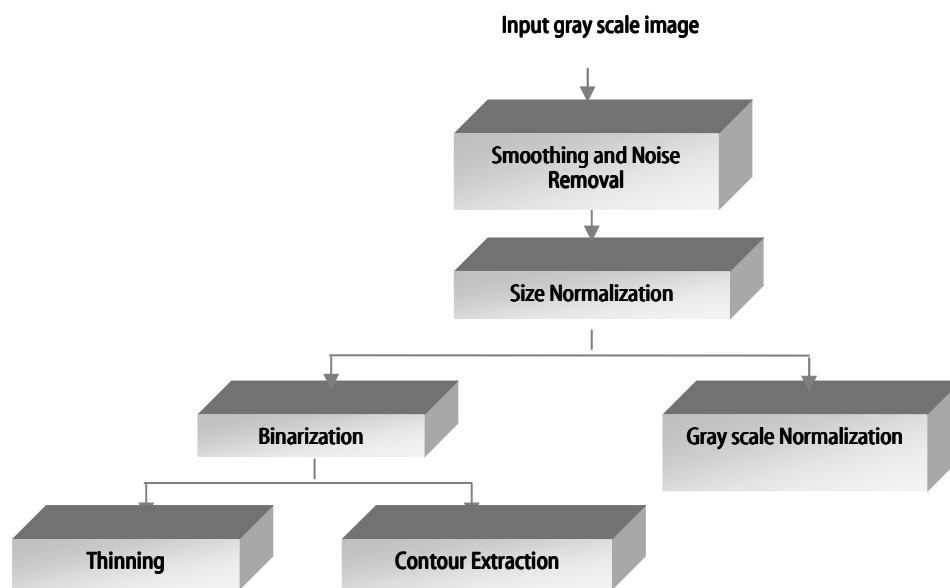


Fig. 3.12 Steps in preprocessing

3.8.1 Smoothing and Noise Removal

Smoothing operations in a character images are used for blurring and for noise reduction. It is used for the removal of small details from an image which in turn improves the tolerance of character shape variation. Smoothing operations are used to reduce the noise or to straighten the edges of the character by filling small gaps or removing small bumps [8].

For smoothing, we have applied 3×3 averaging filter (Fig. 3.13) 3 times as this filter can remove small pieces of noise and also can blur the images in order to remove the unwanted details.

$$\frac{1}{9} \times \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

Fig. 3.13 3×3 averaging filter mask

3.8.2 Size Normalization of the Image

Normalization is the process of converting the random sized image into standard sized image. This step is required as the size of characters varies from person to person and even with the same person from time to time. The high variability in the character size and shape pose serious problems in designing handwritten character recognition systems. So size

normalization is needed for size invariance on character images prior to feature extraction. The goal of character normalization is to reduce the within-class variation of the shapes of the handwritten pattern in order to facilitate feature extraction process and also improve their classification accuracy [8]. To normalize the size of the character image, it is mapped onto a standard plane with predefined size so as to give a representation of fixed dimensionality. In our work, bicubic interpolation technique is used to convert each image into 64×64 pixel, where the output pixel value is the weighted average of pixels in the nearest 4×4 neighbourhood [139].

3.8.3 Binarization of the Image

Binarization is required to concentrate more on the shape of the characters and remove background details from the objects. Moreover some of the feature extraction algorithms, as we can see in the next chapter, will work only on binary images. Thresholding is the simplest way of binarization, which involves the conversion of a gray scale image (0-255) into a binary image (0 or 1). There are advantages in storing the image in this format. It is easy to manipulate as there are only two possible values. Further processing will be faster and less computationally expensive and storage will be compact though binarization may induce loss of information from the original image or may cause introduction of noise or anomalies.

There are two types of thresholding, namely local and global. Local thresholding techniques calculate a threshold based on the neighbouring pixels. While global techniques determine one threshold for the whole image. Global thresholding is faster than local thresholding techniques. Otsu's method [138] uses global thresholding method which is ranked as the best and the fastest global thresholding method [140]. Therefore, in this work classic Otsu's method is used for binarization. Otsu's method is based on selecting a low point between two gray level histogram peaks from an image to determine a global threshold. It is based on the concept that foreground and background pixels have different mean levels in a gray scale image and may be described as being random numbers located in one of two normal distributions. Information such as the standard deviation and the variance may be computed from these normal distributions. Therefore using these two categories of pixels, it is possible to calculate the total variance of the gray levels values in an image. It is then possible to proceed and separately calculate the variance of the foreground and the background pixels. These denote the within-class variance values. As a final step, the variations of the mean values for each class from the overall mean of all pixels are termed as being the between-classes variance. The objective here is to obtain an optimal threshold by minimizing the ratio of the between-class variance to the total variance.

The within-class variance is defined as follows:

$$\sigma_w^2(t) = \omega_1(t)\sigma_1^2(t) + \omega_2(t)\sigma_2^2(t) \quad 3.3$$

ω_1 and ω_2 are the probabilities of the two classes separated by a threshold t and σ_1^2 and σ_2^2 are variances of the classes.

According to Otsu, minimizing the within-class variance is the same as maximizing between-class variance and is formulated by the equation

$$\sigma_b^2(t) = \omega_1(t)\omega_2(t)[\mu_1(t) - \mu_2(t)]^2 \quad 3.4$$

which is expressed in terms of class probabilities ω_1, ω_2 and class means μ_1, μ_2

3.8.4 Thinning of the Binary Image

Thinning, also known as skeletonization, is an image preprocessing operation performed to make the image crisper by reducing the binary-valued image regions to lines that approximate the skeletons of the region. The process of thinning reduces the width of the handwritten pattern to a set of thin strokes of one pixel wide that retain important information about the shape of the original pattern [139]. This representation is useful for easier extraction features such as end points and branch points as we can see in the next chapter. The algorithm for thinning [141] used in this work is described below:

- Divide the image into two distinct subfields in a checkerboard pattern.
- In the first subiteration, delete pixel p from the first subfield if and only if the conditions G_1 , G_2 and G_3 are all satisfied.
- In the second subiteration, delete pixel p from the second subfield if and only if the conditions G_1 , G_2 and G_3' are all satisfied.

Condition G_1 : $X_H(p) = 1$

where $X_H(p) = \sum_i^4 b_i$ and

$$b_i = \begin{cases} 1 & \text{if } x_{2i-1} = 0 \text{ and } (x_{2i} = 1 \text{ or } x_{2i+1} = 1) \\ 0 & \text{otherwise} \end{cases}$$

and x_1, x_2, \dots, x_8 are the values of the eight neighbours of p (Fig. 3.14), starting with the east neighbour and numbered in counter-clockwise order.

x_4	x_3	x_2
x_5	p	x_1
x_6	x_7	x_8

Fig. 3.14 Eight Neighbours of p

Condition G_2 : $2 \leq \min\{n_1(p), n_2(p)\} \leq 3$

where $n_1(p) = \sum_i^4 x_{2i-1} \vee x_{2i}$ and $n_2(p) = \sum_i^4 x_{2i} \vee x_{2i+1}$

Condition G_3 : $(x_2 \vee x_3 \vee \bar{x}_8) \wedge x_1 = 0$

Condition G_3' : $(x_8 \vee x_7 \vee \bar{x}_4) \wedge x_5 = 0$

The two sub-iterations together make up one iteration of the thinning algorithm. These iterations are repeated until there is no further change in the character pattern.

Samples of normalized, thinned, binary character images are displayed in Fig. 3.15

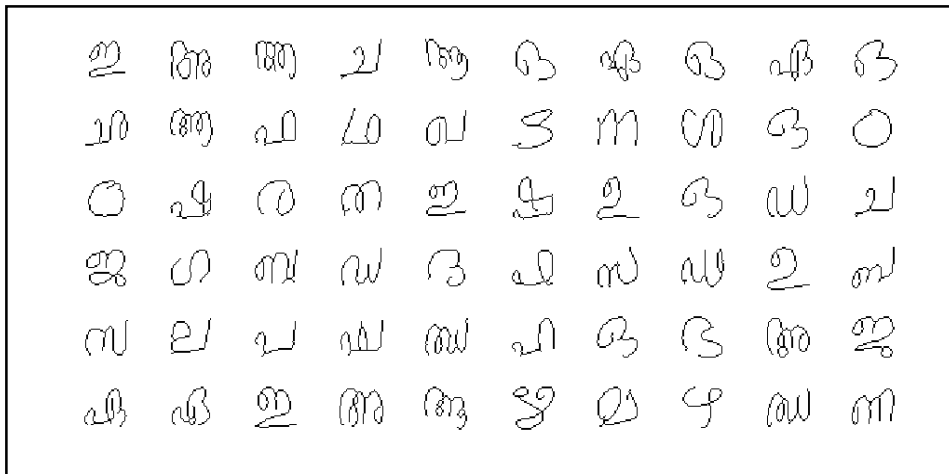


Fig. 3.15 Samples of normalized thinned binary image

3.8.5 Boundary (Contour) Extraction of the Binary Image

The purpose of the boundary extraction algorithm is to extract the information of the border of a handwritten character. The four-connected neighbourhood method is adopted in our work. This algorithm uses the binary image. Boundary is a set of pixels in the image that have at least one background neighbour. The boundary extracting algorithm checks whether the foreground pixel is a boundary pixel or not. If it is a boundary or contour pixel, it is retained; otherwise it is replaced by the background pixel.

3.8.6 Gray Scale Normalization of the Image

To capture features from gray scale images, invariance to contrast between background and foreground is needed along with invariance to the mean gray level value [23]. The background is eliminated from the gray scale image using thresholding. The gray scale images have variable foreground gray levels over different samples. To eliminate the dependence of feature values on gray levels, we rescale the gray levels of foreground pixels of each sample into a standard mean [63].

Illustration on the effect of each preprocessing step on a handwritten character ക (/*ka*/) is provided in Fig. 3.16

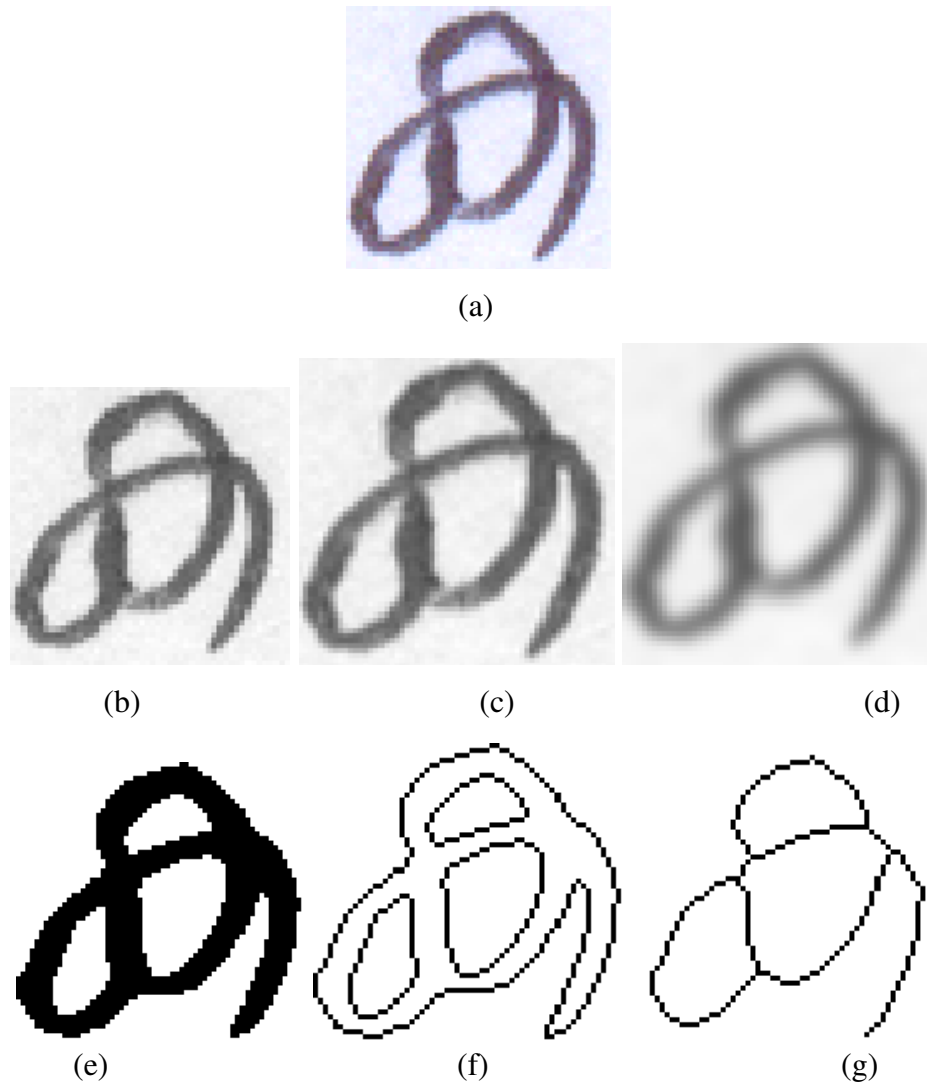


Fig. 3.16 Summary of preprocessing steps (a) Original Image (b) Gray Scale Image (c) Size Normalized Gray Scale Image (d) Smoothed Gray Scale Image (e) Binary Image (f) Contour Representation of Binary Image (g) Skeleton Representation of Binary Image

3.9 Conclusion

In this chapter, we have provided a detailed description the Malayalam language and its script details. The data collection sheet is designed based on this information. We have demonstrated the techniques used for data collection and data preparation. The major contribution in this chapter is the creation of benchmark handwritten database of 18000 samples containing the most important classes of the language. This is the first database containing 90 different character classes. This database contains handwritten samples of all the character patterns in the modern Malayalam script and some of the frequently used compound characters in the old script.

We also created a small database of Malayalam handwritten characters through mobile phone camera owing to the increased usage of cell phone cameras and digital cameras. This is the first attempt in this language.

Preprocessing step enhances the quality of the character samples. We have discussed data preprocessing methods adopted in this work such as smoothing and noise removal, normalization, binarization, thinning and contour extraction. Some feature extraction algorithm can work directly on grayscale images while some work only on binary images. The binary image can also be represented in its contour form or skeleton form. Using appropriate preprocessing method, we have represented a character image, suitable for each feature extraction method.

FEATURE EXTRACTION

4.1	Introduction
4.2	Topological Features
4.3	Distribution of Foreground Pixels
4.5	Transition Count Features
4.6	Local Binary Pattern Descriptors
4.7	Wavelet Features
4.8	Chain Code Features
4.9	Gradient Features
4.10	Curvature Features
4.11	Design of a Novel Feature Descriptor Based on Image Gradient
4.12	Reduction of Feature Dimension
4.13	Conclusion

4.1 Introduction

Feature extraction is important for any character recognition system to achieve high recognition performance. A feature extraction algorithm must be robust enough to handle variety of instances of the same character. A commonly used approach is to represent character pattern by a vector of features and classify the feature vector into its classes. Therefore, the aim of feature extraction method is to find a mapping from the two dimensional image into a smaller one dimensional feature vector $X^T = (x_1, x_2, \dots, x_m)$, that extracts most of the relevant

information of the image so that the intra class variance is small and inter class variance is large. Due to the variability of human writing, this feature based approach is prevalent in handwritten character recognition. The feature based approach is of two types, namely spatial domain and transform domain approaches. Spatial domain approaches derive features directly from the pixel representation of the pattern. Both statistical and structural features can be extracted from character patterns. The statistical features are derived from statistical distribution of points whereas structural features concentrate on the geometric properties of the character. Structural features provide intuitive aspects of writing such as loops, curves, end points, branch points etc., while statistical features are numerical measures computed over images or regions of images such as pixel densities, moments etc. In a transform domain technique, the pattern image is at first transformed into another space and useful features are derived from the transformed images [18].

We have developed features from both spatial domain and transform domain. From the spatial domain both structural and statistical features are developed. From transform domain, wavelet features are developed. The features are categorized below (Fig. 4.1):

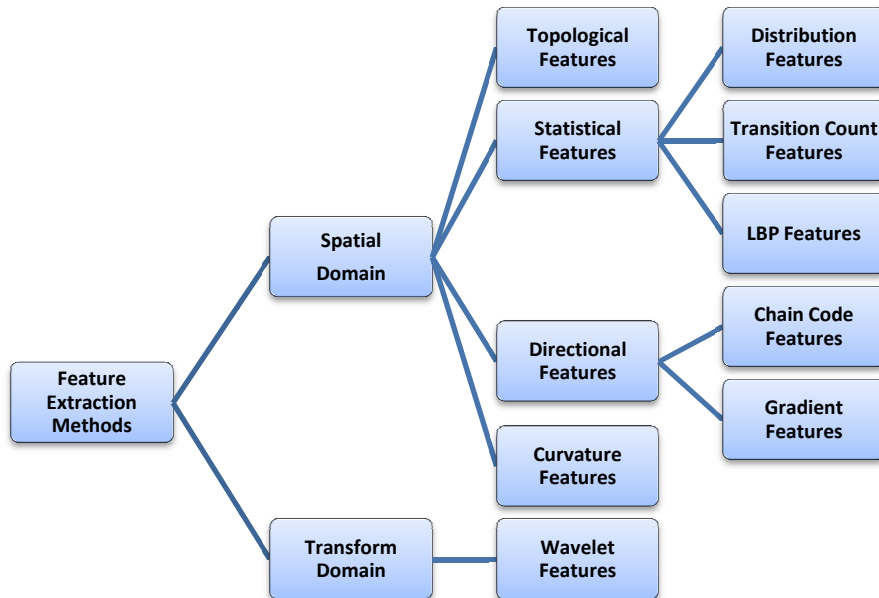


Fig. 4.1 Categorization of used feature extraction methods

Each of these methods may be applied to one or more of the following four representation forms: (a) gray scale character image (b) binary character image (c) character contour (d) character skeleton.

The subsequent sections of this chapter are organized as follows: Section 4.2 demonstrates topological features extracted from the character images. Section 4.3 deals with distribution features. Section 4.4 explains transition count features. Section 4.5 demonstrates LBP features. Topological features provide global characteristics and statistical features provide local characteristics of the pattern. Since these two types of features complement each other, we have integrated them to provide an idea about the overall shape of the character patterns. Section 4.6 explains

Haar wavelet features and feature extraction methods. Section 4.7 and Section 4.8 demonstrates directional features such as chain code and gradient. Section 4.9 explains curvature computation and extraction of curvature features. Section 4.10 demonstrates a novel method of extracting gradient features for efficient discrimination of character patterns. With high dimensionality of the feature, the process of character classification becomes cumbersome. Section 4.11 explains Principal Component Analysis technique for dimensionality reduction and Section 4.12 concludes the chapter.

4.2 Topological Features

After careful analysis of the shape of Malayalam characters, the following topological features were developed.

- **Number of loops:** A loop is a handwritten pattern, when the writing instrument returns to a previous location giving a closed outline with a hole in the centre. There are many characters in Malayalam without any loop, with one loop, with two loops and so on. For example, in the case of the character pattern \bigcirc (*/ra/*), ന്ന (*na*), ന്റ (*ñā*) and ന്നന്റ (*ñña*), the numbers of loops are 0, 1, 2 and 3 respectively. Different types of loops appear in handwritten pattern is displayed in Fig. 4.2. This feature is calculated from the character skeleton.



Fig. 4.2 Samples of loops in handwritten pattern

- **Number of end points:** An end point is defined as the start or end of a line segment. A foreground pixel is considered to be an endpoint if it has exactly one foreground neighbour in the 3×3 neighbourhood. For example, in the case of character pa (/pa/), the number of end points is 2. The number of end points in the skeleton of character pattern da (/da/) of Fig. 4.3 is 3. This feature is extracted from character skeleton.
- **Number of branch points:** A branch point is a junction point that joins three branches. A foreground pixel is considered to be a branch point if it has exactly three foreground neighbours in the 3×3 neighbourhood. For example, there is only one branch point for character image da (/dha/). The number of branch points in the skeleton of character pattern da (/da/) of Fig. 4.3 is 1. This feature is computed from character skeleton.
- **Number of cross points:** A cross point is a junction point connecting four branches. A foreground pixel is considered to be a cross point if it has exactly four foreground neighbors in the 3×3 neighbourhood. For example, there is exactly one cross point in pa

(/zha/). The number of cross points in the skeleton of character pattern Ω (/da/) of Fig. 4.3 is 0. This feature is identified from the skeleton representation of the character pattern.



Fig. 4.3 Character skeleton Ω (/da/)

- **Ratio of width to height:** The width/height ratio is calculated based on the bounding box, which is the rectangle enclosing the character pattern. Some of the characters have width greater than height ($\text{താ}/t\bar{t}a/$, $\text{ന്നാ}/n\bar{n}a/$), some have width less than height ($\text{ടാ}/t\bar{t}a/$, $\text{റാ}/r\bar{r}a/$) and some others have approximately equal width and height ($\text{രാ}/r\bar{a}/$, $\text{ഗാ}/g\bar{a}/$). This feature is calculated from the original segmented image.

The algorithm for the extraction of the above mentioned topological features are provided in Algorithm 4.1

Algorithm 4.1: Extraction of Topological Features

Input: Gray scale images

Output: Topological features

Step 1: Find ratio=width of the image/height of the image

Step 2: Resize character image to 64×64 after smoothing and obtain the skeleton representation of the pattern, where 0 represents white and 1 represents black

Step 3: Find out l , the number of loops in the image

Step 4: Initialize: e , end point count; b , branch point count; and c , the cross point count

Step 5: Find out the 3×3 neighbourhood of each pixel in the character skeleton

Step 6: Find $[n_1, n_2, \dots, n_m]$, the sum of black pixels in each neighbourhood

Step 7: Increment (i) e if $n_k=1$ (ii) b if $n_k=3$ (iii) c if $n_k=4$ for $k=1, 2, \dots, m$

Step 8: create feature vector $[ratio, l, e, b, c]$ and normalize such that the value of feature components range from 0 to 1

4.3 Distribution of Foreground Pixels

For a character image, depending on the shape of the pattern, the pixels are distributed unevenly within the normalized window. To recognize different characters, humans have a natural ability to identify local regions, where the character patterns differ. This ability is imitated by machines to identify characters by dividing the pattern image into a number of equal sized or variable sized regions or zones. This technique is referred to as zoning [142]. From each such region, features are

extracted, which is termed as local features. Due to the nature of Malayalam characters, equal sized zones are adequate for extracting local features. Fig. 4.4 demonstrates the division of character pattern റ (/ra/) into 4×4 equal zones

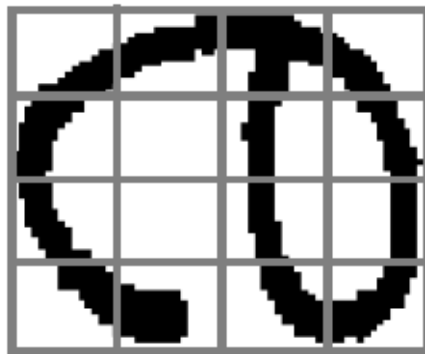


Fig. 4.4 Character pattern റ (/ra/) divided into 4×4 zones

For creating distribution features, each handwritten character pattern is divided into a set of zones using a uniform grid of arbitrary size, and from each zone, the sum of foreground (black) pixels is computed. These features provide an idea about the local distribution of pixels in each zone. The number of local features can be arbitrarily determined by changing the number of zones by varying the size of the grid. The optimal number of zone size is determined experimentally as demonstrated in the next chapter.

4.4 Transition Count Features

Transition count is defined as the number of transitions from a foreground pixel to background pixel along vertical and horizontal lines through a character pattern. These features capture shape information and are less sensitive to the variations of the handwritten pattern. They can be computed directly from the binary image in easy and speedy way and the dimension of the feature descriptor depends on the size of input pattern.

Let the character image I be scanned along each horizontal scan line from top to bottom. For each scan line or row, we count the number of transitions from foreground pixel (black) to background pixel (white).

Let the horizontal transition count, for the i^{th} scan line, be $T_h(i)$. If N is the total number of scan lines, the sequence $\{T_h(i); i = 1, \dots, N\}$ can be treated as the horizontal transition count vector of the image I . To normalize the transition count, we divide each $T_h(i)$ by N , the number of scan lines.

$$NT_h(i) = \frac{T_h(i)}{N} \quad 4.1$$

Normalized horizontal transition count vector of the character image I is defined as:

$$NT_H(I) = \{NT_h(i); i = 1, \dots, N\} \quad 4.2$$

Similarly, a normalized vertical transition count vector $NT_V(I)$ is also created. The transition count feature descriptor for the character image is defined as:

$$TC(I) = \{NT_H(I), NT_V(I)\} \quad 4.3$$

4.5 Local Binary Pattern Descriptors

Timo Ojala et al. [143] created Local Binary Pattern Descriptor (LBP) as a texture descriptor. We have introduced the LBP based feature descriptor for the retrieval of handwritten Malayalam characters [144]

The LBP operator is denoted as $LBP_{P,R}$, where R is the radius of the neighbourhoods and P is the number of neighbourhoods.

To calculate $LBP_{8,1}$ for each pixel, its value is compared to each of its eight neighbours. The comparison gives either a 1 or a 0 indicating whether the centre pixel's value is greater than its neighbours (Eq 4.4). Illustration of LBP operator is provided in Fig. 4.5

$$s(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases}$$

$$LBP_{P,R}(x_c, y_c) = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p \quad 4.4$$

where g_c is the gray scale value in the centre pixel and g_p for $p = 0, \dots, P - 1$ are the gray scale values of the surrounding pixels. This equation results in the generation of 2^P LBP values.

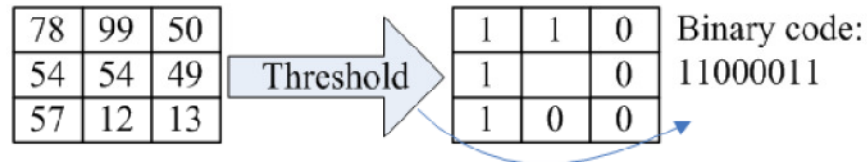


Fig. 4.5 Illustration of LBP operator

To create feature vectors, we find out the LBP code for each pixel in the image. Using this LBP image, we compute the horizontal and vertical projection profiles. The feature extraction method using LBP is explained in Algorithm 4.2.

Algorithm 4.2: Extraction of LBP Features

Input: Gray scale images

Output: LBP features

- Step 1: Find out the LBP code for each pixel in the image*
- Step 2: Compute the horizontal projection profile from the LBP image*
- Step 3: Compute the vertical projection profile from the LBP image*
- Step 4: Concatenate [horzLBP vertLBP] to create feature vector*

Topological features provide global aspects of a character while distribution features take care of statistical distribution of pixels in local regions. Transition count features provide the information such as switching from foreground pixels to background in each horizontal and vertical scan line. As these informations are complimentary to one

another, we have decided to combine the features. To create combinational features, we concatenate topological features, distribution features, transition features and LBP features.

4.6 Wavelet Features

In this section, we present the application of wavelet transforms in the domain of handwritten character recognition. To attain high recognition rate, robust feature extractors that are invariant to degree of variability of human writing are needed.

Wavelet transforms are efficient tool for feature extraction and has been used in literature for the recognition of hand printed characters and numerals [42-44]. But all these works deal with only few classes as opposite to the present large class classification problem. Moreover, Haar wavelets are used for the first time for the recognition of handwritten Malayalam characters.

The wavelet coefficients of an image have multi-resolution representation of original image. The aim of the transform is to extract significant information from a pattern. A wavelet transform decomposes an image of a character into a set of different resolution sub-images, corresponding to the various frequency bands. This results in space frequency localization which is helpful for extracting relevant features. The coarse resolution wavelet coefficients normally represent the overall shape of the image, while the fine resolution coefficients represent the

details of the image. The space frequency localization and multi-resolution analysis capability of a wavelet makes it an efficient tool in analysing images.

4.6.1 Wavelet Theory

The Discrete Wavelet Transform (DWT) is obtained by filtering an image through a series of digital filters at different scales. Discrete wavelet transform can be obtained using the analysis filters for decomposition and the synthesis filters for reconstruction. As we are interested in obtaining the features for classification purpose, we are dealing with only the analysis filter. The scaling function $\varphi(\mathbf{x})$ and the wavelet function $\Psi(\mathbf{x})$ associated with the scaling filter h_φ and the wavelet filter h_ψ are:

$$\varphi(x) = \sum_n h_\varphi(n) \sqrt{2} \varphi(2x - n) \quad 4.5$$

$$\Psi(x) = \sum_n h_\psi(n) \sqrt{2} \varphi(2x - n) \quad 4.6$$

In two-dimensional wavelet decomposition, the analysis scaling function can be written as the product of two one-dimensional scaling functions $\varphi(\mathbf{x})$ and $\varphi(\mathbf{y})$.

$$\varphi(x, y) = \varphi(x) \cdot \varphi(y) \quad 4.7$$

If $\Psi(\mathbf{x})$ is the one-dimensional wavelet associated with the scaling function, then, the three two-dimensional analysis wavelets are defined as:

$$\begin{aligned}\Psi^H(x, y) &= \Psi(x) \cdot \varphi(y) \\ \Psi^V(x, y) &= \varphi(x) \cdot \Psi(y) \\ \Psi^D(x, y) &= \Psi(x) \cdot \Psi(y)\end{aligned}\tag{4.8}$$

where $\Psi^H(x, y)$, $\Psi^V(x, y)$ and $\Psi^D(x, y)$ correspond to horizontal, vertical and diagonal wavelets respectively.

The multi-resolution technique can be implemented using sub-band decomposition in which the image of a character is decomposed into wavelet coefficients [145]. The rows and columns of the original image is convolved with low pass filter \mathbf{h}_φ and high pass filter \mathbf{h}_Ψ followed by decimation by a factor of two in each direction to generate lower scale components namely low-low (LL), and low-high (LH), high-low (HL) and high-high (HH) sub-images. Three of them, LH, HL and HH correspond to the high resolution wavelet coefficients in the horizontal, vertical and diagonal directions respectively. LL image is the approximation of the original image and all the four of them contain one-fourth of the original number of samples. Fig. 4.6 explains the decomposition, in which $j+1$ stands for the starting scale, m and n are row and column directions.

As images are very rich in low frequency content, further analysis can be done by decomposing low pass filtered version of the image as termed as dyadic partitioning. The size of the input image and the size of the image at different levels of decompositions are illustrated in Fig. 4.7

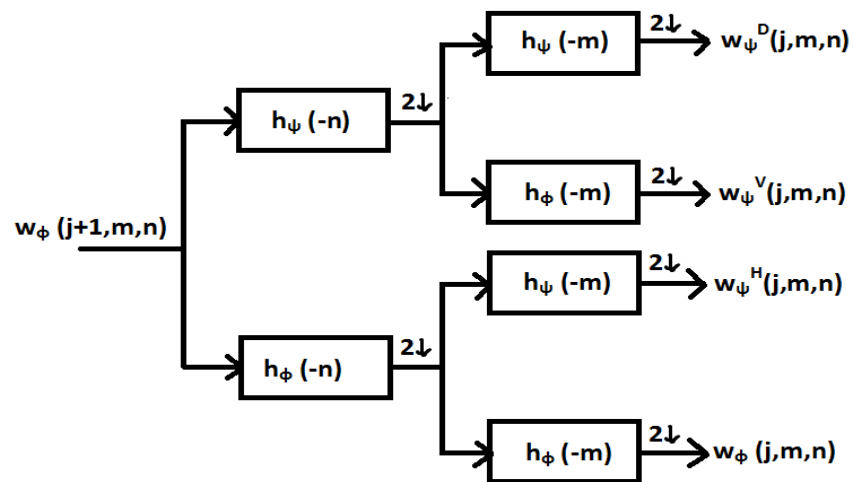


Fig. 4.6 Decomposition using analysis filter bank

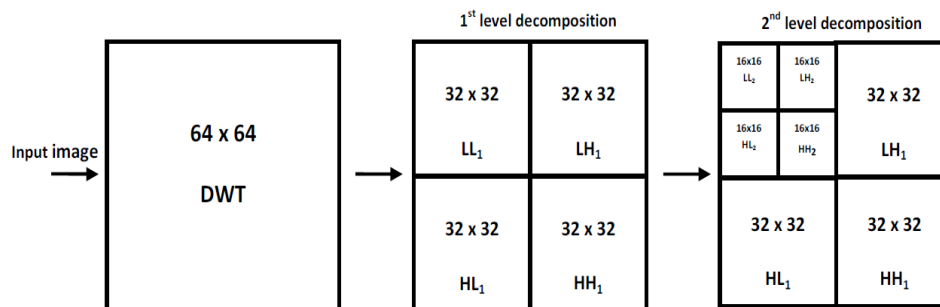


Fig. 4.7 Second-level decomposition of the input image

4.6.2 Haar Wavelets

Haar wavelets [146] have been used for multi-resolution feature extraction. The well-known Haar wavelet is adequate for local detection of edge segments. It enables to have an invariant interpretation of the

character image at different resolutions and presents a multi-resolution analysis in the form of coefficient matrices. Since the details of character image at different resolutions generally characterize different physical structures of the character, and the coefficients obtained from wavelet transform are very useful in recognizing unconstrained handwritten characters [147]. This wavelet was introduced by Hungarian mathematician Alfred Haar in 1910 and it is one of the earliest wavelet with low computing requirements, which is also known as a compact orthonormal wavelet transform.

The Haar scaling function $\boldsymbol{\varphi}(\mathbf{x})$ and the Haar wavelet function $\boldsymbol{\Psi}(\mathbf{x})$ are as follows:

$$\boldsymbol{\varphi}(x) = \begin{cases} \mathbf{1} & \mathbf{0} \leq x < 1 \\ \mathbf{0} & \textit{otherwise} \end{cases} \quad 4.9$$

$$\boldsymbol{\Psi}(x) = \begin{cases} \mathbf{1} & \mathbf{0} \leq x < \frac{1}{2} \\ -\mathbf{1} & \frac{1}{2} \leq x < 1 \\ \mathbf{0} & \textit{otherwise} \end{cases} \quad 4.10$$

4.6.3 Feature Extraction using Wavelet

The original image is represented in binary as well as in gray scale and then size normalized to 64×64 pixels. The Haar wavelet decomposition with Haar analysis filter $\mathbf{h}_{\boldsymbol{\varphi}} = [0.707107, 0.707107]$ and $\mathbf{h}_{\boldsymbol{\Psi}} = [0.707107, -0.707107]$ are applied to each character image to yield

four 32×32 sub images at LL_1 , LH_1 , HL_1 and HH_1 . During the next level decomposition, it yields a 16×16 image $\{LL_2, LH_2, HL_2$ and $HH_2\}$ and then an 8×8 image $\{LL_3, LH_3, HL_3$ and $HH_3\}$. From this decomposition, approximation and detailed coefficients are taken as feature vectors. The method of extracting wavelet features are explained in Algorithm 4.3. The character image അ (/a/) and its decomposition into 3 levels are displayed in Fig. 4.8

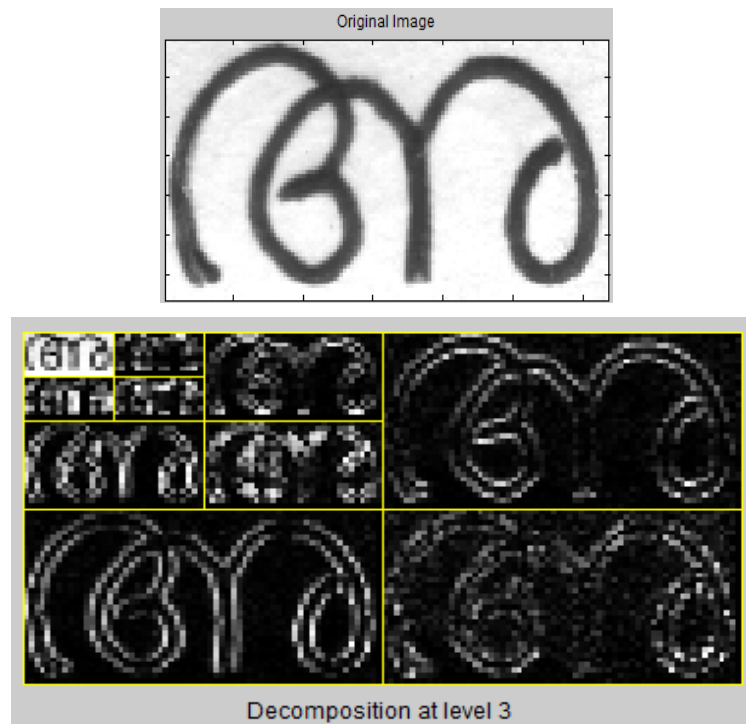


Fig. 4.8 Original image and decomposition at level 3 of character അ (/a/)

Algorithm 4.3: Extraction of Wavelet Features

Input: Gray scale image and binary image

Output: 10 Sets of Wavelet features

- Step 1: Represent the input image in a) gray scale b) binary*
- Step 2: Apply Haar wavelet transform to each character image up to three levels of decomposition.*
- Step 3: Compute approximation coefficients at decomposition level 3 (LL_3 subband) giving 64 features i) from gray scale image ii) from binary image*
- Step 4: Compute approximation coefficients at decomposition level 2 (LL_2 subband) giving 256 features i) from gray scale image ii) from binary image*
- Step 5: Compute detailed coefficients at decomposition level 3 (HL_2 LH_2 HH_2) giving 192 features i) from gray scale image ii) from binary image*
- Step 6: Compute horizontal detailed coefficients at decomposition level 2 giving 256 features i) from gray scale image ii) from binary image*
- Step 7: Compute vertical detailed coefficients at decomposition level 2 giving 256 features i) from gray scale image ii) from binary image*

4.7 Chain Code Features

Chain code based approach is an efficient way to represent the boundary of characters. It can be applied only to binary images. Information such as changes in the direction, cycles, length of the strokes etc. can be obtained from chain code. It represents the boundary by a connected sequence of straight line segments of specified length and direction. In this approach, an arbitrary shape of a character pattern is represented by a sequence of small vectors of unit length and a limited set of possible directions. Starting from a selected point, every consecutive point on the boundary is represented by a chain code showing the transition needed to go from the current point to the next point. The chain can proceed in clock wise or anti clockwise manner until starting point is revisited or end of the boundary is reached. Basically, there are two type of chain code: (i) 4-directional chain codes and (ii) 8-directional chain codes as illustrated in Fig. 4.9 (a) and (b). In 4-directional chain code, each transition is represented by 0,1,2,3 and in 8-directional chain code, it is represented as 0,1,2,3,4,5,6,7. It is obvious that the 4-directional chain code will not take the diagonal direction whereas the 8-directional chain code takes the diagonal direction into consideration. Hence, an 8-directional chain code can represent the shape more efficiently than a 4-directional chain code.

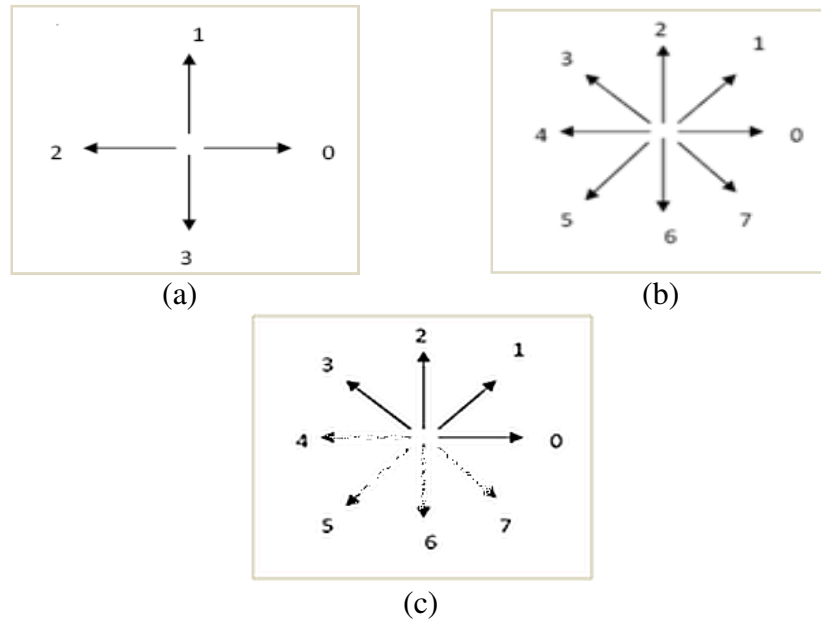


Fig. 4.9 Types of chain codes: (a) 4-directional chain code (b) 8-directional chain codes (c) 4-orientational chain code decomposed from 8-directional chain code

Even though chain code representation was initially proposed by Freeman [148] for encoding arbitrary geometric shapes, chain code based features have shown effective performance in the recognition of character shapes [74, 87, 149].

For extracting chain code features, the preprocessed character pattern is converted to binary and this binary image is represented by its contour, which is a sequence of consecutive boundary points. Contour representation can provide an idea about the shape and thickness of the character. The contour is partitioned into a number of equal sized zones to capture local information from each zone. The direction of each segment

is coded as eight directional chain code $C_i, i = 0,1, \dots,7$ starting with bottom most, left most point in the contour. Chain code histogram is calculated from each zone z using Eq. 4.11. where $n = 7$.

$$CCH_z = \sum_{i=0}^n C_i \quad 4.11$$

Instead of expressing the directional features in terms of 8 directions, we reduce the features into 4 directions: horizontal direction code (direction 0 and 4), vertical direction code (direction 2 and 6), forward diagonal direction code (direction 1 and 5), backward diagonal direction code (direction 3 and 7). Thus, four features representing the histogram in these four directions are obtained.

Since the directions along the contour can be effectively quantized into one of the four possible orientations: 0 or 4, 1 or 5, 2 or 6 and 3 or 7 (Fig. 4.9(c)), the histogram of each zone reduces to four components. Furthermore, as skeleton of a character eliminates writer dependent variations, we have decided to extract chain code histogram features on the skeleton representation of the pattern also. The character skeletons of the patterns are obtained as specified in Section 3.8.4 of Chapter 3. Schematic diagram of the feature extraction method is depicted in Fig. 4.10 and algorithm for extraction of features is depicted in Algorithm 4.4.

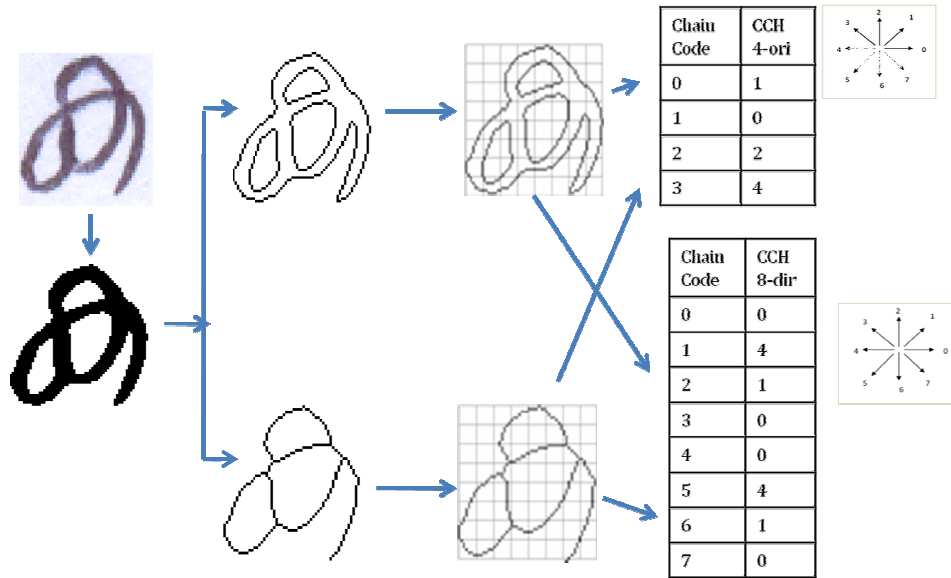


Fig. 4.10 Schematic diagram of chain code based feature extraction method

Algorithm 4.4 : Extraction of Chain Code Features

Input: Binary images

Output: 12 sets of chain code features

Step 1: Apply boundary extraction algorithm for each input character image to obtain contour representation.

Step 2: Apply fixed zoning to divide the image into $w \times w$ zones, where w can be 2,4 or 8. This yields $2 \times 2 = 4$ zones, $4 \times 4 = 16$ zones or $8 \times 8 = 64$ zones.

Step 3: For each zone, obtain

a. 8-directional chain code histogram and output $8 \times w \times w$ feature values along with their class-id

b. 4-orientational chain code histogram and output $4 \times w \times w$ feature values along with their class-id

Step 4: Repeat Steps 2 to 3 for skeleton representation of the character pattern

4.8 Gradient Features

The direction of the character edges can be measured by gradient of image intensity. Unlike chain code features, gradient features can be computed from binary images as well as gray scale images. In this section, we will first define what gradient is and then explain how gradient of a pixel is computed. Finally, we will present the feature extraction method using gradient. Gradient features are one of the efficient directional features successfully used on character recognition problems [150-153]. Contrary to the works reported in literature, we have developed another feature vector by merging two opposite directions, to create orientation plane.

4.8.1 Definition of Gradient

The gradient of an image f at location (x, y) is defined as a vector $[g_x \ g_y]^T$ pointing to the direction of the greatest rate of change of f . Strength (G) and direction (φ) of gradient at location (x, y) is defined as,

$$G(x, y) = \sqrt{g_x^2 + g_y^2} \quad 4.12$$

$$\varphi(x, y) = \tan^{-1} \left(\frac{g_y}{g_x} \right) \quad 4.13$$

To reduce the computational complexity, the strength G at each location (x, y) is approximated using Eq. 4.14

$$G(x, y) \cong |g_x| + |g_y| \quad 4.14$$

4.8.2 Computation of Gradient

Gradient operators templates defined by Sobel [154] is used to compute g_x and g_y components of G and φ . The two gradient components at location (x, y) are approximated as:

$$g_x = f(x+1, y-1) + 2f(x+1, y) + f(x+1, y+1) \\ - f(x-1, y-1) - 2f(x-1, y) - f(x-1, y+1) \quad 4.15$$

$$g_y = f(x-1, y+1) + 2f(x, y+1) + f(x+1, y+1) \\ - f(x-1, y-1) - 2f(x, y-1) - f(x+1, y-1) \quad 4.16$$

These equations can be implemented over the entire image by filtering f with two masks in Fig. 4.11.

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

Horizontal Template

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

Vertical Template

Fig. 4.11 Sobel horizontal and vertical operators

Fig. 4.12 displays gradient strength and direction of character pattern Om (/a/) using the filter masks of Fig. 4.11.

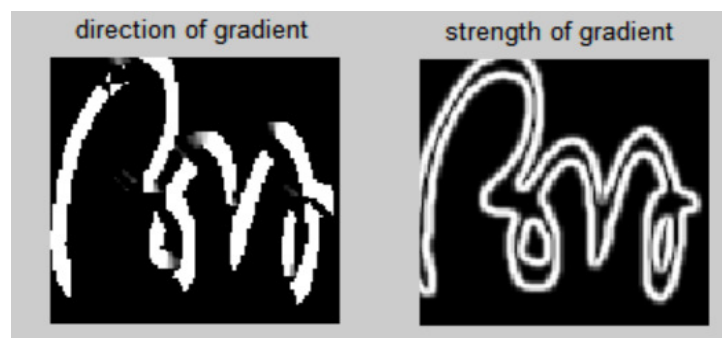


Fig. 4.12 Direction and strength of gradient of character pattern Om (/a/)

4.8.3 Feature Extraction Using Gradient

To extract gradient feature, we first use 3×3 Sobel operators to obtain the horizontal and vertical gradient at each image pixel. The gradient direction $\varphi(x, y)$ and gradient strength $G(x, y)$ is calculated

using Eq. 4.13 and Eq. 4.14. The input gray scale image is divided into $w \times w$ zones through fixed zoning for extracting local characteristics instead of global characteristics. The D directions with an equal interval of $2\pi/D$ is defined and the gradient direction is decomposed into its two nearest directions in a parallelogram manner as demonstrated in Fig. 4.13. The strength of the gradient accumulated in these D directions is calculated for each zone. The accumulated gradient in each zone is concatenated and applied a variable transformation [4] to create the feature vector. In our feature extraction method, we set D to 8 and w to 4. Algorithm 4.5 explains feature extraction method using gradient information.

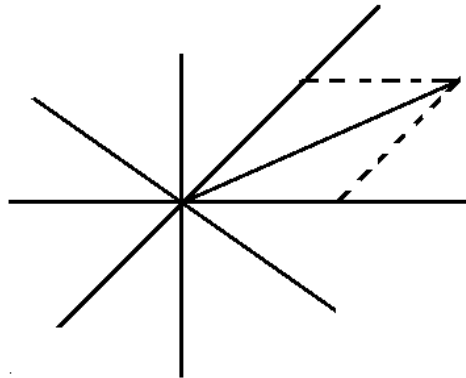


Fig. 4.13 Decomposition of gradient direction

Algorithm 4.5: Extraction of Gradient Features

Input: Gray scale images

Output: Gradient feature vectors

Step 1: Apply fixed zoning to divide the input image into 4×4 zones

Step 2: For each pixel in each zone, compute the gradient strength $G(x,y)$ using Eq. 4.14 gradient direction $\varphi(x,y)$ using Eq. 4.13, where gradient components are computed using Eq. 4.15 and Eq. 4.16

Step 3: Partition the range of gradient direction into 8 standard directions with $\pi/4$ interval as 0^0 , 45^0 , 90^0 , 135^0 , 180^0 , 225^0 , 270^0 and 315^0 . The gradient directions that lie in between these directions are mapped to the nearest standard direction in a parallelogram manner.




Step 4: Calculate the strength of gradient accumulated in these 8 gradient directions in each zone as follows:

$$F_d = \sum_d G(x,y), \text{ where } F_d \text{ is feature associated to the direction } d \in \{0^0, 45^0, 90^0, 135^0, 180^0, 225^0, 270^0, 315^0\}$$

Step 5: Create the local gradient direction histogram with $4 \times 4 \times 8 = 128$ features and apply a variable transformation $y = x^{0.4}$ to make distribution as Gaussian

We create another set of gradient feature vector using orientation plane instead of direction plane. In step 4 of Algorithm 4.5, we are getting 8 directions. Since 0^0 and 180^0 is providing the similar information, these two directions are merged into one. The same process is repeated for 45^0 and 225^0 ; 90^0 and 270^0 ; 135^0 and 315^0 . The F_d feature of Step 5 is recalculated with $d \in \{0^0, 45^0, 90^0, 135^0\}$. Now instead of 8 directions, there are only 4 orientations and the dimension of the feature becomes 64.

4.9 Curvature Features

Many characters in Malayalam language have curved like shapes. Some of the examples include: ,  and . Based on this information, we have decided to develop features based on curvature.

4.9.1 Definition of Curvature

Curvature is the amount by which a geometric object deviates from being straight. It is calculated using bi-quadratic interpolation method as described by Shi et al. [151]. The curvature c at x_0 is defined as:

$$c = \frac{y''}{\sqrt{(1 + y'^2)^3}} \quad 4.17$$

where $y = g(x)$ is the curve passing through x_0 , (x, y) is the spatial coordinates of x_0 , y' and y'' are the first and the second order derivatives of y respectively. The derivatives of y' and y'' are derived from bi-quadratic interpolating surface for the gray scale values in the 8-neighborhood of x_0 . Fig. 4.14 shows the 8-neighbors of x_0 and the pixel values in the neighborhood where f_k denote the pixel value at x_k

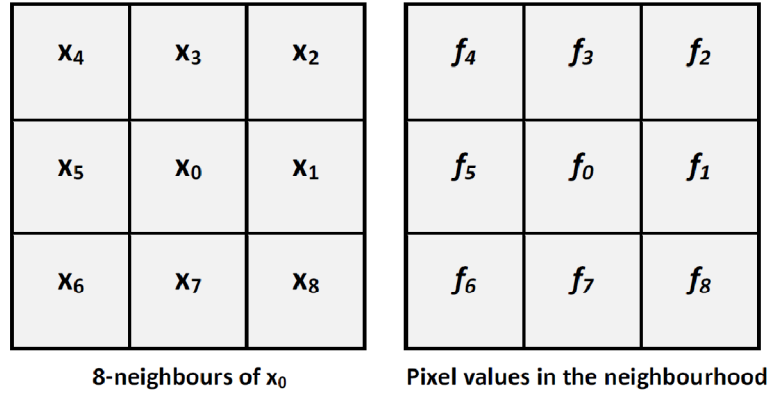


Fig. 4.14 8-neighbours of x_0 and pixel values in the neighborhood

The bi-quadratic surface is given by

$$z = [1 \quad x \quad x^2] \begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} 1 \\ y \\ y^2 \end{bmatrix} \quad 4.18$$

The curve passing through x_0 is defined as,

$$f_0 = a_{00} + a_{10}x + a_{20}x^2 + y(a_{01} + a_{11}x + a_{21}x^2) + y^2(a_{02} + a_{12}x + a_{22}x^2) \quad 4.19$$

where f_0 is the pixel value at x_0 . Differentiating both sides with respect to x and substituting the value $(0, 0)$ of x_0 , the values of y' and y'' at x_0 are given by

$$y' = -a_{10}/a_{01}$$

$$y'' = -2(a_{10}^2 a_{02} - a_{01} a_{10} a_{11} + a_{01}^2 a_{20})/a_{01}^3 \quad 4.20$$

Solving for 8-neighbours of x_0 , the coefficients of the bi-quadratic surface are given by,

$$\begin{aligned}a_{10} &= (f_1 - f_5)/2 \\a_{20} &= (f_1 + f_5 - 2f_0)/2 \\a_{01} &= (f_3 - f_7)/2 \\a_{02} &= (f_3 + f_7 - 2f_0)/2 \\a_{11} &= (f_2 - f_8) - (f_4 - f_6)/4\end{aligned}\tag{4.21}$$

Substituting the values of y' and y'' in Eq. 4.17 the curvature can be calculated as

$$c = -2(a_{10}^2 a_{02} - a_{01} a_{10} a_{11} + a_{01}^2 a_{20}) / (a_{10}^2 + a_{01}^2)^{3/2}\tag{4.22}$$

4.9.2 Feature Extraction using Curvature

To compute the curvature feature, the value of curvature is computed using Eq. 4.22 for each pixel in the preprocessed input image. According to the curvature value, the image is divided into concave region, linear region or convex region (Fig. 4.15), using a threshold t . The threshold t is chosen to be 0.15. The rest of the procedure is similar to gradient feature. The strength of gradient accumulated in 8 gradient directions in each zone for each curvature levels is used as the curvature feature. The detailed steps for extracting curvature feature is depicted in Algorithm 4.6

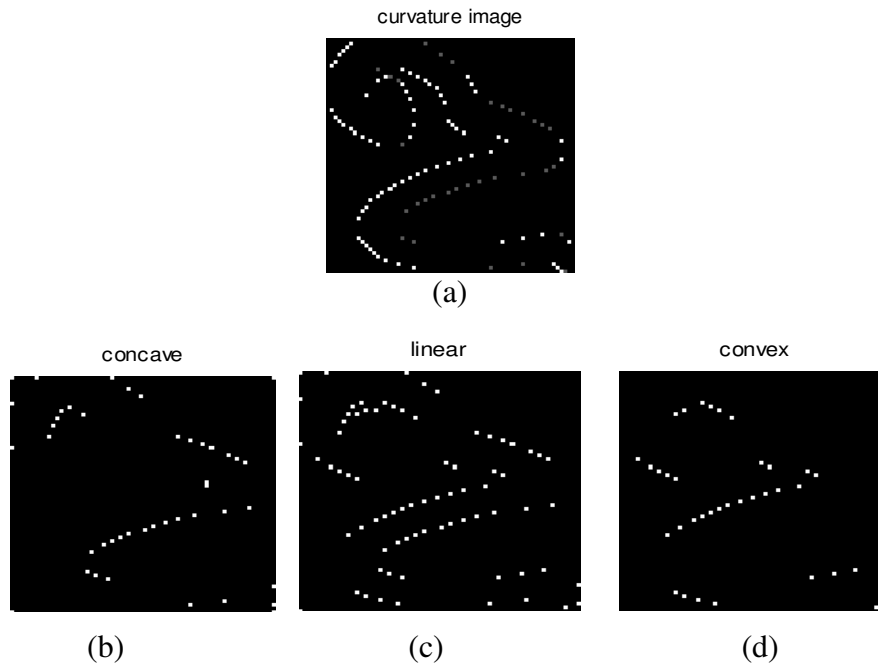


Fig. 4.15 Curvature image of handwritten pattern la (/la/) and its division into three regions (a) curvature image (b) concave region (c) linear region (d) convex region

Algorithm 4.6 : Extraction of Curvature Features

Input: Gray Scale Image

Output: Curvature Features

Step 1: Apply fixed zoning to divide the input image into 4×4 zones for extracting local characteristics instead of global characteristics

Step 2: For each pixel in each zone, compute the curvature c using Eq. 4.22 to obtain the curvature image

- Step 3: Divide the curvature image into three levels to get concave, linear and convex regions using a threshold t . For concave region $c \leq -t$; for linear region $-t < c < t$ and for convex region $c \geq t$.*
- Step 4: For each pixel in each zone, compute the gradient strength $G(x, y)$ using Eq. 4.14 gradient direction $\varphi(x, y)$ using Eq. 4.13, where gradient components are computed using Eq. 4.15 and Eq. 4.16*
- Step 5: Partition the range of gradient direction into 8 standard directions with $\pi/4$ interval as $0^\circ, 45^\circ, 90^\circ, 135^\circ, 180^\circ, 225^\circ, 270^\circ$ and 315° . The gradient directions that lie in between these directions are mapped to the nearest standard direction.*
- Step 6: Calculate the strength of gradient accumulated in these 8 gradient directions in each zone for each curvature levels*
- Step 7: Create curvature feature vector with $4 \times 4 \times 8 \times 3 = 384$ features and apply a variable transformation $y = x^{0.4}$ to make distribution as Gaussian*

4.10 Design of a Novel Feature Descriptor Based on Image Gradient

The gradient feature descriptor described in Section 4.8 captures the directional information providing weight in the horizontal and vertical direction. Instead of just finding gradient information in horizontal and

vertical directions, another operator that provides emphasis on the forward and backward diagonal direction is also needed. Diagonal operator can be created using Robert's [151, 155] or Sobel's. The better noise-suppression characteristics of Sobel masks make it preferable for extracting features from gradients. Moreover, this operator involves more neighbors than Robert's and hence generates better results. For this, filter masks in Fig. 4.11 are modified to reflect strong responses along the diagonal directions (Fig. 4.16) and it is computed as:

$$g_x = f(x-1, y) + 2f(x-1, y+1) + f(x, y+1) - f(x, y-1) - 2f(x+1, y-1) - f(x+1, y) \quad 4.23$$

$$g_y = f(x, y+1) + 2f(x+1, y+1) + f(x+1, y) - f(x, y-1) - 2f(x-1, y-1) - f(x-1, y) \quad 4.24$$

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

Forward Diagonal Template

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

Backward Diagonal Template

Fig. 4.16 Sobel diagonal operators

Using these templates, we have developed a novel feature descriptor, say SSG (Sobel-Sobel Gradient) which has more discriminating power for separating similar character patterns. This feature descriptor is created by convolving each pixel neighbourhood with all the four templates in Fig. 4.11 and Fig. 4.16. Plot of this feature descriptor for two typical samples of class 1 and class 74 are displayed in Fig. 4.17. As SSG gives importance to horizontal, vertical and both diagonal directions, its discriminating power gets enhanced. The total number of operations per pixel is just $2(3 \times 3) + 2(3 \times 3) = 36$. The only problem with this feature is its dimension, which could be controlled by the zone size and the number of directions. Zone size should be small enough to capture the local directional characteristics of the image yielding a large dimension. But this dimensionality can be reduced using dimensionality reduction technique as specified in Section 4.11. Even though these features could be extracted from gray scale as well as binary images, we conducted the experiments in gray scale so as to avoid artefacts introduced due to binarization. Box-Cox (variable) transformation $y = x^p$ was applied to the feature set as it improves classification performance, where the value of p was selected from the range [0.1 0.2 ... 0.9] and experimentally optimized to 0.4. The algorithm can be summed up in three steps:

Algorithm 4.7: Creation of SSG Features

Input: Gray scale images

Output: SSG feature descriptor

Step 1: Compute Sobel horizontal-vertical template based gradient features

Step 2: Compute Sobel forward-backward diagonal template based gradient features

Step 3: Concatenate the features obtained from Step 1 and Step 2 and apply variable transformation $y = x^{0.4}$

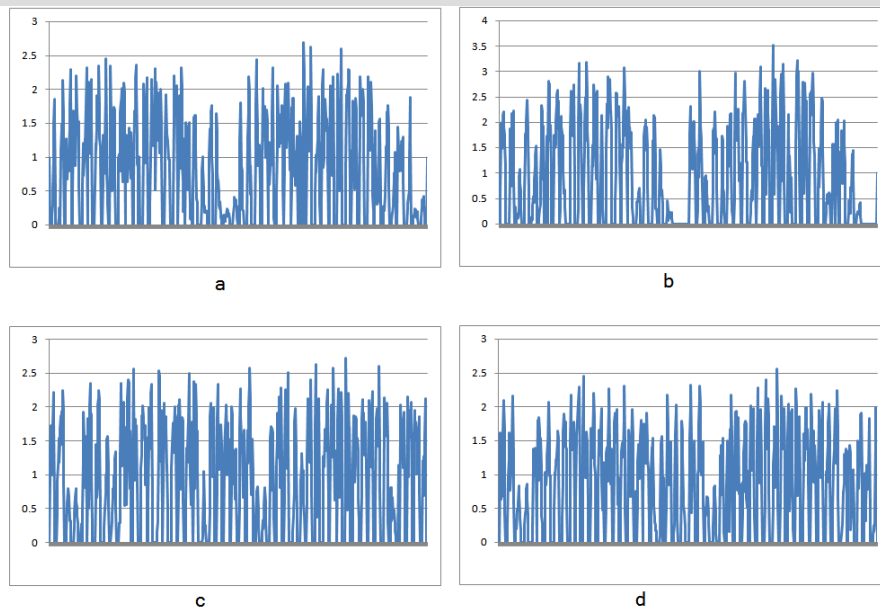


Fig. 4.17 SSG Feature vectors of two typical samples of $\text{അ$ (/a/) and വ്വ (/vva/) (a & b) samples of character അ (/a/) of class 1; (c & d) samples of character വ്വ (/vva/) of class 74

4.11 Reduction of Feature Dimension

High dimensional features provide more information about the shape of the character pattern. However it is not desirable because as the dimensionality increases, the size of data required for training the system and the amount of computation required for recognition grows exponentially. The aim of feature dimension reduction techniques is to capture the essential information of the high dimensional feature vector into a set of smaller number of relevant features. It ensures the reliability of the decision making procedure by removing the redundant and irrelevant information [4].

Principal Component Analysis (PCA) is one of the simplest and most robust ways to reduce the dimensionality of a data set. It is a very well known and efficient orthogonal linear transformation. It reduces the dimension of feature space and the correlation among the feature vectors by projecting the original feature into a smaller subspace through transformation. The goal of PCA is to find a new set of variables, called principal components, which are uncorrelated, and which are ordered so that the first few retain most of the variation present in the original set of variables [156]. The first dimension is chosen to capture as much variability as possible. The second dimension is orthogonal to the first, and, subject to that constraint, captures as much the remaining variability as possible, and so on [157]. Fig. 4.18 [158] shows the first two principal

components Y_1 and Y_2 for the given set of data originally mapped to the axes X_1 and X_2 .

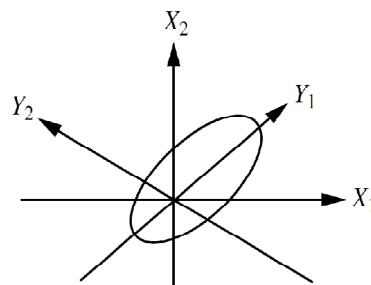


Fig. 4.18 Principal Component Analysis: Y_1 and Y_2 are the first two principal components for the given data

PCA has several characteristics. First, it identifies the strongest patterns in the data. Hence it can be used as a pattern finding technique. Second, most of the variability of the data can be captured by a small fraction of the total set of dimensions. Therefore, dimensionality reduction using PCA can result in low dimensional data. Third, since the noise in the data is weaker in the patterns, dimensionality reduction can eliminate much of the noise [157].

Dimensionality reduction is concerned with mathematical tool for reducing the size of features. Given m by n feature matrix D , whose m rows are samples and n columns are features, the covariance matrix of D is the matrix Z , which has entries z_{ij} defined as $z_{ij} = \text{covariance}(d_{*i}, d_{*j})$. In other words, z_{ij} is the covariance of the i^{th} and j^{th} columns of the data. The covariance of two features is a

measure of how strongly the attributes vary together. The goal of PCA is to find a transformation of data that satisfies the following properties [157]:

- Each pair of new variables has zero covariance
- The variables are ordered with respect to how much of the variance of the data each variable captures
- The first variable captures as much of the variance of the data as possible
- Subjective to the orthogonality requirement, each successive variable captures as much of the remaining variance as possible

The criteria used for deciding how many components to extract are (1) eigen value criterion (2) proportion of variance explained criterion, (3) the minimum communality criterion, and (4) the scree plot criterion [159] where a scree plot is a simple line segment plot that shows the fraction of total variance in the data as explained or represented by each principal component.

For the proposed features, implementation of PCA on this feature space efficiently reduces the feature dimension without losing much information. Hence, PCA is employed to reduce the dimension of the proposed feature space in our work. No earlier attempts were found to reduce the dimensionality in Malayalam handwritten character recognition.

4.12 Conclusion

In this chapter, we have demonstrated the sets of features suitable to represent the shape of Malayalam handwritten character images. It includes topological features, distribution features, transition count features, LBP features, wavelet features, chain code features, gradient features and curvature features.

Based on the shapes of Malayalam handwritten characters, suitable topological features which describe the salient characteristics of the character patterns such as presence of loops, number of end points, number of branch points, number of cross points and the ratio of width to height are calculated. We have identified the distribution of pixels in the normalized image by counting the number of pixels in different zones.

Transition count features provides the information about the transition of pixels from foreground pixels to background pixels, which is useful information hidden in the images. Local binary pattern is a texture descriptor. We have introduced this feature for the recognition of handwritten characters and created a novel feature descriptor based on LBP.

Wavelet features work in the transform domain and Haar wavelets provide more information about the edges of character. We have extracted multi-resolution features from Haar wavelets both from gray scale image

and from binary image to identify which representation is more suitable for the problem.

Chain codes features provide directional information of a character pattern. We have created six sets of chain code features from contour representation and another six sets from skeleton representation of binary images. Gradients features also provide directional information in continuous direction, compared to discrete directions in chain code features. We have created gradient features based on Sobel operators. Many characters in this script have curved like shapes. Due to this nature of Malayalam characters, curvature feature is developed.

Using Sobel gradient templates in horizontal, vertical, forward diagonal and backward diagonal directions, we have developed a novel feature descriptor which has more discriminating power for separating character patterns. We have demonstrated Principal Component Analysis technique useful for reduction of dimension of feature space. The performance of these features needs to be evaluated for designing a recognition system and it is the main focus of the next chapter.

Chapter 5

PERFORMANCE EVALUATION USING DIFFERENT CLASSIFIERS

Contents

5.1 Introduction
5.2 Classification Algorithms
5.3 Performance Evaluation Measures
5.4 Performances Obtained Using Different Feature Sets and Classifiers
5.5 Conclusion

5.1 Introduction

The final step on the recognition of the handwritten characters is the classification process. A classification problem occurs when an object needs to be assigned to a predefined group or class label based on number of observed attributes related to that object. A classifier learns from a training set containing a large number of already labeled objects of each class. The objective of any classifier is to find out the correct class label of unknown character pattern, once its features have been identified. The performance of a classifier depends on the nature of the problem and also on the selection of features created from the data set. The recognition accuracy as well as the learning and testing speed are the crucial aspects

in choosing a classifier. We have used supervised classification methods in which learning is directed by labeled objects.

5.2 Classification Algorithms

For classification, k-Nearest Neighbour (k-NN), Artificial Neural Networks (ANN), and Support Vector Machines (SVM) are successfully applied in character recognition problems. k-Nearest Neighbour classification is one of the most fundamental and simple classification methods and should be one of the first choices for a classification study when there is little or no prior knowledge about the distribution of the data. Artificial Neural Networks are popular since late 1980s [160]. But SVMs are increasingly being used from late 1990s [161] in classification. In many cases SVM outperforms classical neural networks in classification problems [162]. One advantage of SVM over neural networks is that the learning task is insensitive to the training samples and also works well with high dimensional data. SVM provides generalization performance through structural risk minimization as opposed to empirical risk minimization of ANN. Generalization performance refers to the fact that a classifier not only has good accuracy on the training data, but also guarantees high predictive accuracy for the unseen data from the same distribution as the training data [163]. SVM can achieve high generalization accuracies when trained with large number of samples [8]. More recently, a new classifier, namely, Extreme Learning Machine (ELM) [24], is available for training of single layer feed forward neural

network. ELM randomly chooses input weights and analytically determines the output weights of the network. In theory, this algorithm tends to provide a good generalization performance with much shorter learning time. ELM has been applied to a wide variety of problems [164-168] as a promising classifier. In this work, we have introduced optimization based ELM for recognition of handwritten characters.

5.2.1 k -Nearest Neighbour (k -NN)

k -Nearest Neighbour is a non parametric lazy learning algorithm in the sense that it does not make any assumption on the underlying data distribution and they do not need any training or build model apriori. The number k stands for the number of neighbours used in classification. The drawback of this method is the high computational cost when the classification is conducted as all the training data are needed during the test phase. But k -NN work well with arbitrary number of classes.

When $k=1$, the algorithm is termed as nearest neighbour algorithm. The nearest neighbour rule allocates the incoming pattern x to k if the closest sample, x_c , in the training set is with the label k [4].

$$\mathbf{x}_c = \mathbf{arg} \min_i \{d(\mathbf{x}, \mathbf{x}_i)\}, i = 1, 2, \dots, N \quad 5.1$$

The distance measure between the unknown pattern and the training patterns has a general quadratic form:

$$d(\mathbf{x}, \mathbf{x}_k) = (\mathbf{x} - \mathbf{x}_k)^T \mathbf{M} (\mathbf{x} - \mathbf{x}_k) \quad 5.2$$

with $M = \Sigma^{-1}$, the inverse of the covariance matrix in the pattern, the result is the Mahalanobis distance. Euclidean distance is obtained when $M = I$, the identity matrix. In our work, we have used Euclidean distance measure.

With k is more than one, k-NN finds a group of k patterns in the training set that is closer to the test pattern and does a majority voting to make a decision. Typically the value of k is chosen to be odd when the number of classes is two. A similar strategy can be adopted for classifying multi-classes. In majority voting, the vote of each class is counted, and the class having maximum count is predicted to be the class of the test pattern.

The parameter that determines the neighbourhood size, k , is very important to the classification accuracy achieved by k-NN classifier. If k is chosen too small, classification tends to be sensitive to noise and outliers. On the other hand, a too large value for k might cause the result to be affected by too far away objects in the training set [169]. Fig. 5.1 shows 1-, 2- and 3- nearest neighbours of an unknown pattern \mathbf{x} .

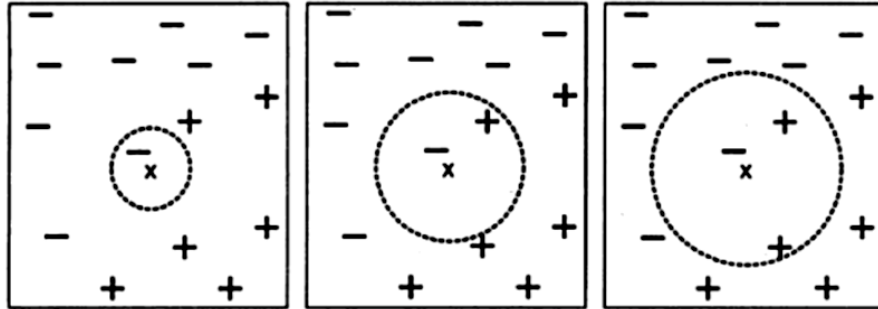


Fig. 5.1 The 1-, 2- and 3- nearest neighbours of an unknown pattern x

5.2.2 Support Vector Machines

Support Vector Machines (SVMs) are one of the most robust and popular techniques for pattern recognition and are considered to be the state-of-the-art tool for linear and non-linear classification. SVMs, which were originally developed for binary classification by Vapnik [170] have empirically good performance in solving many application problems. For a two-class linearly separable learning task, the support vector classifier aims to find the optimal hyperplane in order to maximize the margin of separation. The training vectors that have minimal distance to this margin are called support vectors and thus, the classifier was named support vector machine classifier. Therefore, the number of support vectors rather than the dimensionality of the data characterize complexity of the trained classifier [158].

Given a set of training data $(x_i, y_i), i = 1, \dots, m$ where $x_i \in R^n$ being the instances and $y_i \in \{1, -1\}^m$ being the labels of the training

data, the support vector machine finds the optimal hyperplane that maximises the margin which results in solving Eq. 5.3 [30]. The optimal separating hyperplane in two dimensional space is demonstrated in Fig. 5.2.

$$\min_{w,b} \frac{1}{2} w^T w$$

5.3

subject to $y_i(w^T x_i + b) \geq 1$

where w is the weight vector and b is the bias.

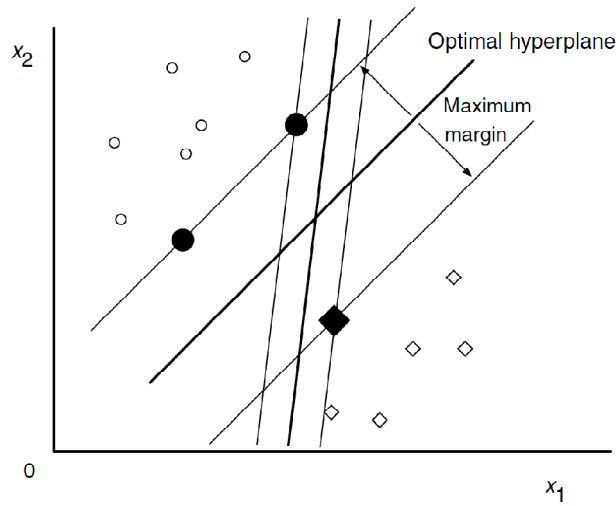


Fig. 5.2 Optimal separating hyper plane in a two-dimensional space

When the training points are not linearly separable (Fig. 5.3), the cost function is reformulated by introducing a slack variable $\xi_i \geq 0, i = 1, \dots, m$ as in Eq. 5.4

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

subject to $y_i(w^T x_i + b) \geq 1 - \xi_i; \xi_i \geq 0$ 5.4

where $C > 0$ is penalty parameter of the error term.

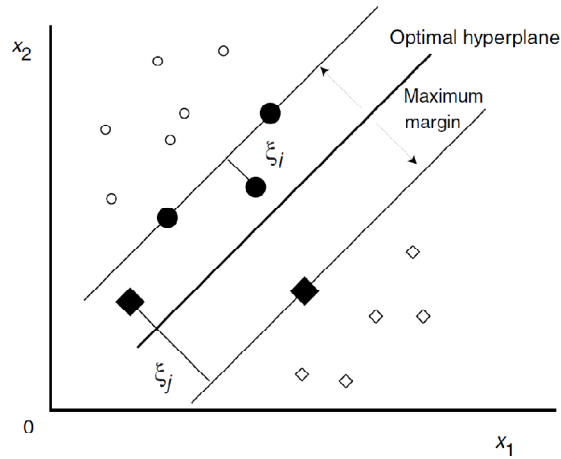


Fig. 5.3 Inseparable case in two-dimensional space

When the decision function is nonlinear Eq. 5.4 cannot be used directly. Instead, SVM performs nonlinear classification using kernel trick in which the training vectors x_i are mapped into a higher dimensional feature space through a nonlinear feature mapping function $\varphi(\cdot)$. SVM then finds a linear separating hyper plane with maximal margin in this higher dimensional feature space by solving the following optimization problem.

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

subject to $\mathbf{y}_i(\mathbf{w}^T \boldsymbol{\varphi}(\mathbf{x}_i) + \mathbf{b}) \geq 1 - \xi_i; \xi_i \geq 0$ 5.5

The commonly used kernels are:

Polynomial kernel with degree d : $\mathbf{k}(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + 1)^d$ 5.6

Radial basis function (RBF): $\mathbf{k}(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2)$ 5.7

where $\gamma > 0$ is the variance parameter of the RBF kernel. Kernels are related to the transform by the equation $\mathbf{k}(\mathbf{x}_i, \mathbf{x}_j) = \boldsymbol{\varphi}(\mathbf{x}_i) \cdot \boldsymbol{\varphi}(\mathbf{x}_j)$. The decision function of binary SVM has the form

$$f(\mathbf{x}) = \text{sign} \left(\sum_{i=1}^N \alpha_i y_i \boldsymbol{\varphi}(\mathbf{x}, \mathbf{x}_i) + \mathbf{b} \right)$$
 5.8

where \mathbf{x}_i is the training data with y_i as its class label, α_i is the Lagrange multiplier to be computed by the learning machines.

Even though, SVM was originally proposed for binary classification, it can be applied to multiclass problems using one-against-all (OAA) and one-against-one (OAO) classification strategy [171]. OAA consists of m SVMs, where m is the number of classes. The i^{th} SVM is trained with all of the samples in the i^{th} class with positive labels and the remaining $m - 1$ classes with negative labels. OAO consists of

$m(m - 1)/2$ SVMs, where each is trained with the samples from two classes only. OAO gives better classification accuracy than OAA [171].

5.2.3 Extreme Learning Machine

A new learning algorithm called extreme learning machine has recently been proposed for single-hidden layer feed forward neural networks (SLFNs) to easily achieve good generalization performance at extremely fast learning speed. ELM proposes to apply random computational nodes in the hidden layer, which may be independent of the training data. It aims to reach the smallest training error but also the smallest norm of output weights. According to ELM theories [24, 172] all the training data are linearly separable by a hyper plane passing through the origin with probability one in the ELM feature space.

The output function of ELM for generalised SLFNs for single output case is

$$f(x) = \sum_{i=1}^L \beta_i h_i(x) = h(x) \quad 5.9$$

where β_i is the output weight from the i^{th} hidden node to the output node and $h_i(x)$ is the output of the i^{th} hidden node with respect to input x ; $\beta = [\beta_1, \dots, \beta_L]^T$; $h(x) = [h_1(x), \dots, h_L(x)]$. $h(x)$ is used to map d -dimensional input space to L -dimensional hidden layer feature space H . The architecture of a single hidden layer neural network which forms the foundation of ELM is shown in Fig 5.4.

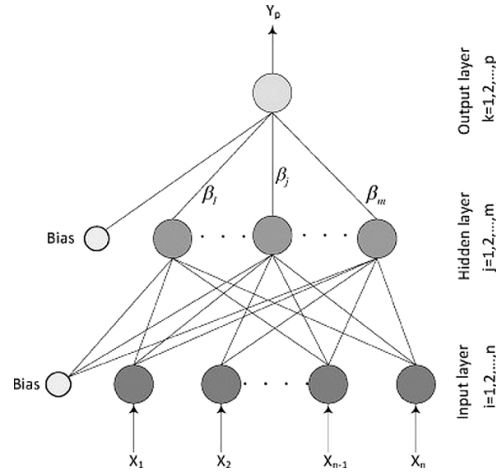


Fig. 5.4 Architecture of single hidden layer neural network

For binary classification applications, the decision function of ELM is

$$f(x) = \mathit{sign}(h(x)\beta) \quad 5.10$$

Different from traditional learning algorithms, ELM minimizes the training error as well as the norm of the output weights [172].

$$\text{Minimize: } \|H\beta - T\|^2 \text{ and } \|\beta\| \quad 5.11$$

where H is the hidden-layer output matrix

$$H = \begin{bmatrix} h(x_1) \\ \vdots \\ h(x_N) \end{bmatrix} \begin{bmatrix} h_1(x_1) & \cdots & h_L(x_1) \\ \vdots & \ddots & \vdots \\ h_1(x_N) & \cdots & h_L(x_N) \end{bmatrix} \quad 5.12$$

Minimizing the norm of the output weights β is actually maximizing the distance of the separating margins of the two different classes in the ELM feature space: $2/\|\beta\|$, where β can be solved as: $\beta = H^\dagger T$ where H^\dagger is the Moore-Penrose generalise inverse of matrix H . Different methods can be used to calculate the Moore-Penrose generalized inverse of a matrix.

There are some advantages of the ELM algorithm: (1) it is extremely fast, (2) it has better generalization performance, (3) it tends to reach the solutions straightforward without trivial issues such as local minima, learning rate, momentum rate and over-fitting encountered in traditional gradient based learning algorithm. The set of training parameters required is listed in Table 5.1.

Table 5.1 Training parameters of ELM network.

Number of layers	Input layer: Number of features
	Hidden layer: 1
	Output layer: 1
	Number of neurons in the hidden layer: 100...4000
Activation functions	Tangent Sigmoid, Sigmoid, Radial basis, Sine
Learning algorithm	ELM for SLFNs

The learning model of ELM can be summarized as follows:

Given a set of training data $(x_i, y_i), i = 1, \dots, n$ where $x_i \in R^N$, $y_i \in R^M$; activation function $f(x)$; number of hidden nodes L

- Randomly assign input weight w_i and bias $b_i, i = 1, \dots, L$
- Compute the hidden layer output matrix H
- Compute the output weight W where $W = H^\dagger Y$ and $Y = [y_1, y_2, \dots, y_n]^T$

In the case of this basic ELM, the only parameter needed is the number of hidden nodes.

5.2.3.1 Constrained Optimization based ELM

The classification problem of ELM with multi output nodes can be formulated as Minimize:

$$L_{PELM} = \frac{1}{2} \|\beta\|^2 + C \frac{1}{2} \sum_{i=1}^N \|\xi_i\|^2$$

$$\text{Subject to: } \mathbf{h}(x_i)\beta = \mathbf{t}_i^T - \xi_i^T, \mathbf{i} = \mathbf{1}, \dots, \mathbf{N} \quad 5.13$$

where $\xi_i = [\xi_{i,1}, \dots, \xi_{i,m}]^T$ is the training error vector of m output nodes with respect to the training sample x_i . For equally constrained optimization based ELM [24], different solutions can be formulated to solve β concerning the efficiency in different size of training datasets. Solving β using a regularization factor C is formulated as Eq. 5.14 or Eq.

5.15, where one can set regularization factor C properly in classification applications [24].

$$\boldsymbol{\beta} = \mathbf{H}^T \left(\frac{\mathbf{1}}{C} + \mathbf{H}\mathbf{H}^T \right)^{-1} \mathbf{T} \quad 5.14$$

$$\boldsymbol{\beta} = \left(\frac{\mathbf{1}}{C} + \mathbf{H}^T \mathbf{H} \right)^{-1} \mathbf{H}^T \mathbf{T} \quad 5.15$$

where

$$\mathbf{T} = \begin{bmatrix} \mathbf{t}_1^T \\ \vdots \\ \mathbf{t}_N^T \end{bmatrix} \begin{bmatrix} \mathbf{t}_{11}^T & \cdots & \mathbf{t}_{1m}^T \\ \vdots & \ddots & \vdots \\ \mathbf{t}_{N1}^T & \cdots & \mathbf{t}_{Nm}^T \end{bmatrix} \quad 5.16$$

If the number of training samples is not huge, one can use Eq. 5.14 and if the number of training samples is much larger than the dimensionality of the feature space, Eq. 5.15 is preferred. In our work, the number of training data is larger than the dimensionality of the feature space. Hence, Eq. 5.15 is applied. In these two cases, the output function of ELM classifier is solved as Eq. 5.16. According to [173], we can write $\mathbf{h}(x) = [G(a_1, b_1, x), \dots, G(a_L, b_L, x)]$ where $G(a, b, x)$ is a nonlinear piecewise continuous function satisfying ELM universal approximation capability theorem and $\{(a_i, b_i)\}_{i=1}^L$ are randomly generated.

For binary classification case, ELM uses a single output node and the class label closer to the input data is chosen as the predicted class label. For multiclass classification there are two solutions (i) ELM uses

only a single output node and among the multiclass labels, the class label closest to the output value is chosen as the predicted class label (ii) ELM uses multi output nodes and the index of the output node with the highest output value or chosen as the predicted class label.

5.3 Performance Evaluation Measures

The *confusion matrix* is a useful tool for analyzing how well a classifier can recognize samples of different classes as it provides information needed to identify how well a classification model performs [158]. Each entry in the confusion matrix, $CF(i, j); i, j = 1, 2, \dots, C$, denotes the number of samples from class i predicted as class j .

The confusion matrix for a two class problem is as shown in Table 5.2, where TP (true positive) is the number of correctly classified positive data, FN (false negative) is the number of misclassified positive data, FP (false positive) is the number of misclassified negative data, and TN (true negative) is the number of correctly classified negative data.

Table 5.2 Confusion matrix for two classes

	Assigned Positive	Assigned Negative
Actual Positive	TP	FN
Actual Negative	FP	TN

When the number of classes is large, it is not easy to visualize the confusion matrix. However, the information provided in the confusion matrix can be expressed to performance metrics such as accuracy,

precision, recall and F-measure. The primary metric for evaluating classifier performance is classification accuracy.

5.3.1 Accuracy

The accuracy of a classifier on a given test set is the percentage of test samples that are correctly classified by the classifier. In the pattern recognition literature, this is also referred to as the overall recognition rate of the classifier as it reflects how well the classifier recognizes samples of the various classes [158].

$$\textit{Accuracy} = \frac{(TP + TN)}{(TP + FP + TN + FN)} \quad 5.17$$

5.3.2 Precision

Precision or positive predictive value is defined as the proportion of predicted positives which are actual positive.

$$\textit{Precision} = \frac{TP}{TP + FP} \quad 5.18$$

5.3.3 Recall

Recall is defined as the proportion of the actual positives which are predicted positive.

$$\textit{Recall} = \frac{TP}{TP + FN} \quad 5.19$$

5.3.4 F-measure

F-measure is the harmonic mean of precision and recall. For F-measure, the maximum possible value is 1. The performance of the classifier is considered to be good if the values are nearer to or equal to 1.

$$F - measure = \frac{2(Precision \cdot Recall)}{(Precision + Recall)} \quad 5.20$$

5.4 Performances Obtained Using Different Feature Sets and Classifiers

We computed five topological features, 128 dimensional transition count features and 128 dimensional LBP features. Using 4 sets of distribution features and five different classifiers, we computed 20 different results. Using five sets of wavelet features, we obtained 25 different results from binary images and 25 different results from gray scale images. Using 12 sets of chain code features and five different classifiers, we computed 60 different results from binary images. 64 dimensional and 128 dimensional features are extracted using gradient. 384 dimensional features are extracted using curvature. 256 dimensional features are extracted using SSG. The following subsection explains the evaluation of all these experiments.

5.4.1 Experimental Setup

For experimentation, the created benchmark database as specified in Chapter 3, Section 3.6 is used. It contains all the 44 basic characters; 5 chillu (pure consonants), 26 compound characters and 15 vowel-consonant signs. Altogether, 90 classes were taken. For uniform distribution of samples per class, 150 samples per class are randomly chosen to create the training database and from the remaining set, 50 samples are randomly chosen to create the testing set. Training set contains 13500 samples and test set contains 4500 samples and the size of the whole dataset is 18000. Random permutations of samples are performed both on training set and testing set. The training set and testing set are disjoint and the same training set and testing set are used throughout the experiments.

For k -NN, we experiment with values of $k = 1, 3, 5, 7, 9$ and 15. The attributes have been scaled to prevent distance measures from being dominated by one of the attributes.

We have used LIBSVM software [174] for learning the SVM classifier, which supports multiclass classification with one against one (OAO) method. We have used both Polynomial kernel and Radial Basis Function (RBF) kernel. For polynomial kernel, the parameter d has only few choices, $d \in [2, 3, 4, 5, 6]$. RBF is a popular, general purpose, yet powerful kernel. Generalization performance of SVM with RBF kernel

usually depends closely on the combination of (C, γ) . The values are selected from the range $C = [2^{-5} 2^{-3} 2^{-1} 2^1 2^3 2^5 2^7 2^9 2^{11} 2^{13} 2^{15}]$ and $\gamma = [2^{-15} 2^{-14} 2^{-13} 2^{-12} 2^{-11} 2^{-10} \dots 2^1 2^3]$ using grid search and cross validation. Contour plot describing the parameter selection is depicted in Fig. 5.5. The optimal parameters obtained were $C=8$ and $\gamma = 0.008$. This plot is drawn with gradient features.

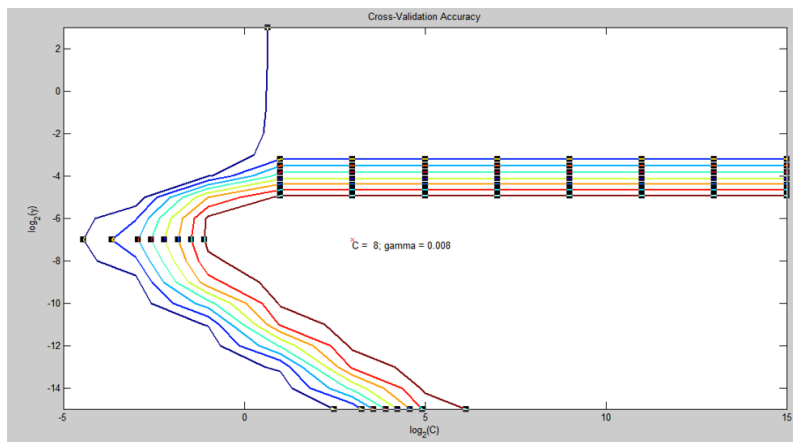


Fig. 5.5 Parameter Tuning for SVM

The number of hidden nodes L and the parameter C for optimization based ELM is chosen from the range $L = [500 1000 1500 2000 2500 3000 3500 4000]$ and $C = [2^{18} 2^{12} 2^6 2^0 2^{-6} 2^{-12} 2^{-18}]$. Fig. 5.6 and Fig. 5.7 demonstrates parameter tuning with gradient features. The optimal values were obtained with $L=4000$ and $C=2^{-6}$ for sigmoid and $C=2^{-12}$ for Gaussian activation function, $G(a, b, x) = \exp(-b\|x - a\|^2)$ where sigmoid

activation function $G(a, b, x) = \frac{1}{(1 + \exp(-a \cdot x + b))}$ is yielding better performance than Gaussian activation function. We have used sigmoid activation function in ELM for all the experiments.

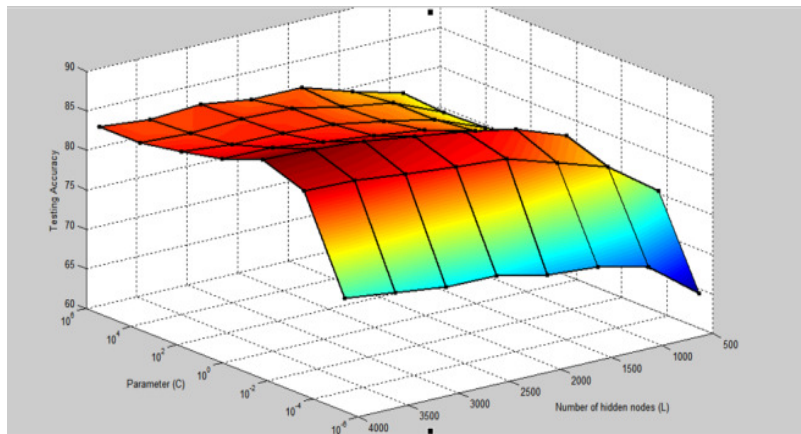


Fig. 5.6 Parameter tuning for ELM with sigmoid activation function

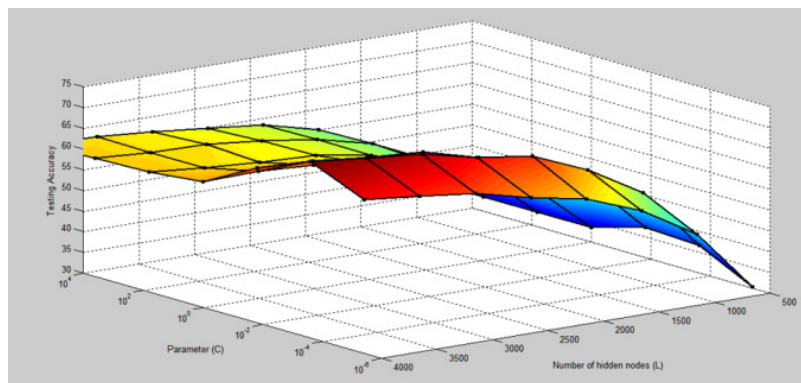


Fig. 5.7 Parameter tuning for ELM with Gaussian activation function

All the experiments are carried out on PC Intel® Core™i3 CPU M 370 @ 2.40GHz processor and 4 GB RAM; MATLAB 7.8.0 (2009a) environment. All feature vectors are normalized in the range [-1 1] before inputting to the classifier.

5.4.2 Performance Evaluation using Topological Features, Distribution Features, Transition Count Features and LBP Features

The topological features calculated from the character images include, number of loops, width/height ratio, number of endpoints, number of cross points and number of branch points. These five global features are not enough to represent 90 different character classes. So we combine distribution features along with these features. In order to find out distribution features, different sized grids namely, 32×32 , 16×16 , 8×8 and 4×4 are placed on pre processed image. This gives 4, 16, 64 and 256 cells and an equal number features are extracted based the occurrence of pixels in each cell. Too few numbers of cells may be incapable of representing the character images efficiently and too large number of cells may induce noise in the feature set. Variations in recognition accuracy based on number of distribution features by varying the number of cells are shown in Fig 5.8. All the five classifiers are giving optimum result with distribution features calculated from 64 cells. The highest

classification result of 86.15% is obtained with SVM classifier with Polynomial kernel.

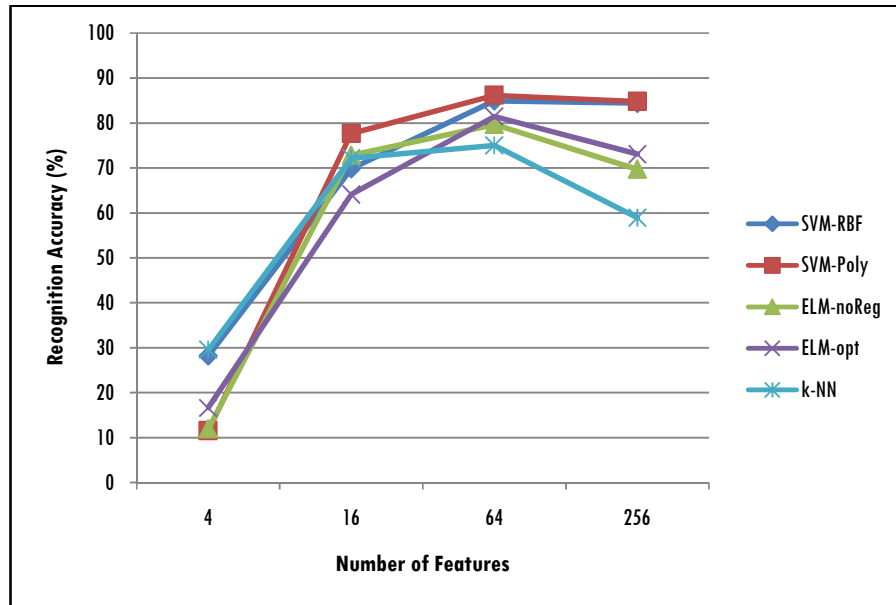


Fig. 5.8 Variations in recognition accuracy based on number of distribution features

Since statistical and structural features are complementary in nature, along with these 64 distribution features, five global features, namely, number of loops, width/height ratio, number of endpoints, number of cross points and number of branch points are added. The width/height ratio is calculated from initial segmented image, while the remaining three are calculated from skeleton of the image. The recognition accuracy of all classifiers has been increased as we combine topological features with

distribution features. The highest accuracy of 87.44% is obtained with SVM classifier with Polynomial kernel (Table 5.3).

Transition Features and LBP features are calculated from the character images as specified in Chapter 4, Section 4.4 and Section 4.5 respectively. The number of features computed is 128. The classification result is depicted in Table 5.3. With transition count features, the highest accuracy obtained is 76.73% and with LBP features, the maximum accuracy obtained is 58.35% both with SVM-RBF classifier.

Table 5.3 Classification Result of Topological Features, Distribution Features, Transition Count Features and LBP Features

Description of Features	Dimension of Features	Recognition Accuracy on 90 character classes (%)				
		SVM-RBF	SVM-Poly	ELM-noReg	ELM-opt	k-NN
Distribution Features	64	84.91	86.15	79.67	81.42	75.02
Topological Features and Distribution Features	69	86.47	87.44	81.42	82.36	76.11
Transition Count Features	128	76.73	74.6	67.64	67.60	67.64
LBP Features	128	58.35	51.75	48.49	51.00	48.78

5.4.2.1 Feature Combination

Individual performances of the above features are not satisfactory. Considering the fact that these features takes care of different aspects of handwriting, we have decided to combine them. To evaluate the performance of the combinational features, we concatenate topological

features, distribution features, transition features and LBP features. As the dimension of feature vector becomes 325, using Principal Component Analysis, we have reduced the number of features to 76, beyond that contribution becomes negligible. Scree plot showing the variance explained by principal components is depicted in Fig. 5.9. Classification result in PCA feature space is provided in Table 5.4. The recognition accuracy of all the classifiers has been increased using the combined features. The highest accuracy of **92.31%** is obtained with SVM-RBF kernel with a testing time of 27.59 seconds.

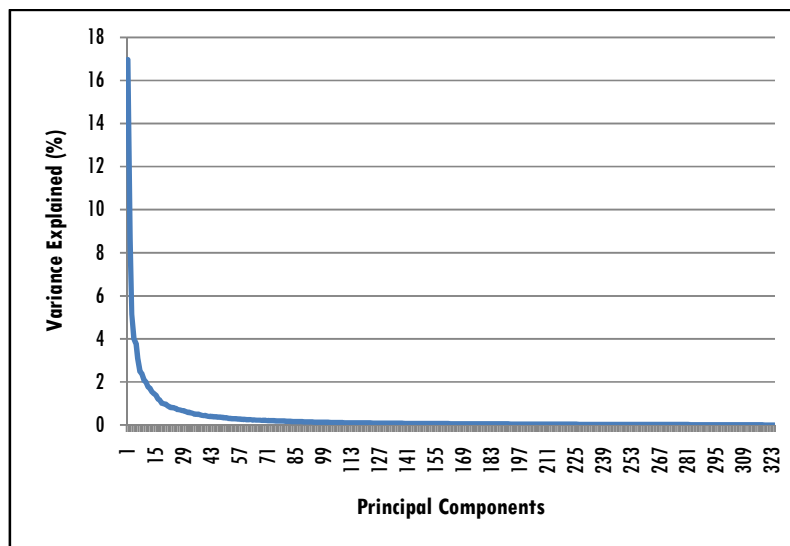


Fig. 5.9 Scree Plot: Variance Explained by Principal Components

Table 5.4 Classification Result of Feature Combination in PCA Feature Space

Feature Combination - Reduced Dimension : 76	SVM-RBF	SVM-Poly	k-NN k=5	ELM-noReg	ELM-opt
Training Accuracy (%)	99.97	99.96	-	98.16	97.88
Testing Accuracy (%)	92.31	91.53	84.78	87.27	89.0
Training Time in seconds	87.53	66.59	-	536	262.97
Testing Time in seconds	27.59	17.77	68.28	3.5	5.72

5.4.3 Performance Evaluation using Wavelet Features

In order to demonstrate the effectiveness of the method proposed in Chapter 4, Section 4.6, five types of experiments have been conducted both in the gray scale image and in binary image. In the first experiment, approximation of Haar wavelet coefficients at decomposition level 3 (LL₃ subband) have chosen and in the second experiment, all the detailed coefficients at decomposition level 3 (LL₃, LH₃, HL₃ and HH₃) have considered. The third experiment uses approximation coefficients at level 2 (LL₂ subband) as features. In the fourth and fifth experiment, horizontal detailed coefficients (LH₂ subband) and vertical detailed coefficients (HL₂ subband), respectively, are chosen as features. Summary of classification results are displayed in Table 5.5. Column 2 of this table gives the decomposition level and column 3 displays the corresponding dimension. In most cases, results on gray scale representation yields better compared to binary representation. The best performance is obtained using approximation coefficients at decomposition level 2 as features. Here

also, the highest accuracy is obtained with SVM-RBF using 256 features computed from gray scale image.

Table 5.5 Classification Result of Wavelet Features

Description of Features	Level	Dimension of Features	Recognition Accuracy on 90 character classes (%)			
			SVM-RBF		ELM-opt	
			Gray Scale	Binary	Gray Scale	Binary
Approximation Coefficients	3	64	85.92	85.91	82.77	82.53
Detailed Coefficients	3	192	73.733	73.63	61.93	61.29
Approximation Coefficients	2	256	89.09	89.00	82.78	82.51
Detailed Coefficients (H)	2	256	60.42	60.36	50.11	49.93
Detailed Coefficients (V)	2	256	60.22	61.38	50.82	52.87

We use 256 wavelet features chosen from the previous experiment and reduce its dimension to 83 using PCA. The performances of all the classifiers are listed in Table 5.6. SVM with RBF kernel with parameters $\gamma = 0.02$, $C = 100$ gives the best performance of 89.47%. The performance of optimization based ELM is also increased to 84.29% in the reduced dimension.

Table 5.6 Classification Result of Wavelet Features in PCA Feature Space

Wavelet Features: Reduced Dimension : 83	SVM-RBF	SVM-Poly	k-NN	ELM-noReg	ELM-opt
Training Accuracy (%)	99.70	99.76	-	98.30	99.17
Testing Accuracy (%)	89.47	89.02	81.57	81.82	84.29
Training Time in seconds	116.74	97.92	-	425.27	117.06
Testing Time in seconds	27.14	21.13	75.91	1.73	2.50

5.4.4 Performance Evaluation using Chain Code Features

The experiments are conducted using chain code features extracted from contour representation as well as skeleton representation of binary character images. Both 8-direction and 4-orientation features as explained in Chapter 4, Section 4.7 are used for experimentation.

Table 5.7 Classification Result of Chain Code Features

Description of Features	No. of zones	Size of a zone	8-direction plane			4-orientation plane		
			Dimension	Acc (%)		Dimension	Acc (%)	
				SVM-RBF	ELM-opt		SVM-RBF	ELM-opt
Contour based chain code features	2×2	32×32	32	45.36	42.56	16	42.93	39.42
	4×4	16×16	128	81.82	79.91	64	80.56	77.89
	8×8	8×8	512	89.78	82.4	256	89.56	82.44
Skeleton based chain code features	2×2	32×32	32	61.31	59.22	16	58.80	53.64
	4×4	16×16	128	87.97	81.38	64	86.96	80.64
	8×8	8×8	512	90.13	82.71	256	90.16	82.80

The classification results of 12 chain code features are depicted in Table 5.7. The images are divided into 2 × 2, 4 × 4 and 8 × 8 zones for

extracting features. Classification accuracy increases as we increase the number of zones. Chain code features based on skeleton is better than chain code features based on contour. 4-orientation chain code has comparable performance with 8-direction chain code. The difference in accuracy is however not as prominent as between contour based chain code and skeleton based chain code. The best performance is obtained with 8×8 zone. SVM-RBF classifies with 90.13% accuracy using 8-direction chain code and 90.16% accuracy using 4-orientation chain code both computed from skeleton representation of character pattern. So 4-orientation chain code from skeleton is chosen for further experiment. The dimension of feature vector is 256 with 8×8 zone. Using PCA, we have reduced the number of features to 160, beyond that contribution becomes negligible. Classification result in PCA feature space is provided in Table 5.8.

Table 5.8 Classification Result of Chain Code Features in PCA Feature Space

Chain Code Features - Reduced Dimension: 160	SVM-RBF	SVM- Poly	k-NN	ELM-noReg	ELM-opt
Training Accuracy (%)	99.44	99.43	-	93.22	97.67
Testing Accuracy (%)	90.11	90.04	80.15	76.98	83.57
Training Time in seconds	149.70	149.01	-	95.87	125.22
Testing Time in seconds	35.08	35.04	145.14	1.29	3.15

5.4.5 Performance Evaluation using Gradient Features

The experiments are conducted using gradient features extracted directly from gray scale representation of character images, to avoid artifacts induced due to binarization. The images are divided into 4×4 zone for

extracting features. Both 8-direction and 4-orientation features as explained in Chapter 4, Section 4.8 are used for experimentation. In this case also, SVM-RBF and SVM with Polynomial kernel are giving the best performances. Merging 8 directions to 4 orientations didn't degrade the performance much. The difference between the highest results among them is only 0.73%. The classification result in the original feature space and reduced feature space is depicted in Table 5.9 and Table 5.10 respectively.

Table 5.9 Classification Result of Gradient Features

Description of Features	Dimension	Recognition Accuracy on 90 classes (%)				
		SVM RBF	SVM – Poly	K-NN	ELM-no Reg	ELM-opt
Gradient 8 direction	128	94.8	94.73	88.69	90.96	92.78
Gradient 4 orientation	64	94.07	93.09	83.82	89.16	91.13

Table 5.10 Classification Result of Gradient Features in PCA Feature Space

Gradient Features - Reduced Dimension: 60	SVM-RBF	SVM- Poly	k-NN	ELM-noReg	ELM-opt
Training Accuracy (%)	99.96	99.53	-	98.90	98.70
Testing Accuracy (%)	94.53	94.17	88.24	92.84	92.96
Training Time in seconds	80.51	59.55	-	494.96	193.46
Testing Time in seconds	21.63	15.43	126.25	3.69	6.04

5.4.6 Performance Evaluation using Curvature Features

The experiments are conducted using curvature features extracted from gray scale representation of character images using 4×4 zones as discussed in Section 4.9 of Chapter 4. The classification result in the

original feature space is depicted in Table 5.11 and the result in reduced feature space is depicted Table 5.12.

Table 5.11 Classification Result of Curvature Features

Description of Features	Dimension of Features	Recognition Accuracy on 90 character classes (%)				
		SVM RBF	SVM – Poly	K-NN	ELM-no Reg	ELM-opt
Curvature	384	93.91	93.89	84.13	84.42	89.29

Table 5.12 Classification Result of Curvature Features in PCA Feature Space

Curvature Features - Reduced Dimension: 183	SVM-RBF	SVM-Poly	k-NN k=15	ELM-noReg	ELM-opt
Training Accuracy (%)	99.65	99.77	-	98.90	98.94
Testing Accuracy (%)	93.58	93.29	84.91	86.78	89.62
Training Time in seconds	367.69	295.98	-	958.42	420.73
Testing Time in seconds	85.61	69.60	341.33	6.82	9.13

5.4.7 Performance Evaluation using SSG Features

The experiments are conducted using SSG features extracted from gray scale representation of character images. As SSG gives importance to horizontal, vertical and both diagonal directions, its discriminating power gets enhanced. The original dimension of the feature is 256. It is reduced to 78 using PCA. The recognition accuracy of **95.40%** (Table 5.13) obtained with SVM-RBF is the best result obtained so far.

Table 5.13 Classification Result of SSG Features in PCA Feature Space

SSG Features - Reduced Dimension: 78	SVM-RBF	SVM- Poly	k-NN k=7	ELM-noReg	ELM-opt
Training Accuracy (%)	99.77	99.76	-	99.02	99.28
Testing Accuracy (%)	95.40	94.93	89.09	92.0	93.07
Training Time in seconds	89.46	72.6	-	302.4	243.83
Testing Time in seconds	23.42	18.45	153.8	3.65	7.21

5.4.8 Analysis of Results

Nine different types of features with different sets of feature vectors were classified using classifiers such as k-NN, SVM and ELM. The type of features include topological features, distribution features, transition count features, LBP features, wavelet features, chain code features, gradient features, curvature features and SSG features. Simple features such as topological, LBP, distribution features were not sufficient to capture all the information. But combined features performed very well in recognition stage. The recognition accuracy of 92.31% was obtained with SVM-RBF classifier with a testing time of 27.59 seconds.

Among the five different sets of features from wavelet domain, approximation coefficients yield the best result. In all the cases, gray scale image based features provided better results than binary image based features. The highest accuracy obtained was 89.47% using SVM-RBF kernel in a testing time of 27.14 seconds.

Both chain code and gradient features provide directional information. In the case of chain code features, skeleton based chain code

yield better result than contour based features. The highest accuracy obtained was 90.11%. Gradient features and curvature features were extracted from gray scale images. The highest accuracy obtained were 94.53% and 93.58% respectively using SVM-RBF.

The best testing accuracy of 95.40% among all the features described above were obtained using SSG features and SVM-RBF classifiers with just 78 features. Since the context in which the character written is not known at this stage, the result obtained is promising.

In majority of the experiments, SVM classifier with RBF kernel outperforms all other classifiers in terms of recognition accuracy. But the testing time of ELM is extremely less compared to SVM and k-NN. Accuracy of SVM was not very sensitive to the dimension feature space. It was also noted that in ELM and k-NN, when the dimension of features are being reduced, the complexity of classifier decreases and this results in better performance.

When we analysed the confusion matrix, it is clear that most of the errors are due to confusion among similar character patterns. Several such pairs exist in Malayalam language. Some of them are displayed in Fig. 5.10.

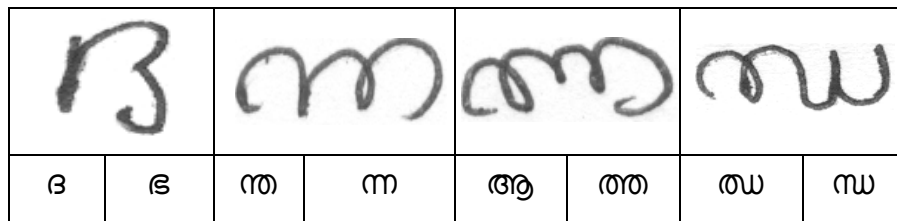


Fig. 5.10 Some similar characters in Malayalam

5.4.9 Performance Evaluation on the Dataset Created Through Mobile Phone Camera

We have conducted experiments with the database created through mobile phone camera as explained in Chapter 3, Section 3.7. The best features selected in each category are selected for evaluation. As SVM with RBF kernel is giving good performance, it selected for classification. Since the number of samples per class is not uniform, stratified 10 fold cross-validation is used for the experiment. Table 5.14 summarizes the classification result using this database.

Table 5.14 Classification Result on the Dataset Created through Mobile Phone Camera

Description of Features	Dimension of Features	Accuracy (%)	Precision	Recall	F-measure
Distribution Features	64	76.92	0.772	0.769	0.767
Topological Features and Distribution Features	69	78.81	0.789	0.788	0.786
Transition Count Features	128	66.26	0.654	0.663	0.645
LBP Features	128	60.05	0.61	0.601	0.593
Wavelet Features	256	89.20	0.895	0.892	0.892
Chain Code Features – 4 Orientation (Contour Based)	256	90.01	0.9	0.9	0.897
Chain Code features – 4 Orientation (Skeleton Based)	256	84.34	0.848	0.843	0.843
Gradient Features	64	91.22	0.915	0.912	0.913
Curvature Features	192	89.20	0.895	0.892	0.892
SSG Features	128	92.4	0.924	0.926	0.925

The recognition accuracy obtained is low for most of the features even though the number of classes is only 25. This is mainly because the images captured are of low resolution. With distribution features, the recognition accuracy obtained is 76.92%. With transition count feature, the accuracy obtained is only 66.26%. Since the images obtained are of low resolution, thinning induces several artifacts. Wavelet features and curvature features are providing good classification results where as directional features such as chain code and gradient are giving the best two classification results. Performance of SSG feature is better than gradient feature.

5.5 Conclusion

In this chapter, we discussed classifiers such as k-NN classifier, SVM classifier with two kernels and ELM with two variants. Nine different types of features with different sets of feature vectors are classified using these classifiers. The best zone size for distribution features is identified experimentally. The individual performances of topological features, distribution features, transition count features, LBP features are discussed. The classification result of the combined feature in the reduced dimension is provided along with training and testing time.

To demonstrate the effectiveness of wavelet features, five different feature vectors are developed both from gray scale image and from binary image. Feature vectors using approximation coefficients at decomposition

level 2 of gray scale image is found to achieve best result among other wavelet features. Its performance in the PCA feature space is also given.

Twelve sets of chain code features are classified using the classifiers mentioned above. Features were extracted from contour as well as skeleton representation of binary character patterns. The best zone size for feature extraction is found out experimentally. The dimensionality of the best feature vector in this domain is reduced and performance evaluation is done.

The directional feature such as gradient captures the information content in the character pattern. Experiments are conducted using gradient features extracted from gray scale image as this representation provides more result than binary images. The curvature features are also extracted from gray scale image. The performance of gradient and curvature features in the original and reduced feature space is carried out.

Based on gradient feature, a novel feature descriptor named SSG is developed, which has more discriminating power. Empirical results convey that the feature descriptor, SSG is efficient in selecting salient features from the images as this feature descriptor provides the best classification result of **95.40%** among all other feature descriptors. The benefit of recognition on gray scale images is justified as our methods yield high recognition accuracies on this representation of character images. The results obtained are encouraging considering the complex shapes of character images and the large number of classes in the database.

Experiments are conducted using all the above features on camera captured dataset. The recognition results are lower in this case, mainly because of low resolution of character images.

Main difficulty of any recognition system is shape similarity. In Malayalam, many characters have shape similarity. This challenge increases even further in handwritten symbols. Further analysis of the results reveals the fact that most of the errors are due to confusion among similar characters. A single classifier with designed-for-all global features is found to be incapable in estimating class boundaries in the feature space for large number of classes.

Chapter 6

ACCURACY IMPROVEMENT THROUGH TWO-STAGE APPROACH AND CLASS SPECIFIC FEATURES

Contents

6.1 Introduction
6.2 Design of a Two-Stage Recognizer
6.3 Results and Discussions
6.4 Conclusion

6.1 Introduction

The task of improving accuracy becomes more challenging as there are large numbers of classes and high similarity between characters. Since our work is based on a large database of real-life handwritten samples, a single classifier will not be sufficient to discriminate highly similar character images. The challenging part of the work is in the distinction of similar shaped components as a small variation in writing creates two or more different classes. To deal with these issues, we propose a two-stage classification approach in which, the first stage classifier identifies potential conflicts in classification and appropriate grouping is done to enhance the classification performance.

In the first stage of the proposed two stage recognition scheme, we classify input characters into smaller groups using a group classifier

which is able to detect the potential conflicts in classification. In the second stage, discriminative and specific features are used to resolve the conflicts among similar characters in each individual group using a bank of classifiers, each corresponding to one group of character classes.

Two-stage classification approach is a novel attempt in Malayalam HCR. Even though, this approach was used in Devnagari [83], Bangla [85] and Tamil [111], the method designed by us is different. We have introduced a new technique to discriminate each character class from a group. In the second stage, we have developed special features suitable to discriminate characters that fall under each group depending on the class members of the group. This chapter is organized as follows: The next section covers the design of a two-stage recognizer with its architecture, methodology for the creation of groups and the design of second stage classifier. The specific features for each group are specified in this section. Section 6.3 provides the results and discussion. Section 6.4 concludes the chapter.

6.2 Design of a Two-Stage Recognizer

Single stage classification for a problem with 90 unique character shapes is not much effective to separate all classes in the feature space. The performance of such classifier degrades primarily due to the presence of many similar shaped character classes. A feature extractor that is designed globally for all classes is incapable for the differentiation of

similar character patterns. So we have designed a two-stage approach to achieve this goal. A two-stage classification approach has several advantages: First of all, it makes the number of classes in each stage small and significantly reduces the error of misclassification of similar shaped characters in the first stage. The misclassification rate could be reduced based on the idea that, if class C_i is misclassified into class C_j with $x_{ij}\%$ of error, and C_j be misclassified into class C_i with $x_{ji}\%$ of error, then by treating C_i and C_j in a single group, the error $x_{ij}+x_{ji}$ will disappear. The pairs of classes such as (C_i, C_j) , (C_j, C_k) can be further grouped as (C_i, C_j, C_k) . This process can be repeated till optimal group classification accuracy is obtained. If a group contains only one class, no further processing needs to be done in the second stage. If a group contains more than one class, each individual class can be identified using a separate classifier for each group. This classifier need not be same for each group. It is easy to handle each group because it contains only a few numbers of classes. We can use different features for separating classes from groups.

6.2.1 Architecture of the Two-Stage Recognizer

The classifier consists of two stages. The first stage is a group classifier, which puts the test data into one of groups which contain similar character classes. If it is classified into one of the groups, then it is fed to a second stage classifier and this new classifier decides the label of this unknown pattern. If the decision of the first stage classifier is wrong,

it cannot be corrected in the second stage. The architecture of a two-stage recognizer is given in Fig. 6.1

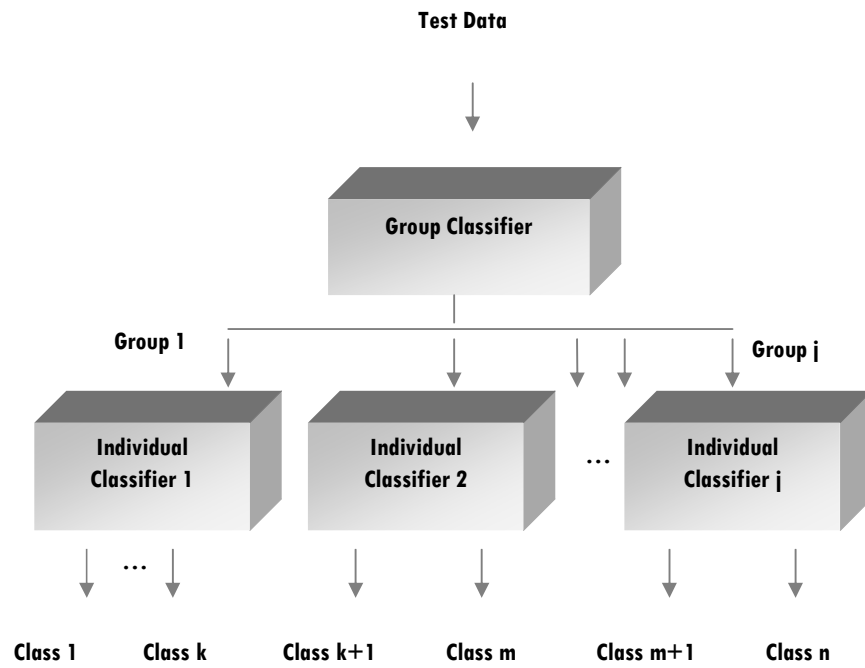


Fig. 6.1 Architecture of two stage recognizer

6.2.2 Creation of Groups

In the first stage, all the 90 character patterns are classified using an SVM classifier with RBF kernel using the feature extractor described in Chapter 4, Section 4.8. The confusion matrices are created from the training set using 5-fold cross validation and it is used to identify the possible misclassification among all pattern classes. The steps performed for creating groups are listed below:

- Read the confusion matrix $CF(i, j); i, j = 1, 2, \dots, C$ obtained for C classes, where $C = 90$
- Find out the similarity between classes i and j , defined as $n(i, j) = CF(i, j) + CF(j, i); i < j$
- Merge the two classes i and j with highest $n(i, j)$, into a group G_p ; Repeat this step until $n(i, j) > t$, a threshold determined experimentally.
- Merge two groups G_p with m classes i_1, i_2, \dots, i_m and G_q with n classes j_1, j_2, \dots, j_n into a single group if they are similar. The similarity between two groups G_p and G_q is defined as $SIM(p, q) = \min_{i < j} n(i, j)$

Using the above algorithm, we have varied the threshold t to merge classes which resulted in different sets of groups. The best sets of groups are identified experimentally. The optimal classification accuracy is

obtained with 16 groups (Fig. 6.2). Certain character patterns do not belong to any group is treated as group with single class.

Groups					
Group-Id	Characters in a Group				Class-Id of Group Members
1	ന	ഞ	ന്ന	ണ	60,18,61,23
2	മ്പ	ബ	സ		63,31,40
3	ഇ	ഉ	ഉ		3,4,42
4	പ	വ	ഖ	ച	29,37,10,14
5	ഡ	ഡ			21,22
6	ഠ	ഠ			85,20
7	ഭ	ഭ	ഭ		32,26,74
8	ൂ	ൂ			79,80
9	ക	ക	ക		9,51,50
10	ആ	ത			2,59
11	ഒ	ഒ			13,72
12	ത	സ			17,73
13	ി	ി			77,78
14	ൻ	ൽ	ൻ	ൾ	46,47,48,49
15	എ	എ			6,7
16	്യ	്യ			81,89

Fig. 6.2 Groups

6.2.3 Design of Second Stage Classifier

The task of the second stage individual classifier is to distinguish and classify individual members from the multi-class group. Initially each individual classifier is trained with the same features used in the first stage to classify each group separately. Since the feature space of the second stage classifier contains at most four classes, this feature may be enough to discriminate classes in the feature space. But when the character patterns in two different classes are too similar, it could be differentiated only by concentrating on the regions where the difference is clear. The advantage of second stage classifier is that, it clearly knows what all character classes are belonging to each group. So, attention can be provided to the regions in which patterns differ the most. Considering these facts, we have designed specific features to handle each class separately to enhance the classification performance of the second stage classifier.

- **Group-1 case:** The second and fourth class characters of group 1, the left starting has a full loop, making a hole. Though people may not always write a full loop, they make a right move from the starting position and then a clockwise rotation by almost 270° or more. On the other hand for the first and third class, there is no loop at the left end and the pen goes up from the starting position, without making rotation. So, **‘the existence of loop or near-loop at the beginning’** is used as a feature. In addition, it is noted that first and the second class of group 1 has another loop at the **third**

leg, but no such loop for third and fourth class. So, **'the existence of loop at the third leg'** is considered as another feature. These two features are considered as additional features in this case.

- **Group-2 case:** Similar to the case of Group-1, here also, **'the existence of loop or near-loop at the beginning'** is used as an additional feature to separate second class from the other classes.
- **Group-3 case:** For characters like ഇ, ഉ, ഊ, people used to write elongated horizontal line at the bottom part of the symbol (Fig. 6.3), which causes error in classification. Such horizontal lines are removed by using a rectangular window at the top of the character and the bounding box of the image is recalculated to compute the SSG feature vector.

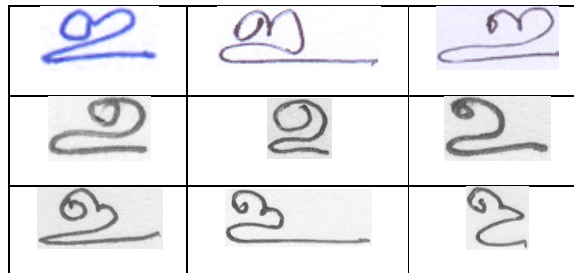


Fig. 6.3 Samples of the characters ഇ, ഉ, ഊ

- **Group-4 case:** Presence of a **loop or near-loop** in the beginning is a distinguishing factor to separate third class from the rest of the classes. So here also this feature is used as an additional feature.

- **Group-5 case:** The characters in the group (Ω , Ω9), are different just due to **presence of a loop in the upper right corner**. So the presence of a hole is detected and added to the existing features.
- **Group-6 case:** For classes like (O, O), both look like circles, but one is big and the other is small. So, **area inside the hole** is used as the only feature to distinguish these two classes.
- **Group-7 case:** In the case of group containing handwritten patterns of $\text{B}, \text{B}, \text{B}$, the right most part of the patterns are different from each other. An end point can be considered as a distinguishing feature here. The number of end points in the first character pattern is 3, second is 2 and third is 4. The number of end points in the pattern is used as an additional feature here.
- **Group-8 case:** The patterns in these two classes are distinct just due to an extra loop at the bottom. In this case, the input pattern is divided into 64 zones using a grid of size 8×8 and sum of foreground pixels in each zone computed. These features provide an idea about the local distribution of pixels in the character patterns. Only these features are used to distinguish these two classes.
- **Group-9 case:** Similar to group-8 case, here also, the distributions of pixels in each zone of the character patterns are found to be

different. Therefore, along with SSG features, distribution of pixels in 64 zones is used here.

- **Group-10 case:** The first character in this group has a marking in the second leg which is not there in the second class. But this marking may not be clear while writing. But the elongated line towards the end of the character is visible in the first class while it is absent in the second class (Fig. 6.4). To differentiate these two classes, we have used vertical transition count feature, which is defined as the number of transitions from a foreground pixel to background pixel along vertical lines through a character pattern. These features capture shape information and are less sensitive to the variations of the handwritten pattern.

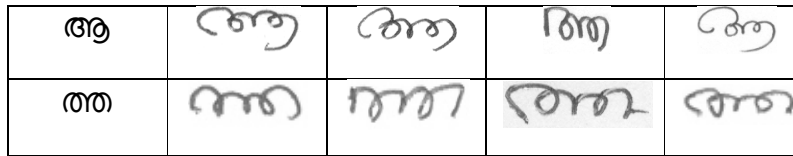


Fig. 6.4 Samples of ആ, അ

- **Group-11 case:** Here the differentiating factor is a loop in the lower left corner. As in the case of **Group-1** here also people may not write a full loop. So the presence of a loop or near-loop in the beginning is detected and used as a new additional feature.
- **Group-12 case:** In this group, the first character pattern has a loop in the second leg and such a loop is not observed in the

second pattern. Number of loops in the first pattern is one and in the second pattern is zero. Therefore, numbers of loops in the pattern are used as distinguishing feature here along with SSG features.

- **Group-13 case:** Similar to group-8 case, here also, the distributions of foreground pixels in the top segment of the patterns are different. The same features used in group-8 case are used here also.
- **Group-14 case:** In the case of pure consonants (ൻ, റ, ൽ, ശ), People may write a loop in the upper part of pure consonants. So there are two different ways of writing each pure consonants (Fig. 6.5). Since this group contains only four character classes, we have ignored the upper part of the character by considering only the lower part of the handwritten pattern using a rectangular window for feature extraction.

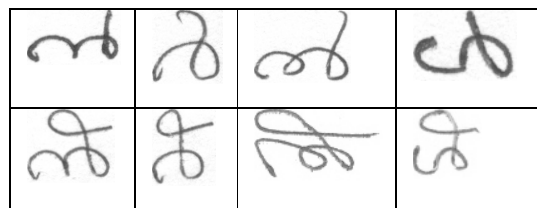




Fig. 6.5 Samples of ഞ, റ, ൽ, ശ

- **Group-15 case:** The shape of the second class has a small mark in the right segment of the pattern. To identify this, we have used

vertical transition count feature, which is defined as the number transitions from a foreground pixel to background pixel along vertical lines through a character pattern. The transition count features are used along with SSG features to distinguish these two classes.

- **Group-16 case:** The symbols like  and  are very similar. The only difference is at the bottom most position. For one class there is a complete hole at the bottom. In the other class the hole is not complete. Some times people may write a complete hole for the first class. So simple dilation with 3×3 structuring element is applied to make it complete and presence of a complete hole at the bottom is used as the only feature.

6.3 Results and Discussions

In this section, we present the results obtained by the method described. We compare the performance of single stage classification with the proposed two-stage approach. Among all classifiers described in Chapter 5, SVM with RBF kernel is providing the highest classification accuracy. We have used SSG feature descriptor extracted from gray scale representation of the image as it yields the best result. The classification result obtained in single stage classification was **95.40%** as depicted in Table 5.13 of Chapter 5.

In the two stage classification, we need a group classifier. SVM-RBF is used to find out the groups using SSG features. The highest group classification accuracy obtained is with 16 groups as specified in Section 6.2.2. The group classification accuracy obtained is **97.62%** (4393/4500) with $C=8$; $\gamma=0.008$ (Table 6.1).

Table 6.1 Group Classification Accuracy with SVM

Group Classification Accuracy with SVM	
Training Accuracy (%)	99.87
Testing Accuracy (%)	97.62

Individual classification accuracy of all the groups is found out using a second stage classifier. SVM with RBF kernel is used for classification in the second stage. We have computed the classification accuracy of the second stage classifier using the same features used in first stage. This result is further enhanced using special features described in Section 6.2.3. The detailed classification result is provided in Table 6.2. Individual classification accuracy during the second stage classification using the same features as in first stage is **93.29%** (1959/2100) and the same using special features are **97.71%** (2052/2100). The overall classification accuracy obtained using the same feature is **96.24%** (6352/6600) and using special features is **97.65%** (6445/6600). The improvements obtained compared to single stage classification are **0.85%** and **2.25%** respectively.

Table 6.2 Second Stage Classification Results

Group	Classes	Number of classes	No. of training samples	No. of test samples	Second Stage Classification Accuracy (%)			
					using the same features		using additional/new features	
					Train Acc (%)	Test Acc (%)	Train Acc (%)	Test Acc (%)
1	60,18,61,23	4	600	200	100	86.5	100	98.5
2	63,31,40	3	450	150	100	91.3	100	95.33
3	3,4,42	3	450	150	100	92.67	100	97.33
4	29,37,10,14	4	600	200	100	97	100	98
5	21,22	2	300	100	100	96	100	98
6	85,20	2	300	100	100	91	99	94
7	32,26,74	3	450	150	99.78	92	99.78	95.33
8	79,80	2	300	100	100	96	100	100
9	9,51,50	3	450	150	100	96	100	98.67
10	2,59	2	300	100	100	93	100	100
11	13,72	2	300	100	100	92	100	98
12	17,73	2	300	100	100	91	100	98
13	77,78	2	300	100	100	94	100	100
14	46,47,48,49	4	600	200	100	97.5	100	99
15	6,7	2	300	100	100	98	100	99
16	81,89	2	300	100	100	88	98.67	94
Weighted Average		42	6300	2100	99.98	93.29	99.87	97.71

6.4 Conclusion

We have presented an efficient two-stage classification approach for the recognition of unconstrained handwritten Malayalam characters. Two-stage approach is not attempted previously in Malayalam. For the recognition task, we have used a large database of real-life handwritten samples belonging to 90 different character classes. In the proposed two-stage approach, the best classifier selected from the previous chapter is chosen as the first stage classifier. The second stage classifier recognizes characters using specific features designed for that particular group. The overall result obtained was **97.65%**. The recognition result reveal that the proposed approach is quite efficient.

Chapter 7

CONCLUSION AND FUTURE SCOPE

Contents

7.1 Conclusion and Major Contributions

7.2 Future Scope

This chapter summarizes the thesis and mentions the possible extensions for future work. The chapter is divided into two sections. Section 7.1 concludes the theses by pointing out the major contributions. Section 7.2 provides future directions of research.

7.1 Conclusion and Major Contributions

In this thesis novel methodologies that assist offline handwritten character recognition of Malayalam script was presented. We developed an efficient system for the recognition of unconstrained handwritten characters in Malayalam script consisting of vowels, consonants, pure consonants, vowel signs, consonant signs and compound characters.

Problem definition, motivation, challenges and objectives were provided. A detailed literature review in the specific field of offline handwritten character recognition problem, describing the feature extraction and classification methods in a wide variety of scripts was conducted. The peculiarities and challenges of written Malayalam were

well studied. All the set of symbols in the modern Malayalam script along with frequently used symbols in the old script were included for collection of character samples. A benchmark database of totally unconstrained Malayalam handwritten samples were created for research purpose. This work is the first exclusive work using 90 different character classes in the Malayalam script. A novel attempt of creating database using mobile phone camera was also provided.

Efficient feature descriptors based on topological, distribution, transition count, LBP, wavelet, gradient and curvature features were developed. Moreover, a novel feature extraction method based on image gradient was introduced. Principal Component Analysis technique for the reducing feature dimension was proposed.

Performance evaluation was provided with classifiers such as k-NN, SVM with two kernels and ELM with two variants. Classifier design was based on extensive experimentation for fine-tuning several parameters that influence the performance. The created benchmark database of 90 character classes and the dataset created through mobile phone camera were used to evaluate the performance of these methodologies.

Even though the feature extraction method itself is quite efficient, there is more to be improved in classification accuracy. This is achieved by an efficient two-stage classification approach. Usage of discriminant

special features to separate similar classes in the second stage to enhance the performance of the classifier is a novel attempt. High recognition accuracy was obtained using this method. All the objectives of the work are satisfied.

The major contributions are summarized below:

- This work is the first exclusive work using 90 different character classes in the Malayalam script.
- A benchmark database of 18,000 unconstrained handwritten character samples is created in Malayalam.
- Document acquisition, enhancement and database creation and recognition using mobile phone camera was performed. An accuracy of 92.4% was obtained with 25 character classes.
- Introduced a novel feature extraction method based on image gradient.
- A novel attempt is made to enhance the performance using discriminant special features in the second stage.
- The highest recognition accuracy of 97.65% was made in the Malayalam handwritten character recognition using 90 character classes.

7.2 Future Scope

In this thesis, we developed an efficient two stage classification approach with discriminant features for the recognition of unconstrained handwritten Malayalam characters. This work opens up many interesting problems in this domain.

- The developed methodology can be adopted for the recognition of other Indian language scripts that share similar features.
- Recognition of degraded handwritten characters is another issue not addressed so far.
- More studies are required in the domain of camera captured document recognition.
- The work can be extended to recognize unconstrained handwritten words or sentences

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

REFERENCES

- [1] Plamondon, R. and S.N. Srihari, *Online and off-line handwriting recognition: a comprehensive survey*. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2000. 22(1): p. 63-84.
- [2] Impedovo, S., *More than twenty years of advancements on Frontiers in handwriting recognition*. Pattern Recognition, 2014. 47(3): p. 916-928.
- [3] Cheriet, M., et al., *Handwriting recognition research: Twenty years of achievement... and beyond*. Pattern Recognition, 2009. 42(12): p. 3131-3135.
- [4] Tzanakou, E.M., *Supervised and unsupervised pattern recognition: feature extraction and computational intelligence*. 2002: CRC Press.
- [5] Suen, C.Y., M. Berthod, and S. Mori, *Automatic recognition of handprinted characters—the state of the art*. Proceedings of the IEEE, 1980. 68(4): p. 469-487.
- [6] Mori, S., C.Y. Suen, and K. Yamamoto, *Historical review of OCR research and development*. Proceedings of the IEEE, 1992. 80(7): p. 1029-1058.

References

- [7] Mori, S., H. Nishida, and H. Yamada, *Optical character recognition*. 1999: John Wiley & Sons, Inc.
- [8] Cheriet, M., et al., *Character Recognition Systems: A Guide for Students and Practitioners*. 2007: Wiley-Interscience.
- [9] Mantas, J., *An overview of character recognition methodologies*. *Pattern Recognition*, 1986. 19(6): p. 425-430.
- [10] Srihari, S.N., *Recognition of handwritten and machine-printed text for postal address interpretation*. *Pattern Recognition Letters*, 1993. 14(4): p. 291-302.
- [11] Camastra, F., *A SVM-based cursive character recognizer*. *Pattern Recognition*, 2007. 40(12): p. 3721-3727.
- [12] Dong, J.-x., A. Krzyżak, and C.Y. Suen, *An improved handwritten Chinese character recognition system using support vector machine*. *Pattern Recognition Letters*, 2005. 26(12): p. 1849-1856.
- [13] Liu, H. and X. Ding. *Handwritten character recognition using gradient feature and quadratic classifier with multiple discrimination schemes*. in *Document Analysis and Recognition, Proceedings. Eighth International Conference on*. 2005. IEEE.
- [14] Yamada, H., K. Yamamoto, and T. Saito, *A nonlinear normalization method for handprinted Kanji character*

- recognition—line density equalization*. Pattern Recognition, 1990. 23(9): p. 1023-1029.
- [15] Amin, A., H. Al-Sadoun, and S. Fischer, *Hand-printed Arabic character recognition system using an artificial network*. Pattern Recognition, 1996. 29(4): p. 663-675.
- [16] Amin, A., *Off-line Arabic character recognition: the state of the art*. Pattern Recognition, 1998. 31(5): p. 517-530.
- [17] Lorigo, L.M. and V. Govindaraju, *Offline Arabic handwriting recognition: a survey*. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2006. 28(5): p. 712-724.
- [18] Pal, U. and B.B. Chaudhuri, *Indian script character recognition: a survey*. Pattern Recognition, 2004. 37(9): p. 1887-1899.
- [19] Sethi, I.K. and B. Chatterjee, *Machine recognition of constrained hand printed Devanagari*. Pattern Recognition, 1977. 9(2): p. 69-75.
- [20] Chinnuswamy, P. and S. Krishnamoorthy, *Recognition of handprinted Tamil characters*. Pattern Recognition, 1980. 12(3): p. 141-152.
- [21] Pal, U., R. Jayadevan, and N. Sharma, *Handwriting Recognition in Indian Regional Scripts: A Survey of Offline Techniques*. ACM Transactions on Asian Language Information Processing (TALIP), 2012. 11(1): p. 1-35.

References

- [22] Lajish, V.L. *Handwritten Character Recognition using Perceptual Fuzzy-Zoning and Class Modular Neural Networks*. in *Innovations in Information Technology, IIT '07. 4th International Conference on*. 2007.
- [23] Due Trier, Ø., A.K. Jain, and T. Taxt, *Feature extraction methods for character recognition—a survey*. *Pattern Recognition*, 1996. 29(4): p. 641-662.
- [24] Huang, G.B., et al., *Extreme learning machine for regression and multiclass classification*. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 2012. 42(2): p. 513-529.
- [25] Arica, N. and F.T. Yarman-Vural, *An overview of character recognition focused on off-line handwriting*. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 2001. 31(2): p. 216-233.
- [26] Govindan, V. and A. Shivaprasad, *Character recognition—a review*. *Pattern Recognition*, 1990. 23(7): p. 671-683.
- [27] Gader, P., et al., *Recognition of handwritten digits using template and model matching*. *Pattern Recognition*, 1991. 24(5): p. 421-431.
- [28] Jain, A.K. and D. Zongker, *Representation and recognition of handwritten digits using deformable templates*. *Pattern*

- Analysis and Machine Intelligence, IEEE Transactions on, 1997. 19(12): p. 1386-1390.
- [29] Shridhar, M. and A. Badreldin, *A high-accuracy syntactic recognition algorithm for handwritten numerals*. Systems, Man and Cybernetics, IEEE Transactions on, 1985(1): p. 152-158.
- [30] Knerr, S., L. Personnaz, and G. Dreyfus, *Handwritten digit recognition by neural networks with single-layer training*. Neural Networks, IEEE Transactions on, 1992. 3(6): p. 962-968.
- [31] Lee, Y., *Handwritten digit recognition using k nearest-neighbor, radial-basis function, and backpropagation neural networks*. Neural computation, 1991. 3(3): p. 440-449.
- [32] Bunke, H., M. Roth, and E.G. Schukat-Talamazzini, *Off-line cursive handwriting recognition using hidden Markov models*. Pattern Recognition, 1995. 28(9): p. 1399-1413.
- [33] Chen, M.-Y., A. Kundu, and J. Zhou, *Off-line handwritten word recognition using a hidden Markov model type stochastic network*. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 1994. 16(5): p. 481-496.
- [34] Abuhaiba, I. and P. Ahmed, *A fuzzy graph theoretic approach to recognize the totally unconstrained handwritten numerals*. Pattern Recognition, 1993. 26(9): p. 1335-1350.

References

- [35] Chi, Z., J. Wu, and H. Yan, *Handwritten numeral recognition using self-organizing maps and fuzzy rules*. Pattern Recognition, 1995. 28(1): p. 59-66.
- [36] Ager, S., *Writing systems and languages of the world*. Omniglot (1998-2007) <http://www.omniglot.com/writing/tifinagh.html> 2009.
- [37] Ghosh, D., T. Dube, and A.P. Shivaprasad, *Script Recognition; A Review*. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2010. 32(12): p. 2142-2161.
- [38] LeCun, Y. and C. Cortes, *The MNIST database of handwritten digits*, 1998.
- [39] Zhang, H., et al. *HCL2000-A large-scale handwritten Chinese character database for handwritten character recognition*. in *Document Analysis and Recognition, ICDAR'09. 10th International Conference on*. 2009. IEEE.
- [40] Liu, C.-L., et al. *CASIA online and offline Chinese handwriting databases*. in *Document Analysis and Recognition (ICDAR), International Conference on*. 2011. IEEE.
- [41] Bortolozzi, F., et al. *Recent advances in handwriting recognition*. in *the Proceedings of International Workshop on Document Analysis (IWDA), India*. 2005. Citeseer.

- [42] Wunsch, P. and A.F. Laine, *Wavelet descriptors for multiresolution recognition of hand printed characters*. Pattern Recognition, 1995. 28(8): p. 1237-1249.
- [43] Lee, S.-W., et al., *Multiresolution recognition of unconstrained handwritten numerals with wavelet transform and multilayer cluster neural network*. Pattern Recognition, 1996. 29(12): p. 1953-1961.
- [44] Chen, G.Y., T.D. Bui, and A. Krzyzak, *Contour-based handwritten numeral recognition using multiwavelets and neural networks*. Pattern Recognition, 2003. 36(7): p. 1597-1604.
- [45] Bellili A, G.M., Gallinari P, *An MLP-SVM combination architecture for offline handwritten digit recognition: reduction of recognition errors by support vector machine rejection mechanisms*. International Journal of Document Analysis and Recognition, 2003. 5: p. 244-252.
- [46] Liu, C.-L., et al., *Handwritten digit recognition: benchmarking of state-of-the-art techniques*. Pattern Recognition, 2003. 36(10): p. 2271-2285.
- [47] Zhang, P., T.D. Bui, and C.Y. Suen, *A novel cascade ensemble classifier system with a high recognition performance on handwritten digits*. Pattern Recognition, 2007. 40(12): p. 3415-3429.

References

- [48] Kavallieratou, E., N. Fakotakis, and G. Kokkinakis. *Handwritten character recognition based on structural characteristics*. in *Pattern Recognition, Proceedings. 16th International Conference on*. 2002. IEEE.
- [49] Vamvakas, G., B. Gatos, and S.J. Perantonis, *Handwritten character recognition through two-stage foreground sub-sampling*. *Pattern Recognition*, 2010. 43(8): p. 2807-2816.
- [50] Casey, R. and G. Nagy, *Recognition of printed Chinese characters*. *Electronic Computers, IEEE Transactions on*, 1966(1): p. 91-101.
- [51] Agui, T. and H. Nagahashi, *A description method of handprinted Chinese characters*. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 1979(1): p. 20-24.
- [52] Liu, C.-L., et al., *Handwritten digit recognition: investigation of normalization and feature extraction techniques*. *Pattern Recognition*, 2004. 37(2): p. 265-279.
- [53] Ding, K., et al. *A comparative study of Gabor feature and gradient feature for handwritten Chinese character recognition*. in *Wavelet Analysis and Pattern Recognition, ICWAPR'07. International Conference on*. 2007. IEEE.
- [54] Gao, T.-F. and C.-L. Liu, *High accuracy handwritten Chinese character recognition using LDA-based compound distances*. *Pattern Recognition*, 2008. 41(11): p. 3442-3451.

- [55] Zhang, Z., et al. *Character-SIFT: a novel feature for offline handwritten Chinese character recognition*. in *Document Analysis and Recognition, ICDAR'09. 10th International Conference on*. 2009. IEEE.
- [56] Leung, K. and C. Leung, *Recognition of handwritten Chinese characters by critical region analysis*. *Pattern Recognition*, 2010. 43(3): p. 949-961.
- [57] Ni, E., M. Jiang, and C. Zhou, *Radical Extraction for Handwritten Chinese Character Recognition by Using Radical Cascade Classifier*, in *Electrical, Information Engineering and Mechatronics*, 2012, Springer. p. 419-426.
- [58] He, Z., Y. Zhong, and Y. Cao, *High Accuracy Handwritten Chinese Character Recognition Based on Support Vector Machine and Independent Component Analysis*, in *Informatics and Management Science V*. 2013, Springer. p. 725-733.
- [59] Kimura, F., et al., *Improvement of handwritten Japanese character recognition using weighted direction code histogram*. *Pattern Recognition*, 1997. 30(8): p. 1329-1337.
- [60] Kato, N., et al., *A handwritten character recognition system using directional element feature and asymmetric Mahalanobis distance*. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 1999. 21(3): p. 258-262.

References

- [61] Mowlaei, A., K. Faez, and A.T. Haghghat. *Feature extraction with wavelet transform for recognition of isolated handwritten Farsi/Arabic characters and numerals*. in *Digital Signal Processing, DSP, 14th International Conference on*. 2002. IEEE.
- [62] Mozaffari, S., K. Faez, and H.R. Kanan. *Feature comparison between fractal codes and wavelet transform in handwritten alphanumeric recognition using SVM classifier*. in *Pattern Recognition, ICPR, Proceedings of the 17th International Conference on*. 2004. IEEE.
- [63] Liu, C.-L. and C.Y. Suen, *A new benchmark on the recognition of handwritten Bangla and Farsi numeral characters*. *Pattern Recognition*, 2009. 42(12): p. 3287-3295.
- [64] Shayegan, M.A. and C.S. Chan. *A New Approach to Feature Selection in Handwritten Farsi/Arabic Character Recognition*. in *Advanced Computer Science Applications and Technologies (ACSAT), International Conference on*. 2012. IEEE.
- [65] Kim, H.-Y. and J.H. Kim, *Hierarchical random graph representation of handwritten characters and its application to Hangul recognition*. *Pattern Recognition*, 2001. 34(2): p. 187-201.
- [66] Kang, K.-W. and J.H. Kim. *Handwritten hangul character recognition with hierarchical stochastic character*

- representation. in *Document Analysis and Recognition. Proceedings. Seventh International Conference on*. 2003. IEEE.
- [67] Kang, K.-W. and J.H. Kim, *Utilization of hierarchical, stochastic relationship modeling for Hangul character recognition*. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2004. 26(9): p. 1185-1196.
- [68] Park, G.-R., I.-J. Kim, and C.-L. Liu, *An evaluation of statistical methods in handwritten hangul recognition*. *International Journal on Document Analysis and Recognition (IJDAR)*, 2012: p. 1-11.
- [69] John, J., K.V. Pramod, and B. Kannan, *Handwritten Character Recognition of South Indian Scripts: A Review*. *National Conference on Indian Language Computing*, arXiv preprint arXiv:1106.0107, 2011.
- [70] Mukhtar, O., S. Setlur, and V. Govindaraju, *Experiments on Urdu text recognition*, in *Guide to OCR for Indic Scripts*. 2010, Springer. p. 163-171.
- [71] Bhattacharya, U. and B.B. Chaudhuri. *Databases for research on recognition of handwritten characters of Indian scripts*. in *Document Analysis and Recognition, Proceedings. Eighth International Conference on*. 2005.
- [72] Bhaskarabhatla, A.S. and S. Madhvanath. *Experiences in collection of handwriting data for online handwriting*

- recognition in Indic scripts. in Proceedings of the Fourth International Conference on Linguistic Resources and Evaluation (LREC). 2004. Citeseer.*
- [73] Jayadevan, R., et al., *Offline recognition of Devanagari script: A survey. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 2011. 41(6): p. 782-796.
- [74] Sharma, N., et al., *Recognition of off-line handwritten devnagari characters using quadratic classifier, in Computer Vision, Graphics and Image Processing. 2006, Springer. p. 805-816.*
- [75] Hanmandlu, M. and O. Murthy, *Fuzzy model based recognition of handwritten numerals. Pattern Recognition*, 2007. 40(6): p. 1840-1854.
- [76] Arora, S., et al. *A Two Stage Classification Approach for Handwritten Devnagari Characters. in Conference on Computational Intelligence and Multimedia Applications, International Conference on. 2007. IEEE.*
- [77] Arora, S., et al. *Combining multiple feature extraction techniques for handwritten Devnagari character recognition. in Industrial and Information Systems, ICIIS, IEEE Region 10 and the Third international Conference on. 2008. IEEE.*

- [78] Pal, U., et al. *Handwritten numeral recognition of six popular Indian scripts*. in *Document Analysis and Recognition, ICDAR Ninth International Conference on*. 2007. IEEE.
- [79] Pal, U., et al. *Off-line handwritten character recognition of devnagari script*. in *Document Analysis and Recognition, ICDAR, Ninth International Conference on*. 2007. IEEE.
- [80] Pal, U., T. Wakabayashi, and F. Kimura. *Comparative study of Devnagari handwritten character recognition using different feature and classifiers*. in *Document Analysis and Recognition, ICDAR'09. 10th International Conference on*. 2009. IEEE.
- [81] Mukherji, P. and P.P. Rege, *Shape Feature and Fuzzy Logic Based Offline Devnagari Handwritten Optical Character Recognition*. *Journal of Pattern Recognition Research*, 2009. 4: p. 52-68.
- [82] Jangid, M., *Devanagari Isolated Character Recognition by using Statistical features*. *International Journal on Computer Science and Engineering (IJCSE) ISSN*, 2011. 3.
- [83] Shelke, S. and S. Apte, *A multistage handwritten Marathi compound character recognition scheme using neural networks and wavelet features*. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, 2011. 4(1): p. 81-94.

References

- [84] Bhattacharya, U., et al., *A hybrid scheme for handprinted numeral recognition based on a self-organizing network and MLP classifiers*. International Journal of Pattern Recognition and Artificial Intelligence, 2002. 16(07): p. 845-864.
- [85] Rahman, A.F.R., R. Rahman, and M.C. Fairhurst, *Recognition of handwritten Bengali characters: a novel multistage approach*. Pattern Recognition, 2002. 35(5): p. 997-1006.
- [86] Bhowmik, T.K., U. Bhattacharya, and S.K. Parui. *Recognition of Bangla handwritten characters using an MLP classifier based on stroke features*. in *Neural Information Processing*. 2004. Springer.
- [87] Bhattacharya, U., M. Shridhar, and S.K. Parui, *On recognition of handwritten Bangla characters*, in *Computer Vision, Graphics and Image Processing*. 2006, Springer. p. 817-828.
- [88] Pal, U., T. Wakabayashi, and F. Kimura. *Handwritten Bangla compound character recognition using gradient feature*. in *Information Technology,(ICIT). 10th International Conference on*. 2007. IEEE.
- [89] Bhowmik T K, G.P., A. Roy, Parui S. K, *SVM-based hierarchical architectures for handwritten Bangla character recognition*. International Journal of Document Analysis and Recognition, 2009.

- [90] Das, N., et al., *Handwritten Bangla basic and compound character recognition using MLP and SVM classifier*. arXiv preprint arXiv:1002.4040, 2010.
- [91] Das, N., et al. *A Novel GA-SVM Based Multistage Approach for Recognition of Handwritten Bangla Compound Characters*. in *Proceedings of the International Conference on Information Systems Design and Intelligent Applications (INDIA 2012) held in Visakhapatnam, India, January 2012*. Springer.
- [92] Reza, K.N. and M. Khan. *Grouping of Handwritten Bangla Basic Characters, Numerals and Vowel Modifiers for Multilayer Classification*. in *Frontiers in Handwriting Recognition (ICFHR), International Conference on*. 2012.
- [93] Prasad, J.R., U. Kulkarni, and R.S. Prasad. *Template Matching Algorithm for Gujrati Character Recognition*. in *Emerging Trends in Engineering and Technology (ICETET), 2nd International Conference on*. 2009. IEEE.
- [94] Desai, A.A., *Gujarati handwritten numeral optical character reorganization through neural network*. *Pattern Recognition*, 2010. 43(7): p. 2582-2589.
- [95] Roy, K., et al. *Oriya handwritten numeral recognition system*. in *Document Analysis and Recognition, Proceedings. Eighth International Conference on*. 2005. IEEE.

References

- [96] Bhowmik, T.K., et al. *An HMM based recognition scheme for handwritten Oriya numerals.* in *Information Technology, ICIT'06. 9th International Conference on.* 2006. IEEE.
- [97] Pal, U., T. Wakabayashi, and F. Kimura. *A system for off-line Oriya handwritten character recognition using curvature feature.* in *Information Technology,(ICIT). 10th International Conference on.* 2007. IEEE.
- [98] Padhi, D. and D. Senapati. *Zone Centroid Distance and Standard Deviation Based Feature Matrix for Odia Handwritten Character Recognition.* in *Proceedings of the International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA).* 2013. Springer.
- [99] Garg, N. and K. Verma, *Handwritten Gurumukhi character recognition using neural networks.* M. Tech Thesis, Thapar University, 2009.
- [100] Siddharth, K.S., et al., *Handwritten Gurmukhi character recognition using statistical and background directional distribution features.* *International Journal on Computer Science and Engineering*, 2011. 3(6): p. 2332-2345.
- [101] Singh, S., A. Aggarwal, and R. Dhir, *Use of Gabor Filters for Recognition of Handwritten Gurmukhi Character.* *International Journal of Advanced Research in Computer Science and Software Engineering*, 2012. 2(5).

- [102] Sagheer, M.W., et al., *A New Large Urdu Database for Off-Line Handwriting Recognition*, in *Image Analysis and Processing–ICIAP*. 2009, Springer. p. 538-546.
- [103] Yusuf, M. and T. Haider. *Recognition of handwritten Urdu digits using shape context*. in *Multitopic Conference, Proceedings of INMIC. 8th International*. 2004. IEEE.
- [104] Haider, T. and M. Yusuf. *Accelerated recognition of handwritten Urdu digits using Shape Context based gradual pruning*. in *Intelligent and Advanced Systems, ICIAS. International Conference on*. 2007. IEEE.
- [105] Pathan, I.K., A.A. Ali, and R. RJ, *Recognition of offline handwritten isolated urdu character*. *Advances in Computational Research*, ISSN, 2012. 4(1): p. 117-121.
- [106] Paulpandian, T. and V. Ganapathy. *Translation and scale invariant recognition of handwritten Tamil characters using a hierarchical neural network*. in *Circuits and Systems, ISCAS'93, IEEE International Symposium on*. 1993. IEEE.
- [107] Suresh, R., S. Arumugam, and L. Ganesan. *Fuzzy approach to recognize handwritten Tamil characters*. in *Computational Intelligence and Multimedia Applications, ICCIMA'99. Proceedings. Third International Conference on*. 1999. IEEE.

References

- [108] Hewavitharana, S. and H. Fernando, *A two stage classification approach to Tamil handwriting recognition*. Tamil Internet, 2002. p. 118-124.
- [109] Pal, U., et al., *Handwritten character recognition of popular south Indian scripts*, in *Arabic and Chinese Handwriting Recognition*. 2008, Springer. p. 251-264.
- [110] Sutha, J. and N. Ramaraj. *Neural network based offline Tamil handwritten character recognition system*. in *Conference on Computational Intelligence and Multimedia Applications, International Conference on*. 2007. IEEE.
- [111] Bhattacharya, U., S. Ghosh, and S. Parui. *A two stage recognition scheme for handwritten Tamil characters*. in *Document Analysis and Recognition, ICDAR. Ninth International Conference on*. 2007. IEEE.
- [112] Shanthi, N. and K. Duraiswamy, *A novel SVM-based handwritten Tamil character recognition system*. *Pattern Analysis and Applications*, 2010. 13(2): p. 173-180.
- [113] Subashini, A. and N. Kodikara. *A novel SIFT-based codebook generation for handwritten Tamil character recognition*. in *Industrial and Information Systems (ICIIS), 6th IEEE International Conference on*. 2011. IEEE.
- [114] Rajashekararadhya, S. and V. Ranjan. *Zone-based hybrid feature extraction algorithm for handwritten numeral*

- recognition of four Indian scripts.* in *Systems, Man and Cybernetics, SMC. IEEE International Conference on.* 2009. IEEE.
- [115] Pradhan, S.K. and A. Negi. *A syntactic PR approach to Telugu handwritten character recognition.* in *Proceeding of the workshop on Document Analysis and Recognition.* 2012. ACM.
- [116] Soman, S.T., A. Nandigam, and V.S. Chakravarthy. *An efficient multiclassifier system based on convolutional neural network for offline handwritten Telugu character recognition.* in *Communications (NCC), National Conference on.* IEEE.
- [117] Sharma, N., U. Pal, and F. Kimura. *Recognition of handwritten Kannada numerals.* in *Information Technology, ICIT'06. 9th International Conference on.* 2006. IEEE.
- [118] Rajput, G. and M. Hangarge, *Recognition of isolated handwritten Kannada numerals based on image fusion method,* in *Pattern Recognition and Machine Intelligence.* 2007, Springer. p. 153-160.
- [119] Manjunath Aradhya, V., G. Hemantha Kumar, and S. Noushath. *Robust unconstrained handwritten digit recognition using radon transform.* in *Signal Processing, Communications and Networking, ICSCN'07. International Conference on.* 2007. IEEE.

- [120] Rajashekararadhya, S., P. Vanaja Ranjan, and V. Manjunath Aradhya. *Isolated handwritten Kannada and Tamil numeral recognition: A novel approach*. in *Emerging Trends in Engineering and Technology, ICETET'08. First International Conference on*. 2008. IEEE.
- [121] Niranjana, S., et al. *FLD based unconstrained handwritten kannada character recognition*. in *Future Generation Communication and Networking Symposia, FGCNS'08. Second International Conference on*. 2008. IEEE.
- [122] Ragha, L. and M. Sasikumar. *Using moments features from Gabor directional images for Kannada handwriting character recognition*. in *Proceedings of the International Conference and Workshop on Emerging Trends in Technology*. 2010. ACM.
- [123] Rajput, G., R. Horakeri, and S. Chandrakant, *Printed and handwritten mixed Kannada numerals recognition using SVM*. International Journal on Computer Science and Engineering (IJCSSE), 2010. 2(5).
- [124] Rajput, G. and R. Horakeri. *Shape descriptors based handwritten character recognition engine with application to Kannada characters*. in *Computer and Communication Technology (ICCCCT), 2nd International Conference on*. 2011. IEEE.

- [125] Lajish, V.L. *Handwritten Character Recognition Using Gray-scale Based State-Space Parameters and Class Modular NN*. in *Signal Processing, Communications and Networking, ICSCN '08. International Conference on*. 2008.
- [126] Raju, G. *Recognition of Unconstrained Handwritten Malayalam Characters Using Zero-crossing of Wavelet Coefficients*. in *Advanced Computing and Communications, ADCOM. International Conference on*. 2006.
- [127] Raju, G. *Wavelet Transform and Projection Profiles in Handwritten Character Recognition - A Performance Analysis*. in *Advanced Computing and Communications, ADCOM. 16th International Conference on*. 2008.
- [128] Chacko, B.P. and A.P. Babu. *Discrete Curve Evolution Based Skeleton Pruning for Character Recognition*. in *Advances in Pattern Recognition, ICAPR '09. Seventh International Conference on*. 2009.
- [129] Chacko, B.P. and A.P. Babu. *Pre and Post Processing Approaches in Edge Detection for Character Recognition*. in *Frontiers in Handwriting Recognition (ICFHR), International Conference on*. 2010.
- [130] Chacko, B.P., et al., *Handwritten character recognition using wavelet energy and extreme learning machine*. *International Journal of Machine Learning and Cybernetics*, 2011.

References

- [131] Moni, B.S. and G. Raju. *Modified quadratic classifier for handwritten Malayalam character recognition using run length count*. in *Emerging Trends in Electrical and Computer Technology (ICETECT), International Conference on*. 2011. IEEE.
- [132] Moni, B.S. and G. Raju, *Modified quadratic classifier and directional features for handwritten Malayalam character recognition*. *Int. J. Comput. Appl*, 2011: p. 30-34.
- [133] Moni, B.S. and G. Raju. *Handwritten character recognition system using a simple feature*. in *Proceedings of the International Conference on Advances in Computing, Communications and Informatics*. 2012. ACM.
- [134] Govindaraju, V. and S. Setlur, *Guide to OCR for Indic Scripts: Document Recognition and Retrieval*. 2009: Springer Publishing Company, Incorporated. 325.
- [135] *Journal of Language Technology*. in *Technology Development for Indian Languages*, Viswabharat@tdil. July 2003.
- [136] *Malayalam Script - Adoption of New Script for Use*. in *Government of Kerala*, www.malayalamresourcecentre.org/Mrc/order.pdf 1971.
- [137] Neeba, N., *Large Scale Character Classification*. MS Thesis, cvit.iiit.ac.in, 2010.

- [138] Otsu, N., *A threshold selection method from gray level histograms*. IEEE Transactions on System Man and Cybernetics, 1979. 9(1): p. 62-66.
- [139] Gonzalez, R.C., R.E. Woods, and S.L. Eddins, *Digital image processing using MATLAB*. Vol. 2. 2009: Gatesmark Publishing Tennessee.
- [140] Trier, O.D. and T. Taxt, *Evaluation of binarization methods for document images*. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 1995. 17(3): p. 312-315.
- [141] Lam, L., S.-W. Lee, and C.Y. Suen, *Thinning methodologies-a comprehensive survey*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1992. 14(9): p. 869-885.
- [142] Impedovo, S., et al. *Zoning methods for hand-written character recognition: an overview*. in *Frontiers in Handwriting Recognition (ICFHR), International Conference on*. 2010. IEEE.
- [143] Ojala, T., M. Pietikainen, and T. Maenpaa, *Multiresolution gray-scale and rotation invariant texture classification with local binary patterns*. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2002. 24(7): p. 971-987.
- [144] Joseph, S., John, J., Balakrishnan, K., Pramod, K. V., *Content based image retrieval system for Malayalam handwritten*

- characters. in *Electronics Computer Technology (ICECT), 3rd International Conference on*. 2011. IEEE.
- [145] Mallat, S., G, *A theory for multiresolution signal decomposition: The wavelet representation*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1989. 11(7): p. 674-693.
- [146] Stanković, R.S. and B.J. Falkowski, *The Haar wavelet transform: its status and achievements*. Computers & Electrical Engineering, 2003. 29(1): p. 25-44.
- [147] Tang, Y.Y., *Wavelet theory and its applications to pattern recognition*. Vol. 36. 2000: World Scientific.
- [148] Freeman, H., *On the encoding of arbitrary geometric configurations*. Electronic Computers, IRE Transactions on, 1961(2): p. 260-268.
- [149] Bhattacharya, U., S. Ghosh, and S.K. Parui. *A two stage recognition scheme for handwritten Tamil characters*. in *Document Analysis and Recognition, ICDAR 2007. Ninth International Conference on*. 2007. IEEE.
- [150] Liu C L, N.K., Sako H, Fujisawa H, *Handwritten digit recognition: Benchmarking of state-of-the-art techniques*. Pattern Recognition, 2003. 36(10): p. 2271-2285.
- [151] Shi, M., et al., *Handwritten numeral recognition using gradient and curvature of gray scale image*. Pattern Recognition, 2002. 35(10): p. 2051-2059.

- [152] Hailong, L. and D. Xiaoqing. *Handwritten character recognition using gradient feature and quadratic classifier with multiple discrimination schemes*. in *Document Analysis and Recognition, Proceedings. Eighth International Conference on*. 2005.
- [153] Srikantan, G., S.W. Lam, and S.N. Srihari, *Gradient-based contour encoding for character recognition*. *Pattern Recognition*, 1996. 29(7): p. 1147-1160.
- [154] Sobel, I., *Camera models and machine perception*, 1970, DTIC Document.
- [155] Roberts, L.G., *Machine perception of three-dimensional solids*, 1963, DTIC Document.
- [156] Jolliffe, I.T., *Principal Component Analysis*. 2002: Springer.
- [157] Tan, P.-N., *Introduction to data mining*. 2007: Pearson Education India.
- [158] Han, J., M. Kamber, and J. Pei, *Data mining: concepts and techniques*. 2006: Morgan kaufmann.
- [159] Larose, D.T., *Data mining methods & models*. 2006: Wiley.com.
- [160] LeCun, Y., et al., *Backpropagation applied to handwritten zip code recognition*. *Neural computation*, 1989. 1(4): p. 541-551.
- [161] Christopher, B., J, C., *A tutorial on support vector machines for pattern recognition*. *Data Mining and Knowledge discovery*, 1998. 2(2): p. 121-167.

References

- [162] Soman, K., S. Diwakar, and V. Ajay, *Data Mining: Theory and Practice [with CD]*. 2006: PHI Learning Pvt. Ltd.
- [163] Wu, X. and V. Kumar, *The top ten algorithms in data mining*. 2010: CRC Press.
- [164] Lan, Y., et al., *An extreme learning machine approach for speaker recognition*. *Neural Computing & Applications*, 2013: p. 1-9.
- [165] Liu, Y., H. Loh, and S. Tor, *Comparison of extreme learning machine with support vector machine for text classification*. *Innovations in Applied Artificial Intelligence*, 2005: p. 390-399.
- [166] Zheng, W., Y. Qian, and H. Lu, *Text categorization based on regularization extreme learning machine*. *Neural Computing & Applications*, 2012: p. 1-10.
- [167] Kan, E.M., et al., *Extreme learning machine terrain-based navigation for unmanned aerial vehicles*. *Neural Computing & Applications*, 2012: p. 1-9.
- [168] Zhou, Z.H., J.W. Zhao, and F.L. Cao, *Surface reconstruction based on extreme learning machine*. *Neural Computing & Applications*, 2012: p. 1-10.
- [169] Taniar, D., *Data mining and knowledge discovery technologies*. 2008: Igi Publishing.
- [170] Vapnik, V.N., *The Nature of Statistical Learning Theory*, 1999, Springer, Information science and statistics: Berlin.

- [171] Hsu, C.W. and C.J. Lin, *A comparison of methods for multiclass support vector machines*. IEEE Trans Neural Netw, 2002. 13(2): p. 415-25.
- [172] Huang, G.B., Q.Y. Zhu, and C.K. Siew, *Extreme learning machine: theory and applications*. Neurocomputing, 2006. 70(1): p. 489-501.
- [173] Huang, G.B., X. Ding, and H. Zhou, *Optimization method based extreme learning machine for classification*. Neurocomputing, 2010. 74(1): p. 155-163.
- [174] Lin, C.-C.C.C.-J., *LIBSVM: a library for support vector machines*. ACM Transactions on Intelligent Systems and Technology, 2011. 2(27): p. 1-27.
