Offline Handwritten Gurmukhi Script Recognition

A Thesis

Submitted in fulfillment of the requirements for the award of the degree of

Doctor of Philosophy

Submitted by

Munish Kumar

(Registration No. 950811010)

Under the supervision of

Dr. R. K. Sharma
Professor,
Thapar University,
Patiala

Dr. Manish Kumar
Associate Professor,
Panjab University Regional Centre,
Muktsar

Thapar University, Patiala

School of Mathematics and Computer Applications
Thapar University
Patiala–147004 (Punjab) India

July, 2014

CERTIFICATE

I, **Munish Kumar** hereby certify that the work which is being presented in this thesis entitled **"OFFLINE HANDWRITTEN GURMUKHI SCRIPT RECOGNITION"**, in partial fulfillment of requirements for the award of degree of the **DOCTOR OF PHILOSOPHY** in the School of Mathematics & Computer Applications (SMCA), Thapar University, Patiala, is an authentic record of my own work carried under the supervision of **Dr. R. K. Sharma** (Professor, SMCA, Thapar University, Patiala) and **Dr. Manish Kumar** (Associate Professor, Panjab University Regional Centre, Muktsar).

The matter presented in this thesis has not been submitted either in part or full to any other University or Institute for the award of any degree.

14/07/14

Date: July 14, 2014

(Munish Kumar)
Signature of Candidate

This certified that the above statement made by the candidate is correct to the best of my knowledge.

14.7.14

(Dr. R. K. Sharma)
Professor,
School of Mathematics & Computer Applications
Thapar University, Patiala
PIN –147004 (INDIA).
Supervisor

14.07.14

(Dr. Manish Kumar)
Associate Professor,
Computer Science & Applications
Panjab University Regional Centre, Muktsar
PIN –152026 (INDIA).
Supervisor

Dated: July 14, 2014

Abstract

Over the last few years, a good number of laboratories all over the world have been involved in research on handwriting recognition. Handwriting recognition is a complex problem owing to the issues of variations in writing styles and size of the characters *etc*. The main objective of this work is to develop an offline handwritten *Gurmukhi* script recognition system. *Gurmukhi* is the script used for writing *Punjabi* language which is widely spoken in certain regions of north India.

This thesis is divided into eight chapters. A brief outline of each chapter is given in the following paragraphs.

The first chapter introduces the process of OCR and various phases of OCR like digitization, pre-processing, segmentation, feature extraction, classification and post-processing. Applications of offline handwritten character recognition system are also discussed in this chapter. In an overview of *Gurmukhi* script, the nature of handwriting in *Gurmukhi* script and character set of *Gurmukhi* script has also been presented. Major contributions and assumptions in this research work have also been discussed in this chapter.

Chapter 2 contains a review of literature on various methods used for non-Indian and Indian scripts recognition. In this chapter, a detailed literature survey on established procedures for numeral and character recognition techniques has been presented. We have reviewed literature for different scripts, namely, *Arabic*, *Bangla*, *Devanagari*, *French*, *Gujarati*, *Gurmukhi*, *Kannada*, *Japanese*, *Malayalam*, *Oriya*, *Roman*, *Tamil*, *Telugu* and *Thai* in this thesis.

Chapter 3 describes essential phases of an offline handwritten *Gurmukhi* script recognition system. These have been discussed in four sections entitled data collection phase, digitization phase, pre-processing phase and segmentation phase. In data collection phase, we have collected 300 samples of offline handwritten *Gurmukhi* script documents. These documents have been divided into three categories. *Category* 1 consists of one hundred samples of offline handwritten *Gurmukhi* script documents where each *Gurmukhi* script document is written by a single writer. *Category* 2 contains one hundred samples where each

Gurmukhi script document is written ten times by ten different writers. In *category* 3, one *Gurmukhi* script document is written by one hundred different writers. These samples of offline handwritten *Gurmukhi* script documents of different writers have been collected from schools, colleges, government offices and other public places. In digitization phase, the procedure to produce the digital image of a paper based handwritten document has been presented. In pre-processing phase, size normalization and thinning of text has been done. In segmentation phase, a new technique has been proposed for line segmentation of offline handwritten *Gurmukhi* script document. Line segmentation accuracy of about 98.4% has been achieved with the use of this technique. Water reservoir based method has also been implemented for touching character segmentation with an accuracy of 93.5%.

Chapter 4 presents a framework for grading of writers based on offline *Gurmukhi* characters. Samples of offline handwritten *Gurmukhi* characters from one hundred writers have been taken in this work. In order to establish the correctness of our proposed approach, we have also considered *Gurmukhi* characters taken from five *Gurmukhi* fonts. These fonts are: *amrit*, *GurmukhiLys*, *Granthi*, *LMP_TARAN* and *Maharaja* (F_1, F_2, …, F_5, respectively). For training data set of handwriting grading system, we have used printed *Gurmukhi* font *Anandpur sahib*. Some of statistical features, namely, zoning features, diagonal features, directional features, intersection and open end points features have been used to assign a unique classification score to a writer. The gradation results are based on the values obtained by two classifiers, namely, Hidden Markov Model (HMM) and Bayesian classifier.

Chapter 5 presents curve fitting based novel feature extraction techniques, namely, parabola curve fitting based features and power curve fitting based features for offline handwritten *Gurmukhi* character recognition. In order to assess the quality of these features in offline handwritten *Gurmukhi* character recognition, the performance of the recently used feature extraction techniques, namely, zoning features, diagonal features, directional features, transition features and intersection and open end points features have been compared with these proposed feature extraction techniques. Each technique has been tested on 5600 samples of isolated offline handwritten *Gurmukhi* characters. The classifiers that have been employed in this work are *k*-Nearest Neighbours (*k*-NN) and Support Vector Machine (SVM) with three flavors, *i.e.*, Linear-SVM, Polynomial-SVM and RBF-SVM. The proposed system achieves maximum recognition accuracy of 97.9%, 94.6%, 94.0% and 92.3% using *k*-NN,

Linear-SVM, Polynomial-SVM and RBF-SVM classifier, respectively, when power curve fitting based features are used in classification process. As such, the results obtained using power curve fitting based features are promising. It has also been seen that the results achieved using parabola curve fitting based features are also better than the other recently used feature extraction techniques. A maximum recognition accuracy of 95.4% has been achieved when the parabola curve fitting based features were used with *k*-NN classifier.

In Chapter 6, we have presented an offline handwritten *Gurmukhi* character recognition system using zoning based novel feature extraction methods and *k*-fold cross validation technique. In this work, we have used various feature extraction techniques, namely, zoning features, diagonal features, directional features, intersection and open end points features, transition features, shadow features, centroid features, peak extent based features and modified division point based features for offline handwritten *Gurmukhi* character recognition. For classification, we have considered *k*-NN, Linear-SVM, Polynomial-SVM and MLPs classifier. In this study, we have considered 5600 samples of isolated offline handwritten *Gurmukhi* characters. We have concluded that peak extent based features are preeminent features than other feature extraction techniques. Using 5-fold cross validation technique, we have achieved recognition accuracy, with peak extent based features, of 95.6%, 92.4%, 95.5% and 94.7% with Linear-SVM, Polynomial-SVM, *k*-NN and MLPs classifier, respectively.

Chapter 7 presents a Principal Component Analysis (PCA) based offline handwritten *Gurmukhi* character recognition system. PCA is used for extracting more representative features for data analysis and to reduce the dimensions of data. In this work, we have collected 16,800 samples of isolated offline handwritten *Gurmukhi* characters. These samples are of three categories. In *category* 1, each *Gurmukhi* character has been written 100 times by a single writer (5600 Samples). In *category* 2, each *Gurmukhi* character has been written 10 times by 10 different writers (5600 Samples). For *category* 3, we have again collected each *Gurmukhi* character written by 100 writers (5600 Samples). Here, we have also used different combinations of classifiers as LPR (Linear-SVM + Polynomial-SVM + RBF kernel), LRK (Linear-SVM + Polynomial-SVM + *k*-NN), PRK (Polynomial-SVM + RBF-SVM + *k*-NN) and LRK (Linear-SVM + RBF-SVM + *k*-NN) for recognition purpose. We have used different combinations of output of each classifier in parallel and recognition is

done on the basis of voting scheme. The partition strategy for selecting the training and testing patterns has also been experimented in this work. We have used all 16,800 images of offline handwritten *Gurmukhi* characters for the purpose of training and testing. The proposed system achieves a recognition accuracy of 99.9% for *category* 1 samples, of 99.7% for *category* 2 samples and of 92.3% for *category* 3 samples. In this chapter, we have also presented a hierarchical technique for offline handwritten *Gurmukhi* character recognition. In this technique, we have proposed a strong feature set of 105 feature elements using four types of topological features, namely, horizontally peak extent features, vertically peak extent features, diagonal features, and centroid features. We have also applied various feature set reduction techniques, namely, Principal Component Analysis (PCA), Correlation Feature Set (CFS) and Consistency Based Feature Set (CON). We have seen that PCA performs better than other feature selection techniques for character recognition. A maximum recognition accuracy of 91.8% has been achieved with hierarchical technique when we considered PCA based feature set and Linear-SVM classifier with 5-fold cross validation technique.

Finally, Chapter 8 presents the conclusion drawn from the results of various experiments conducted in this thesis. Also, some pointers to the future research on the topics considered in this thesis are discussed briefly.

Acknowledgement

The real spirit of achieving a goal is through the way of excellence and austerous discipline. I would have never succeeded in completing my task without the cooperation, encouragement and help provided to me by a range of personalities.

I have, indeed, been privileged to have worked under the guidance of Dr. R. K. Sharma (Professor, School of Mathematics & Computer Applications, Thapar University, Patiala) and Dr. Manish Kumar (Associate Professor, Department of Computer Science & Applications, Panjab University Regional Centre, Muktsar). I do not find adequate words to express my deep sense of gratitude towards them. Their personal guidance, encouragement, constructive criticism, invaluable feedback and stimulating discussion at all-time have been a source of inspiration to me in my work. This work has become possible only because of their priceless and unvarying efforts.

I extend my gratitude to Dr. Prakash Gopalan, Director, Thapar University, Patiala, for providing me an opportunity in the Thapar University, Patiala to carry out this research work. I also extend my heartiest thanks to the Doctoral Committee for monitoring the progress and providing priceless suggestions for improvement of my Ph.D. research work.

I am deeply grateful to Dr. Manish Kumar for introducing me with Dr. R. K. Sharma and making this research possible. I, gratefully, acknowledge the co-operation to Dr. Rajesh Kumar, Head SMCA, Thapar University, Patiala, for providing me university's resources and the necessary facilities for carrying out this work. I owe a special vote of thanks notably to the writers for writing the Gurmukhi script documents.

I am also grateful to all academic, administrative and technical staff from the Thapar University, Patiala for their encouragement, timely assistance and acquaintance throughout my candidature.

I wish to express my profound gratitude to my parents Smt. Benti Devi and Sh. Sohan Lal who have been a source of inspiration to undertake this work.

Munish Kumar

List of Publications by the Author

Papers in International Journals:

1. **Munish Kumar**, R. K. Sharma and M. K. Jindal, "A Novel Feature Extraction Technique for Offline Handwritten *Gurmukhi* Character Recognition", *IETE Journal of Research*, Vol. 59(6), pp. 687-692, 2013. (SCI-E)

2. **Munish Kumar**, R. K. Sharma and M. K. Jindal, "Efficient Feature Extraction Techniques for Offline Handwritten *Gurmukhi* Character Recognition", *National Academy Science Letters*, Vol. 37 (4), pp. 381-391, 2014. (SCI-E)

3. **Munish Kumar**, M. K. Jindal and R. K. Sharma, "A Novel Hierarchical Techniques for Offline Handwritten Gurmukhi Character Recognition", *National Academy Science Letters*, Vol. 37 (6), pp. 567-572, 2014. (SCI-E)

4. **Munish Kumar**, M. K. Jindal and R. K. Sharma, "A Novel Technique for Line Segmentation in Offline Handwritten *Gurmukhi* Script Documents", *INFORMATION - An International Interdisciplinary Journal*, 2013 (Accepted for publication) (SCI-E).

5. **Munish Kumar**, M. K. Jindal and R. K. Sharma, "MDP Feature Extraction Technique for Offline Handwritten *Gurmukhi* Character Recognition", *Smart Computing Review,* Vol. 3(6), pp. 397-404, 2013.

6. **Munish Kumar**, M. K. Jindal and R. K. Sharma, "Segmentation of Isolated and Touching Characters in Offline Handwritten *Gurmukhi* Script Recognition", *International Journal of Information Technology and Computer Science*, Vol. 6(2), pp. 58-63, 2014.

7. **Munish Kumar**, R. K. Sharma and M. K. Jindal, "A Framework for Grading Writers using Offline *Gurmukhi* Characters", *International Journal of Pattern Recognition and Image Analysis*, 2014 (Communicated).

Papers Published in conference proceedings (full length):

8. **Munish Kumar**, M. K. Jindal and R. K. Sharma, "Review on OCR for Documents in Handwritten *Gurmukhi* Scripts", *Proceedings of the National Conference on Recent Advances in Computational Techniques in Electrical Engineering*, SLIET Longowal, pp. 1-6, 2010.

9. **Munish Kumar**, R. K. Sharma and M. K. Jindal, "Segmentation of Lines and Words in Handwritten *Gurmukhi* Script Documents", *Proceedings of the First International Conference on Intelligent Interactive Technologies and Multimedia,* Allahabad, pp. 28-30, 2010. (ACM New York, NY, USA ©2010, ISBN: 978-1-4503-0408-5, DOI-10.1145/1963564.1963568)

10. **Munish Kumar**, M. K. Jindal and R. K. Sharma, "Review on OCR for Handwritten Indian Scripts Character Recognition", *Proceedings of the First International Conference on Digital Image Processing and Pattern Recognition, DPPR*, Tirunelveli, Tamil Nadu, Vol. 205, pp. 268-276, 2011. (Springer Berlin Heidelberg ©2011, Print ISBN: 978-3-642-24054-6, Online ISBN: 78-3-642-24055-3, DOI-10.1007/978-3-642-24055-3_28)

11. **Munish Kumar**, M. K. Jindal and R. K. Sharma, "*k*-Nearest Neighbor Based Offline Handwritten *Gurmukhi* Character Recognition", *Proceedings of International Conference on Image Information Processing*, Jaypee University of Information Technology, Waknaghat (Shimla), pp. 1-4, 2011. (IEEE, Print ISBN: 978-1-61284-859-4, INSPEC Accession Number: 12459905, DOI-10.1109/ICIIP.2011.6108863)

12. **Munish Kumar**, R. K. Sharma and M. K. Jindal, "Classification of Characters and Grading Writers in Offline Handwritten *Gurmukhi* Script", *Proceedings of International Conference on Image Information Processing*, Jaypee University of Information Technology, Waknaghat (Shimla), pp. 1-4, 2011. (IEEE, Print ISBN: 978-1-61284-859-4, INSPEC Accession Number: 12459901, DOI-10.1109/ICIIP.2011.6108859)

13. **Munish Kumar**, R. K. Sharma and M. K. Jindal, "SVM based Offline Handwritten *Gurmukhi* Character Recognition", *Proceedings of International Workshop on Soft Computing Applications and Knowledge Discovery*, National Research University Higher School of Economics, Moscow (Russia), pp. 51-62, 2011. (University Higher School of Economics, ISSN: 1613-0073)

14. **Munish Kumar**, M. K. Jindal and R. K. Sharma, "Offline Handwritten *Gurmukhi* Character Recognition Using Curvature Feature", *Proceedings of International Conference on Advances in Modeling, Optimization and Computing*, IIT Roorkee, pp. 981-989, 2011.

15. **Munish Kumar**, M. K. Jindal and R. K. Sharma, "*Weka* based Offline Handwritten *Gurmukhi* Character Recognition", *Proceedings of International Conference on Soft Computing for Problem Solving*, JK Lakshmipat University, Jaipur, pp. 711-722, 2012.

16. **Munish Kumar**, M. K. Jindal and R. K. Sharma, "Offline Handwritten *Gurmukhi* Character Recognition: Study of different features and classifiers combinations", *Proceedings of Workshop on Document Analysis and Recognition*, IIT Bombay, pp. 94-99, 2012. (ACM New York, NY, USA ©2012, ISBN: 978-1-4503-1797-9, DOI-10.1145/2432553.2432571)

List of Figures

List of Tables

Abbreviations

2D	Two Dimensional
AI	Artificial Intelligence
ANN	Artificial Neural Networks
ART	Adaptive Resonance Theory
BPNN	Back Propagation Neural Network
CFS	Correlation Feature Set
CON	Consistency Based
DPI	Dots Per Inch
DTW	Dynamic Time Warping
HCR	Handwritten Character Recognition
HMM	Hidden Markov Model
HP	Horizontal Projection
k-NN	*k*-Nearest Neighbours
LPK	Linear Kernel-SVM + Polynomial Kernel-SVM + *k*-NN
LPR	Linear Kernel-SVM + Polynomial Kernel-SVM + RBF Kernel-SVM
LRK	Linear Kernel SVM + RBF Kernel SVM + *k*-NN
MLP	Multi-Layer Perceptron
NN	Nearest Neighbours
OCR	Optical Character Recognition
PCA	Principal Component Analysis

PRK	Polynomial Kernel-SVM + RBF Kernel-SVM + *k*-NN
RBF	Radial Bias Function
SSM	State Space Map
SSPD	State Space Point Distribution
SVM	Support Vector Machines
VP	Vertical Projection

CONTENTS

Chapter 1
Introduction

==========

Nowadays, computers have a great influence on us and we process almost all the important works of our lives electronically. Keeping in mind the usage of computers these days, we need to develop efficient, easy and fast methods for data transfer between human beings and computers. Document Analysis and Recognition (DAR) systems play a major role in data transfer between human beings and computers. Optical Character Recognition (OCR) system is an essential part of a document analysis and recognition system. OCR systems have been developed to recognize printed texts as well as handwritten texts. Handwritten text recognition systems essentially provide an interface for improving communication between users and computers. These empower computers to read and process handwritten documents. These systems shall further contribute significantly in bridging the gap between man and machine. Although, many researchers have worked to recognize the characters of Indian scripts, the problem of data exchanging between people and machines is still a challenge in these scripts. The work carried out in this thesis addresses the problem of handwritten character recognition for *Gurmukhi* script. *Gurmukhi* script is used to write *Punjabi* language. This language is one of the official languages of India. *Gurmukhi* script is the tenth most widely used script in the world.

1.1 Background of character recognition systems

Character recognition is a process that associates a predefined code to the objects (*letters, symbols* and *numerals*) drawn on a surface (electronic or paper). Research work in

the field of character recognition has been going on at a rapid pace throughout the world since the late sixties. Due to the complex nature of character recognition field, it is an active area of research even now. Character recognition systems can be classified into a number of categories based on data acquisition process, as shown in Figure 1.1.
In further sub-sections, these categories have been described, in brief.

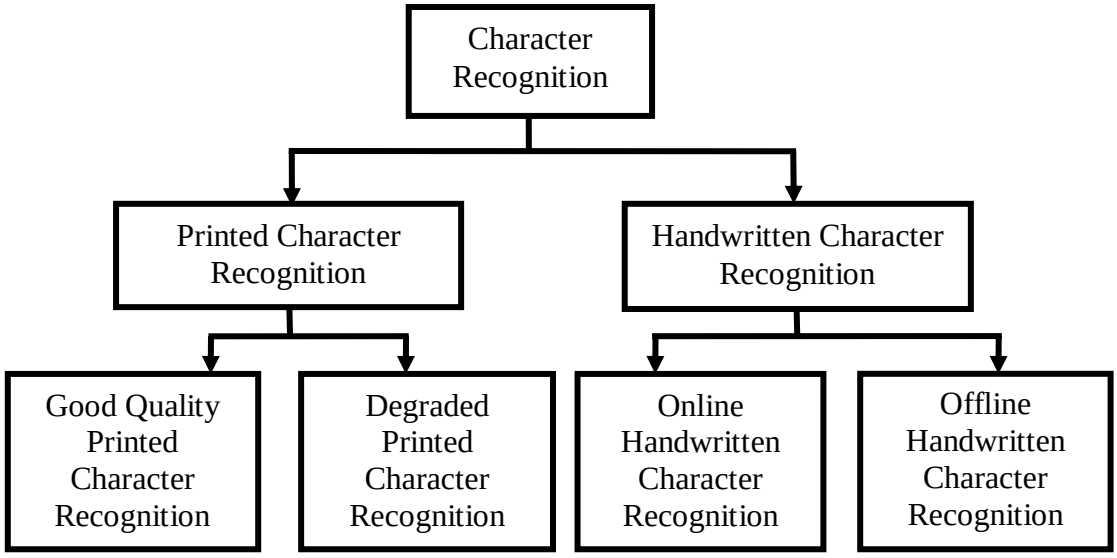


Figure 1.1: Character recognition systems, a representative classification

1.1.1 Printed character recognition

Printed character recognition systems deal with the recognition of machine printed characters. In these systems, a printed document is first scanned and converted into a machine processable format. These machine processable images are further pre-processed and segmented to character level for extracting features from it. These features are now used to recognize a character. As of now, the problem of printed character recognition has considerably been solved. Various commercial and precise systems are now available for printed text recognition.

The printed character recognition can be divided further into two types: good quality printed character recognition and degraded printed character recognition.

1.1.1.1 Good quality printed character recognition

Good quality printed characters are those characters that are noiseless, sharp and well printed. A reasonably good recognition accuracy has been achieved by researchers for this category of characters for various scripts around the world. This accuracy is, probably, sufficient for many real-life applications.

1.1.1.2 Degraded printed character recognition

Degraded printed characters include touching characters, broken characters, heavily printed characters and characters with backside text visibility. These types of degradations in the scanned text image occur from various sources, such as defects in the paper, defects introduced during printing, defects introduced during digitization through scanning and defects introduced during copying through photocopiers and fax machines. One needs to address these issues while dealing with degraded printed character recognition.

1.1.2 Handwritten character recognition

Handwritten character recognition systems deal with the recognition of characters that are written by users on a paper or on an electronic surface using a special device. Unfortunately, achievements acquired in the printed character recognition systems cannot be transmitted automatically to the handwritten character recognition systems. Handwritten character recognition has two streams: online handwritten character recognition and offline handwritten character recognition. These are described, in brief, in the next two sub-sections.

1.1.2.1 Online handwritten character recognition

In online handwritten character recognition, one writes on an electronic surface with the help of a special pen and the data, in the form of (x, y) coordinates, is captured during the writing process. A number of devices including personal digital assistant and tablet PCs are available these days that can be used for data capturing. In these systems, characters are

captured as a sequence of strokes. Features are then extracted from these strokes and strokes are recognized with the help of these features (Table 1.1). Generally, a post-processing module helps in forming the characters from the stroke(s).

1.1.2.2 Offline handwritten character recognition

Offline Handwritten Character Recognition system, commonly abbreviated as offline HCR, is the process of converting offline handwritten text into a format that is understood by machine. It involves processing of documents containing scanned images of a text written by a user, generally on a sheet of paper. In this kind of systems, characters are digitized to obtain 2*D* images.

Table 1.1 shows the comparison between online handwritten character recognition and offline handwritten character recognition. As given in Table 1.1, offline handwritten character recognition is significantly different from online handwritten character recognition, because here, stroke information is not available. Recognition speed and accuracy of offline handwritten character recognition system is also less than online handwritten character recognition system.

Table 1.1: Comparison between online and offline handwritten character recognition

Sr. No.	Basis of comparison	Online handwritten character recognition	Offline handwritten character recognition
1.	Stroke information	Yes	No
2.	Description of raw data	Number of samples/second + Number of dots/inch	Number of dots/inch
3.	Writing media	Digital pen on an electronic surface	Paper document
4.	Recognition speed	Sufficiently high	Low
5.	Accuracy	Sufficiently high	Low

Most of the published work on optical character recognition of Indian scripts deals with printed characters whereas a few articles deal with the handwritten character recognition problem. These articles mainly deal with *Bangla, Devanagari* and *Kannada* scripts. The

pioneering work in *Bangla, Devanagari* and *Kannada* scripts has been done by Pal and Chaudhuri (1994), Bansal and Sinha (2002) and Ashwin and Sastry (2002), respectively.

The work presented in this thesis is an effort towards the recognition of offline handwritten *Gurmukhi* script. This work will also facilitate the progress of the expansion of such systems for recognition of handwritten texts of other Indian scripts that are structurally similar to *Gurmukhi* script.

Recognition of offline handwritten documents is an active research area in the field of pattern recognition. Over the last couple of years, a number of laboratories all over the world have been involved in research on handwriting recognition. The recognition of cursive handwriting is very difficult due to a large number of variations found in shapes and overlapping of characters. In the offline handwriting recognition system, the pre-written text document is converted into a digital image through an optical scanner. In a handwritten text, there is a good amount of variation in the writing style and size of a character *etc*.

One can find handwritten documents at various places such as post offices, banks, insurance offices, and colleges *etc*. In these places, one may note that a huge amount of handwritten data is being generated in the form of faxes, data collected through forms, postal addresses, signatures *etc*. In these circumstances, this proposed research would be highly beneficial for the recognition of handwritten documents. Numerous researchers have been trying to tackle this problem, but an integrated solution to this problem has not been achieved yet.

This research work describes the design of a system that can convert offline handwritten *Gurmukhi* script documents into a machine processable format. Offline handwritten *Gurmukhi* script recognition offers a new way to improve the interface between human beings and computers. As such, in this work, we have focused on offline handwritten *Gurmukhi* script recognition. *Gurmukhi* script has been introduced in Section 1.4.

1.2 Stages of an offline handwritten character recognition system

A typical offline handwritten character recognition system involves activities, namely, digitization, pre-processing, segmentation, feature extraction, classification and post-processing. The sequence of these activities is shown in Figure 1.2.

1.2.1 Digitization

Converting a paper based handwritten document into an electronic form is referred as digitization. The electronic conversion is carried out using a process wherein a document is scanned and then a bitmap image of the original document is produced. Digitization yields the digital image which is then fed to the pre-processing phase.

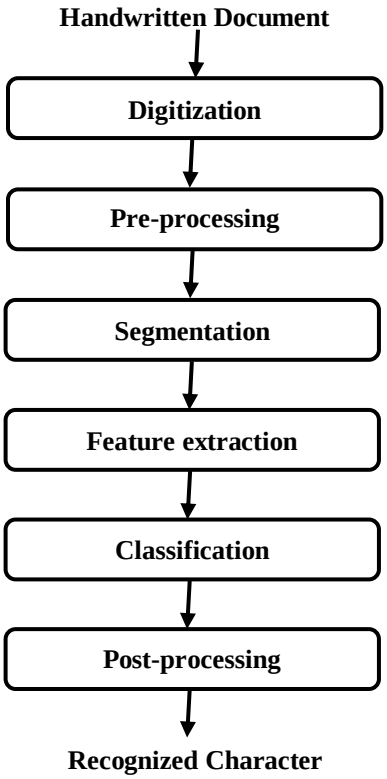


Figure 1.2: Block diagram of offline HCR system

1.2.2 Pre-processing

Pre-processing is the preliminary stage of character recognition. It encompasses skew detection and correction, skeletonization, and noise reduction/removal. Skewness means the tilt of the bit mapped image of the scanned document. It usually surfaces when the document is not correctly fed to the scanner. Skeletonization is applied in order to decrease the line width of the text from several pixels to a single pixel. Noise removal is carried out to remove those unwanted bits that do not play a substantial role in the document. After pre-processing, we have the digital document that is inputted to the segmentation phase.

1.2.3 Segmentation

In character recognition, the process of segmentation plays a very important role. Segmentation is used to break the document into lines, words and characters (*akhars*). For the task of segmentation, an algorithm is used to find the segmentation points in a handwritten document. *Gurmukhi* script document can be segmented into paragraphs, lines, words and characters. The challenge of a segmentation technique lies in the detection of the best segmentation point for lines, words and characters in isolation. Incorrect segmentation can lead to the incorrect recognition. Segmentation of a handwritten text is a challenging task owing to a variety of writing styles.

1.2.4 Feature extraction

Feature extraction is an important task of the recognition process which is used to measure the relevant shape contained in the character. In the feature extraction phase, one can extract the features of the character. The performance of the recognition system depends on features which are being extracted. In OCR applications, it is essential to extract those features that will make possible the system which can differentiate between all the character classes that exist. The extracted features may be structural or statistical based. Structural features depict a pattern in terms of its topology and geometry by giving it local and global

properties. Characteristics of the distribution of pixel values on the bitmap image are captured as statistical features.

1.2.5 Classification

Classification phase is the phase of an OCR system wherein one makes the decisions. It uses the features extracted in the feature extraction stage, for making class membership in the recognition system. The preliminary aim of the classification phase of an OCR system is to develop a constraint that can help to reduce the misclassification relevant to feature extraction. Effectiveness of any character recognition system is highly dependent on the capability of identifying the unique features of a character and the capability of the classifier to relate features of a character to its class. Various classification methods, namely, *k*-Nearest Neighbours (*k*-NN), Hidden Markov Model (HMM), Support Vector Machines (SVMs) and Bayesian *etc*. exist in literature.

1.2.6 Post-processing

OCR results, in general, contain errors since classification phase does not always give one hundred percent accurate results. To further refine the results of classification, post-processing is applied. There are two most commonly used post-processing techniques for error correction. These are (i) dictionary lookup and (ii) statistical approach (Lehal and Singh, 2002).

1.3 Applications of offline handwritten character recognition system

In this section, we have discussed some important applications of offline handwritten character recognition system.

- **Handwritten notes reading:** Offline handwritten character recognition system can be used for reading handwritten notes. Notes are, normally, used to record facts, topics, or thoughts, written down as an assist to memory.

- **Cheque reading**: Offline handwritten character recognition system can be used for cheque reading in banks. Cheque reading is a very important commercial application of offline handwritten character recognition system. Offline handwritten character recognition system plays a very important role in banks for signature verification and for recognition of amount filled by user.

- **Postcode recognition:** Offline handwritten character recognition system can be used for reading handwritten postal address on letters. Offline handwritten character recognition system can also be used for recognition of handwritten digits of postcodes. This system can read these codes and help to sort mails automatically.

- **Form processing:** Offline handwritten character recognition system can also be used for form processing. Forms are normally used to collect information from the public. This information can be processed by using a handwritten character recognition system.

- **Signature verification:** Offline handwritten character recognition system can be used to identify a person through her signature. Signature identification is the specific field of handwriting OCR in which the writer is verified by some specific handwritten text. Offline handwritten character recognition system can be used to identify a person by handwriting, as handwriting varies from person to person.

1.4 Overview of the *Gurmukhi* script

Gurmukhi script is the script used for writing *Punjabi* language and is derived from the old *Punjabi* term "*Guramukhi*", which means "from the mouth of the Guru". *Gurmukhi* script is the 10th most widely used script in the world [Source: Growth of Scheduled Languages: 1971, 1981, 1991, 2001 and 2011, Census of India, Ministry of Home Affairs, Government of India]. The writing style of the *Gurmukhi* script is from top to bottom and left to right. *Gurmukhi* script has three vowel bearers, thirty two consonants, six additional consonants, nine vowel modifiers, three auxiliary signs, and three half characters. In *Gurmukhi* script, there is no case sensitivity. The character set of the *Gurmukhi* script is given in Table 1.2.

The present study is focused on recognizing these characters written in offline handwriting mode.

Table 1.2: *Gurmukhi* characters and their names

S. No.	Character	Character name	S. No.	Character	Character name
1	ੳ	*ūṛā*	2	ਅ	*aiṛā*
3	ੲ	*īṛī*	4	ਸ	*sassā*
5	ਹ	*hāhā*	6	ਕ	*kakka*
7	ਖ	*khakkhā*	8	ਗ	*gaga*
9	ਘ	*ghaggā*	10	ਙ	*ṅaṅṅā*
11	ਚ	*caccā*	12	ਛ	*chacchā*
13	ਜ	*jajjā*	14	ਝ	*jhajjā*
15	ਞ	*naññā*	16	ਟ	*ṭaiṅkā*
17	ਠ	*ṭhaṭṭhā*	18	ਡ	*ḍaḍḍā*
19	ਢ	*ḍhaḍḍā*	20	ਣ	*ṇāṇā*
21	ਤ	*tattā*	22	ਥ	*thatthā*
23	ਦ	*daddā*	24	ਧ	*dhaddā*
25	ਨ	*nannā*	26	ਪ	*papa*
27	ਫ	*phapphā*	28	ਬ	*babbā*
29	ਭ	*bhabbhā*	30	ਮ	*mamma*
31	ਯ	*yayyā*	32	ਰ	*rārā*
33	ਲ	*lallā*	34	ਵ	*vāvā*
35	ੜ	*ṛāṛā*	36	ਸ਼	*shashshā*

37	ਜ਼	*zazzā*	38	ਖ਼	*k̲ha k̲k̲hā*
39	ਫ਼	*faffā*	40	ਗ਼	*gaga*
41	.ਲ	*ḷaḷḷā*			

There are some special characters and vowels in *Gurmukhi* script. These special characters and vowels used in *Gurmukhi* script have been listed in Table 1.3.

Table 1.3: Special *Gurmukhi* characters and their names

S. No.	Character	Character name		S. No.	Character	Character name	
1	ਾ	*kannā*	Vowel	8	ੰ	*ṭippī*	Vowel
2	ਂ	*lāṃvāṃ*	Vowel	9	ੁ	*muktā*	Vowel
3	ੱ	*dulāṃvāṃ*	Vowel	10	ੂ	*auṅkaṛ*	Vowel
4	ਿ	*sihārī*	Vowel	11	ੋ	*dulaiṅkaṛ*	Vowel
5	ੀ	*bihārī*	Vowel	12	੍	*pairīṃ rārā*	Special character
6	ੇ	*hōṛā*	Vowel	13	੍	*pairīṃ hāhā*	Special character
7	ੈ	*kanauṛā*	Vowel	14	ਂ	*bindī*	Vowel

In *Gurmukhi* script, most of the characters have a horizontal line at the upper part called headline, and thus, characters are connected with each other through this line. Figure 1.3 illustrates a sample offline handwritten *Gurmukhi* script document.

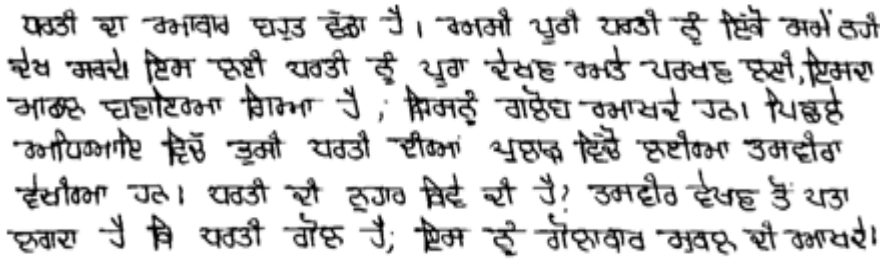


Figure 1.3: Sample of handwritten *Gurmukhi* script document

Gurmukhi words can be divided into three different zones: the upper zone, the middle zone and the lower zone-as shown in Figure 1.4.

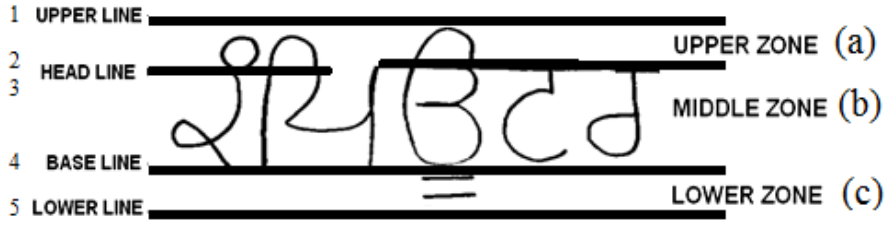


Figure 1.4 *Gurmukhi* script word (ਕੰਪਿਊਟਰ): (a) upper zone from line number 1 to 2, (b) middle zone from line number 3 to 4, (c) lower zone from line number 4 to 5

1.5 Objectives of this work

The objectives of the proposed study are outlined as follows:

1. To study and implement existing algorithms and methods like Bengal Engineering and Science University Shibpur (BESUS) algorithm, Democritus University of THrace-Adaptive Run Length Smearing Algorithm (DUTH-ARLSA), Institute of Language and Speech Processing-Line and Word Segmentation (ILSP-LWSeg) algorithm, and University of Athens-Hough Transformation (UoA-HT) for line and word segmentation of a handwritten document. A new algorithm will be proposed for segmentation of lines for the offline handwritten *Gurmukhi* script document.

2. To explore existing features (structural and statistical) and to propose innovative features for offline handwritten *Gurmukhi* script recognition.
3. To explore HMM, ANN, *k*-NN and SVM classifiers and to propose efficient combinations of these in the form of multiple classifiers.

In order to achieve these objectives, a detailed survey of literature on different stages of a Handwritten Character Recognition (HCR) system has been done. Statistical features have been used for constructing a feature vector for recognition purpose. Various classifiers such as *k*-NN, HMM, SVM, Bayesian and MLP have been used for recognition purpose. Combinations of these classifiers have also been employed for offline handwritten *Gurmukhi* character recognition in the work carried out for this thesis.

1.6 Assumptions

We have considered the following constraints while performing experiments in this thesis.

1. The handwritten documents have been scanned at 300 dpi resolution.
2. The text considered in this work is free from noise. Skew detection/correction is also not required.
3. The data considered in this work does not contain any non-text items such as images, figures *etc*.

1.7 Major contributions and achievements

The major contributions of this thesis can be summarized as follows:

1. A detailed survey of literature on different stages of an OCR system has been done. Also, survey of literature of Indian scripts recognition and handwritten text recognition has been done.
2. Various segmentation algorithms such as strip based projection profiles, smearing technique, and water reservoir based concept have been used for segmentation of offline handwritten *Gurmukhi* script document.
3. Statistical features have been used for constructing a feature vector for recognition purpose.
4. Efficient features, namely, parabola curve fitting based features, power curve fitting based features, peak extent based features and modified division points based features based on the statistical properties, have been proposed for offline handwritten *Gurmukhi* character recognition.
5. A handwriting grading system based on *Gurmukhi* characters using HMM and Bayesian classifiers has been presented.
6. A system for offline handwritten *Gurmukhi* character recognition based on PCA is also presented.

7. Various classifiers such as *k*-NN, HMM, SVMs, Bayesian and MLP have been used for recognition purpose.
8. Combinations of different classifiers have also been employed for offline handwritten *Gurmukhi* character recognition.

1.8 Organization of thesis

The main objective of this thesis is to develop an offline handwritten *Gurmukhi* script recognition system. We have recognized offline handwritten *Gurmukhi* characters with different recognition methods. We have also implemented a handwriting grading system based on offline *Gurmukhi* characters. The organization of the thesis is briefly outlined as below.

In the present chapter, we have discussed various phases of OCR system and issues related to offline handwritten *Gurmukhi* script recognition. An overview of *Gurmukhi* script is also given in this chapter. In Chapter 2, a review of literature has been presented on non-Indian and Indian scripts recognition systems. This review of literature provides essential background knowledge of handwriting recognition. In addition, we have also presented recognition results of other researchers reported in literature. Chapter 3 demonstrates the data collection, digitization and pre-processing of documents. In this chapter, we have explained procedures to segment offline handwritten *Gurmukhi* script document into lines, words and characters with methods, such as, strip based projection profiles, smearing technique, and water reservoir based concept. In Chapter 4, we have presented a framework for grading of writers based on offline *Gurmukhi* characters. In this work, HMM and Bayesian decision making classifiers have been used for classification and grading. In Chapter 5, we have presented two novel feature extraction techniques, namely, parabola curve fitting based features and power curve fitting based features. In this chapter, we have also analyzed the performance of other recently used feature extraction techniques, namely, zoning features, diagonal features, directional features, intersection and open end points features; and transition features. Chapter 6 demonstrates an offline handwritten *Gurmukhi* character recognition system using *k*-fold cross validation technique. In this work, we have explored

shadow features, centroid features, peak extent based features and modified division points based features. In Chapter 7, we have presented a PCA based offline handwritten *Gurmukhi* character recognition system and have also proposed a hierarchical feature extraction technique for offline handwritten *Gurmukhi* character recognition. In this chapter, we have used *k*-NN, Linear-SVM, Polynomial-SVM and RBF-SVM classifiers and their combinations for classification purpose. Various feature selection techniques have also been used in this work. Finally, we have presented the conclusions and the future scope of this work in Chapter 8.

Chapter 2
Review of Literature

==========

In this work, literature review has been divided into two parts. In first part, the literature dealing with the recognition of non-Indian scripts has been reviewed; and in the second part, literature review has been carried out for Indian scripts. The efforts in these two directions are included in section 2.1 and section 2.2 of this chapter. Section 2.3 of this chapter is devoted to brief explanation of existing algorithms used in this work; section 2.4 presents recognition accuracy achieved for different scripts; section 2.5 presents recognition accuracy achieved for complete set of *aksharas* in different scripts and section 2.6 presents a summary of this chapter.

2.1 Recognition of non-Indian scripts

2.1.1 *Arabic*

Arabic script is used for writing *Arabian* and *Persian* languages. Almuallim and Yamaguchi (1987) have presented a recognition system for *Arabic* script. They have used geometrical and topological features for recognition. Impedovo and Dimauro (1990) have proposed a method based on Fourier descriptors for recognition of handwritten *Arabic* numerals. Roy *et al.* (2004a) have presented a postal automation system for sorting of postal documents written in *Arabic*. They have employed a two-stage Multi-Layer Perceptron (MLP) based classifier to recognize *Bangla* and *Arabic* numerals. They have obtained maximum recognition accuracy of about 92.1% for handwritten numerals. Lorigo and Govindaraju (2006) have presented a critical review on offline *Arabic* handwriting recognition systems. They have presented various techniques used for different stages of the

offline handwritten *Arabic* character recognition system. Izadi *et al.* (2006) addressed the issues in *Arabic* alphabet, adopted and evolved, for writing *Persian* language. Abd and Paschos (2007) have achieved a recognition accuracy of 99.0% with Support Vector Machine (SVM) for *Arabic* script. Alaei *et al.* (2009) have presented a handwritten *Arabic* numeral recognition system using a five-fold cross validation technique. They have achieved a recognition accuracy of 99.4% on a 10-class problem with 20,000 samples in testing data set. Alaei *et al.* (2010a) have proposed a technique for segmentation of handwritten *Persian* script text lines into characters. The proposed algorithm finds the baseline of the text image and straightens it. They have extracted features using histogram analysis and removed segmentation points, using baseline dependent as well as language dependent rules. They have achieved a maximum segmentation accuracy of 92.5%. Alaei *et al.* (2010b) have proposed an isolated handwritten *Persian* character recognition system. They employed SVM for classification and achieved a recognition accuracy of 98.1% with modified chain code features. Kacem *et al*. (2012) have used structural features for recognition of *Arabic* names.

2.1.2 *French*

Grosicki and Abed (2009) proposed a *French* handwriting recognition system in a competition held in ICDAR-2009. In this competition, they have presented comparisons between different classification and recognition systems for *French* handwriting recognition. Tran *et al.* (2010) have discussed the problem of *French* handwriting recognition using 24,800 samples. They have worked on both, online and offline handwritten character recognition.

2.1.3 *Japanese*

Nakagawa *et el*. (2005) have presented a model for online handwritten *Japanese* text recognition which is free from line direction constrains and writing format constraints. Zhu *et al*. (2010) have described a robust model for online handwritten *Japanese* text recognition. They obtained a recognition accuracy of 92.8% using 35,686 samples.

2.1.4 *Roman*

Schomaker and Segers (1999) have proposed a technique for cursive *Roman* handwriting recognition using geometrical features. Park *et al*. (2000) have presented a hierarchical character recognition system for achieving high speed and accuracy by using a multi-resolution and hierarchical feature space. They obtained a recognition rate of about 96%. Wang *et al.* (2000) have presented a technique for recognition of *Roman* alphabets and numeric characters. They achieved a recognition rate of about 86%. Bunke and Varga (2007) have reviewed the state of the art in offline *Roman* cursive handwriting recognition. They identified the challenges in *Roman* cursive handwriting recognition. Liwicki and Bunke (2007) have combined the online and offline *Roman* handwriting recognition systems using a new multiple classifier system. They obtained a maximum recognition accuracy of 66.8% for the combination of online and offline handwriting recognition. Schomaker (2007) has presented a method for retrieval of handwritten lines of text in historical administrative documents.

2.1.5 *Thai*

Karnchanapusakij *et al*. (2009) have used linear interpolation approach for online handwritten *Thai* character recognition. They have achieved a recognition accuracy of 90.9%.

2.2 Recognition of Indian scripts

2.2.1 *Bangla*

A good number of researchers have worked for recognition of handwritten characters in *Bangla* script. *Bangla* script is used for writing *Bengali* and *Assamese* languages. Dutta and Chaudhury (1993) have presented a method for isolated *Bangla* alphabets and numerals recognition using curvature features. Pal and Chaudhuri (1994) have proposed a character recognition method using tree classifier. Their method is reported to be fast because pre-

processing like thinning is not necessary in their scheme. They have achieved a recognition accuracy of 96.0% using 5,000 characters data set. Bishnu and Chaudhuri (1999) have used a recursive shape based technique for segmentation of handwritten *Bangla* script documents. Pal *et al.* (2003) have proposed a technique for segmentation of unconstrained *Bangla* handwritten connected numerals. They achieved a segmentation accuracy of 94.8%. Roy *et al.* (2004b) have presented a handwritten numeral recognition system for Indian postal automation and achieved a recognition accuracy of 92.1%. They first decompose the image into blocks using Run Length Smearing Algorithm (RLSA). Now, non-text blocks are detected using the black pixel density and number of components inside a block. Bhattacharya *et al.* (2006) have proposed a scheme for *Bangla* character recognition for 50-class problem. They have achieved a recognition accuracy of 94.7% and 92.1% for training and testing, respectively. Pal *et al*. (2006a) have proposed a technique for slant correction of *Bangla* characters based on Modified Quadratic Discriminant Function (MQDF). They have tested their system with *Bangla* city name images and achieved a recognition accuracy of 87.2%. Bhattacharya *et al.* (2007) have proposed an approach for online *Bangla* handwritten character recognition. They developed a 50-class recognition problem and achieved an accuracy of 92.9% and 82.6% for training and testing, respectively. Pal *et al.* (2007a) dealt with recognition of offline handwritten *Bangla* compound characters using MQDF. The features used for recognition are mainly based on directional information obtained from the arc tangent of the gradient. They obtained 85.9% recognition accuracy using 5-fold cross validation. Pal *et al.* (2008) have proposed a technique for *Bangla* handwritten pin code recognition system. Bhowmik *et al.* (2009) have presented a SVM based hierarchical classification scheme for recognition of handwritten *Bangla* characters. They have achieved accuracies of MLP, RBF and the SVM classifiers are 71.4%, 74.6% and 79.5%, respectively.

Reddy *et al.* (2012a) have presented a handwritten numeral recognition system that can be used for both online and offline situations for *Assamese* language. For online handwritten numeral recognition, they have used x and y coordinates for feature extraction and HMM classifier for recognition. For offline numeral recognition, they have considered projection profile features, zonal discrete cosine transforms, chain code histograms and pixel level features and Vector Quantization (VQ) classifier for recognition. They have achieved a

recognition accuracy of 96.6% and 97.6% for online and offline handwritten numerals, respectively. Reddy *et al.* (2012b) have also presented an HMM based online handwritten digit recognition system using first and second order derivatives at each point as features. They obtained a recognition accuracy of 97.1% on 18,000 samples testing data set. Sarma *et al*. (2013) have presented a handwritten *Assamese* numeral recognition system using HMM and SVM classifiers. They have achieved a recognition accuracy of 96.5% and 96.8% with HMM and SVM classifiers, respectively.

2.2.2 *Devanagari*

Devanagari script is used for writing four languages, namely, *Hindi*, *Marathi*, *Nepali* and *Sanskrit*. Sethi and Chatterjee (1976) have reported work on *Devanagari* numeral recognition. They have used binary decision tree classifier for recognition. Bansal and Sinha (2000) have also developed a technique for *Devanagari* text recognition. In this technique, they recognize a character in two steps. In the first step, they recognize the unknown stroke and in the second step, they recognize the character based on strokes recognized in the first step.

Joshi *et al.* (2005) have presented an online handwritten *Devanagari* character recognition system. They have proposed structural feature based algorithm for recognition. Hanmandlu *et al*. (2007) have used membership functions of fuzzy sets for handwritten *Devanagari* script recognition. Pal *et al*. (2007b) have developed a modified classifier based scheme for offline handwritten numerals recognition of six widely used Indian scripts. They have extracted directional features for numeral recognition. They have obtained 99.6% recognition accuracy. Pal *et al*. (2007c) have reported a method for offline handwritten *Devanagari* character recognition. They have achieved a recognition accuracy of 94.2%. Kumar (2008) has brought in an artificial intelligence based technique for machine recognition of handwritten *Devanagari* script. He has used three levels of abstraction to describe this technique. Garg *et al*. (2010) have developed a line segmentation technique for handwritten *Hindi* text. Lajish and Kopparapu (2010) have described a technique for online

handwritten *Devanagari* script recognition. They have extracted fuzzy direction features for writer independent *Devanagari* character recognition.

Marathi is an Indo-Aryan language spoken in the Indian state of Maharashtra and neighbouring states. Ajmire and Warkhede (2010) have presented a technique based on invariant moments for isolated handwritten *Marathi* character recognition. The proposed technique is size independent. Shelke and Apte (2011) have presented a multi-stage handwritten character recognition system for *Marathi* script. They have achieved the recognition accuracy of 94.2% for testing data set with wavelet approximation features. They have also achieved 96.2% recognition accuracy for testing samples with modified wavelet features. Belhe *et al.* (2012) have presented a *Hindi* handwritten word recognition system. They have used HMM and tree classifier for recognition and obtained a recognition accuracy of 89% using 10,000 *Hindi* words.

2.2.3 *Gujarati*

Antani and Agnihotri (1999) are pioneers in attempting *Gujarati* printed text recognition. For experimental results, they have used dataset of scanned images of printed *Gujarati* texts collected from various internet sites. Dholakia *et al*. (2005) attempted to use wavelet features and *k*-NN classifier on the printed *Gujarati* text recognition system. They have achieved a recognition accuracy of 96.7% with *k*-NN classifier. Prasad *et al*. (2009) have furnished a technique called pattern matching for *Gujarati* script recognition. In this technique, they have identified a character by its shape.

2.2.4 *Gurmukhi*

Lehal and Singh (1999) have presented a hybrid classification scheme for printed *Gurmukhi* text recognition. Using this scheme, they have achieved a recognition accuracy of 91.6%. A post processor for *Gurmukhi* script has been proposed by Lehal *et al.* (2001). Based on the size and shape of a word, they split the *Punjabi* corpora into different partitions. The statistical information of *Punjabi* language syllable combination corpora look up and

holistic recognition of most commonly occurring words have been combined to design the post processor. Jindal *et al.* (2005) have proposed a solution for touching character segmentation of printed *Gurmukhi* script. Also, they have provided a very useful solution for overlapping lines segmentation in various Indian scripts (2007). They have proposed a technique for segmentation of degraded *Gurmukhi* script word into upper, middle and lower zones. They have also provided a complete recognition system for degraded printed *Gurmukhi* script documents. Sharma and Lehal (2006) have presented a technique for segmentation of isolated handwritten *Gurmukhi* words. They segmented the words in an iterative manner by focusing on presence of headline aspect ratio of characters and vertical and horizontal projection profiles. Sharma *et al*. (2008) have developed an online handwritten *Gurmukhi* script recognition system. They have used the elastic matching technique in which the character was recognized in two stages. In the first stage, they recognize the strokes and in the second stage, the character is formed on the basis of recognized strokes. Sharma *et al.* (2009) have expounded a method to rectify the recognition results of handwritten and machine printed *Gurmukhi* OCR systems. Sharma and Jhajj (2010) have extracted zoning features for handwritten *Gurmukhi* character recognition. They have employed two classifiers, namely, *k*-NN and SVM. They have achieved maximum recognition accuracy of 72.5% and 72.0%, respectively with *k*-NN and SVM.

2.2.5 *Kannada*

Kannada is one of the most widely used scripts of Southern India and is spoken by more than fifty million people in India. A little work has been done for handwritten *Kannada* text recognition. Ashwin and Sastry (2002) have presented a font and size independent OCR system for printed *Kannada* documents. They extracted features based on the foreground pixels in the radial and the angular directions. They achieved a maximum recognition accuracy of 94.9% using SVM classifier. Sharma *et al.* (2006a) have employed a quadratic classifier for offline handwritten *Kannada* numerals recognition. They have achieved maximum recognition accuracy of 98.5% using this technique. Kunte and Samuel (2007) have presented efficient printed *Kannada* text recognition system. They considered invariant moments and Zernike moments as features and Neural Network (NN) as classifier. They

obtained a recognition accuracy of 96.8% using 2,500 characters. Acharya *et al.* (2008) have come up with a handwritten *Kannada* numerals recognition system. They have used structural features and multilevel classifiers for recognition. Rajashekararadhya and Ranjan (2008) have evolved a technique based on zoning and distance metric features. They have utilized feed forward back propagation neural network and obtained recognition accuracy of about 98.0% for *Kannada* numerals. They have also achieved a recognition accuracy of 97.8% for *Kannada* numerals with zoning and distance metric features and SVM classifier (2009a). They have utilized Nearest Neighbour classifier for recognition and obtained 97.8% recognition rate for *Kannada* numerals (2009b). Rampalli and Ramakrishnan (2011) have presented an online handwritten *Kannada* character recognition system which works in combination with an offline handwriting recognition system. They improved the accuracy of online handwriting recognizer by 11% when its combination with offline handwriting recognition system is used. Venkatesh and Ramakrishnan (2011) have presented a technique for fast recognition of online handwritten *Kannada* characters. Using this technique, they obtained an average accuracy of 92.6% for *Kannada* characters. Ramakrishnan and Shashidhar (2013) have addressed the challenges in segmentation of online handwritten isolated *Kannada* words. They achieved 94.3% segmentation accuracy using attention feed-based segmentation technique.

2.2.6 *Malayalam*

Malayalam is one of the popular scripts of Southern India. It is the eighth most widely used script in India. Lajish (2007) has presented a system based on fuzzy zoning and normalized vector distance measures for recognition of offline handwritten *Malayalam* characters. He has also presented a method for offline handwritten segmented *Malayalam* character recognition (Lajish, 2008). John *et al.* (2007) have presented a method based on wavelet transform for offline handwritten *Malayalam* character recognition. Arora and Namboodiri (2010) have proposed a system for online handwritten *Malayalam* character recognition. They have used directional information based features and SVM classifier. Their system achieves a stroke level accuracy of 95.7%. Rahiman *et al.* (2010) have evolved an algorithm which accepts the scanned image of handwritten characters as input and produces

the editable *Malayalam* characters in a predefined format as output. Sreeraj and Idicula (2010) have presented a technique for online handwritten *Malayalam* character recognition. They have employed the *k*-NN classifier and achieved a recognition accuracy of 98.1%.

2.2.7 *Oriya*

Tripathy and Pal (2004) have segmented *Oriya* handwritten text using water reservoir based technique. Roy *et al.* (2005a) dealt with offline unconstrained handwritten *Oriya* numerals recognition. They have achieved a recognition accuracy of 90.4% using NN classifier with a rejection rate of about 1.84%. Bhowmik *et al.* (2006) have developed a novel HMM for handwritten *Oriya* numerals recognition. They have achieved a recognition accuracy of 95.9% and 90.6% for training and testing sets, respectively. Pal *et al.* (2007d) have put forth an offline handwritten *Oriya* script recognition system. They have extracted curvature features for recognition and achieved a recognition accuracy of 94.6% from handwritten *Oriya* samples.

2.2.8 *Tamil*

Aparna *et al.* (2004) have presented a system for online handwritten *Tamil* character recognition. They have used shape based features including dot, line terminal, bumps and cusp in their work. Deepu *et al*. (2004) have presented an online handwritten *Tamil* character recognition using PCA. Joshi *et al*. (2004a) have presented comparisons of elastic matching algorithms for online *Tamil* handwritten character recognition. They have also presented a *Tamil* handwriting recognition system using subspace and DTW based classifiers (Joshi *et el*., 2004b). In the subspace methodology the interactions between the features in the feature space are assumed to be linear. In DTW methodology, they investigated an elastic matching technique using dynamic programming principle. Prasanth *et al.* (2007) have described a character based elastic matching technique for online handwritten *Tamil* character recognition. Sundaram and Ramakrishnan (2008) have presented a technique based on Two Dimensional Principal Component Analysis (2*D*-PCA) for online *Tamil* character recognition. They have achieved a recognition accuracy of 81.1% for *Tamil* characters using

2*D*-PCA. Bharath and Madhvanath (2011) have used HMM for *Tamil* word recognition system. They have achieved a maximum recognition accuracy of 98.0%. Sundaram and Ramakrishnan (2013) have proposed script-dependent approach to segment online handwritten isolated *Tamil* words into its constituent symbols. They tested their proposed scheme on a set of 10, 000 isolated handwritten words. Sundaram and Ramakrishnan (2014) reduced the error rate of the *Tamil* symbol recognition system by reevaluate certain decisions of the SVM classifier.

2.2.9 *Telugu*

Prasanth *et al.* (2007) have used elastic matching technique for online handwritten *Telugu* character recognition. They have obtained a recognition accuracy of 90.6%. Pal *et al.* (2007b) have used direction information for *Telugu* numeral recognition. They have used a five-fold and obtained a recognition accuracy of 99.4% for *Telugu* numeral recognition. Arora and Namboodiri (2010) have proposed a system for online handwritten *Telugu* character recognition. They have achieved a stroke level accuracy of 95.1% for *Telugu* character recognition.

2.3 Algorithms used in this work at different stages of recognition system

A typical offline HCR system consists of various activities like digitization, pre-processing, segmentation, feature extraction, classification and post-processing as discussed in section 1.3. These activities and the techniques used in these activities, if any, are described, in brief, in the following sub-sections.

2.3.1 Digitization

All documents in this work are scanned at 300 dots per inch resolution, which is a widely accepted value. In the OCR system, optical scanners are used, which usually consist of a transport apparatus and a sensing device that convert light intensity into gray-levels. While proposing OCR systems, it is a general practice to change the multilevel image into a bi-level

image of black and white pixels. Through the scanning process, a digital image of the original paper document is captured, which is then input to the pre-processing phase of an offline HCR system.

2.3.2 Pre-processing

In this phase, the gray level character image is normalized into a window of size 100×100 using Nearest Neighbourhood Interpolation (NNI) algorithm. NNI algorithm is also known as point sampling algorithm. After normalization, we produce a bitmap image of the normalized image. Now, the bitmap image is converted into a thinned image using the parallel thinning algorithm proposed by Zhang and Suen (1984).

2.3.3 Segmentation

Segmentation is a very significant phase of an OCR system. Segmentation is a complicated task in a handwritten text recognition system. We have implemented two methods for text line segmentation, namely, projection profiles (Shapiro *et al.*, 1993) and strip based projection profiles (Arivazhagan *et al.*, 2007) in this work. For word segmentation, we have considered white space and pitch method. The white space and pitch method of detecting the horizontal white space between successive words in a line is an important concept for dividing the handwritten text line. This technique is not useful for touching and overlapping word segmentation. We have used the water reservoir based concept for touching characters segmentation (Pal and Dutta, 2003). A new technique, as discussed in Chapter 3, has also been proposed in this work for line segmentation.

2.3.4 Feature extraction

We have used zoning features, diagonal features, directional features, transition features, intersection and open end point features for offline handwritten *Gurmukhi* character recognition. We have also proposed efficient feature extraction techniques, namely, parabola curve fitting based features, power curve fitting based features, peak extent based features

and modified division points based features for offline handwritten *Gurmukhi* character recognition which have been discussed in Chapters 5 and 6.

2.3.4.1 Zoning based features [Rajashekararadhya and Ranjan, 2008]

In this technique, we divide the thinned image of a character into *n* (=100) number of equal sized zones as shown in Figure 2.1.

Z_1	Z_2	Z_3	Z_4	Z_5	Z_6	Z_7	Z_8	Z_9	Z_{10}
Z_{11}	Z_{12}	Z_{13}	Z_{14}	Z_{15}	Z_{16}	Z_{17}	Z_{18}	Z_{19}	Z_{20}
Z_{21}	Z_{22}	Z_{23}	Z_{24}	Z_{25}	Z_{26}	Z_{27}	Z_{28}	Z_{29}	Z_{30}
Z_{31}	Z_{32}	Z_{33}	Z_{34}	Z_{35}	Z_{36}	Z_{37}	Z_{38}	Z_{39}	Z_{40}
Z_{41}	Z_{42}	Z_{43}	Z_{44}	Z_{45}	Z_{46}	Z_{47}	Z_{48}	Z_{49}	Z_{50}
Z_{51}	Z_{52}	Z_{53}	Z_{54}	Z_{55}	Z_{56}	Z_{57}	Z_{58}	Z_{59}	Z_{60}
Z_{61}	Z_{62}	Z_{63}	Z_{64}	Z_{65}	Z_{66}	Z_{67}	Z_{68}	Z_{69}	Z_{70}
Z_{71}	Z_{72}	Z_{73}	Z_{74}	Z_{75}	Z_{76}	Z_{77}	Z_{78}	Z_{79}	Z_{80}
Z_{81}	Z_{82}	Z_{83}	Z_{84}	Z_{85}	Z_{86}	Z_{87}	Z_{88}	Z_{89}	Z_{90}
Z_{91}	Z_{92}	Z_{93}	Z_{94}	Z_{95}	Z_{96}	Z_{97}	Z_{98}	Z_{99}	Z_{100}

Figure 2.1: Zones of any input character

Now, the number of foreground pixels in each zone is calculated. These numbers p_1, p_2, …, p_n, obtained for all *n* zones, are normalized to [0, 1] resulting into a feature set of *n* elements.

2.3.4.2 Diagonal features [Pradeep *et al.*, 2011]

Diagonal features are hepful in achiving higher accuracy of the recognition system. Here also, the thinned image of a character is divided into *n* (=100) zones. Now, diagonal features are extracted from the pixels of each zone by moving along its diagonals as shown in Figures 2.2 (a) and 2.2 (b).

The steps that have been used to extract these features are:

Step I: Divide the thinned image into *n* (=100) number of zones, each of size 10×10 pixels.

Step II: Each zone has 19 diagonals; foreground pixels present along each diagonal are summed up in order to get a single sub-feature.

Step III: These 19 sub-feature values are averaged to form a single value which is then placed in the corresponding zone as its feature.

Step IV: Corresponding to the zones whose diagonals do not have a foreground pixel, the feature value is taken as zero.

These steps will again give a feature set with *n* elements.

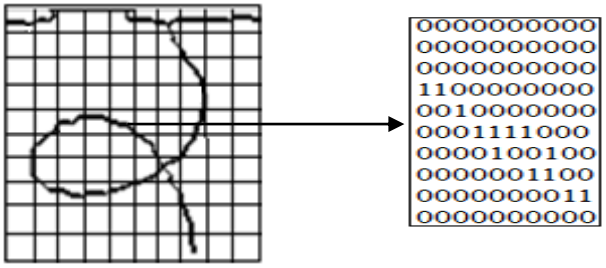


Figure 2.2 (a): Diagonal feature extraction

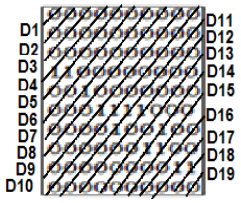


Figure 2.2 (b): Diagonals of Z_{45} zone

As shown in Figure 2.2 (a), we have divided the handwritten character image into *n* (=100) number of zones, each of size 10×10 pixels. Then, we have calculated foreground pixels in each zone by moving along its diagonal as shown in Figure 2.2 (b). These values of sub-features are averaged to form a single value and placed in the corresponding zone as its feature. For example, the zone Z_{45} given in Figure 2.2 (b) shall have a feature value of 0.6842.

2.3.4.3 Directional features [Bhattacharya *et* al., 2007]

In order to extract directional features, the thinned image of a character is divided into *n* (=100) zones. The features are then extracted using the starting (x_1, y_1) and ending (x_2, y_2) foreground pixels of each zone by calculating the slope between these points as shown in Figure 2.3.

Following steps are used to obtain directional features for a given character.

Step I: Divide the thinned image into *n* (=100) zones each of size 10×10 pixels.

Step II: Scan the bitmap of character image from left to right and top to bottom.

Step III: Find the positions of the starting foreground pixel (x_1, y_1) and the ending foreground pixel (x_2, y_2) in each zone and calculate the slope between these points using the formula, $\theta = tan^{-1}\frac{(y_2-y_1)}{(x_2-x_1)}$.

Step IV: For the zones with zero foreground pixels, the feature value is taken as zero.

Thus, a feature set of *n* elements will again be obtained for a given character.

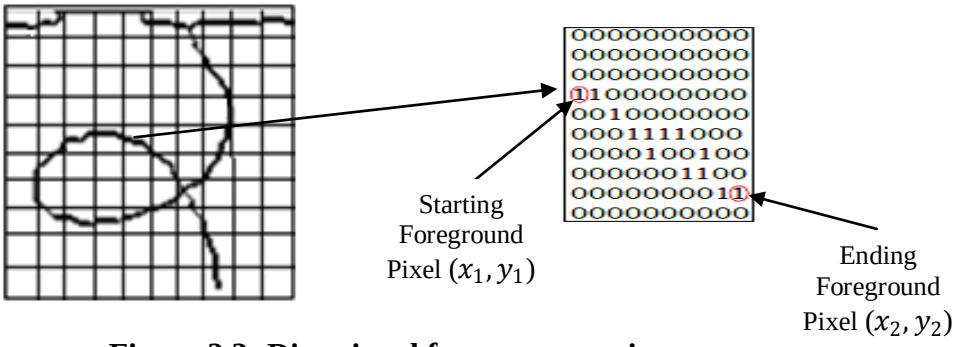


Figure 2.3: Directional feature extraction

As depicted in Figure 2.3, in zone Z_{45}, the position of the starting foreground pixel (x_1, y_1) is (1, 4) and the ending foreground pixel (x_2, y_2) is (10, 9). So, the feature value for this zone is $\tan^{-1}\frac{(9-4)}{(10-1)} = 0.5071$. If x_1 is approximately equal to x_2, then the value of feature has been taken as $\frac{\pi}{2}$. Similarly, we have calculated the feature values of other zones for a character.

2.3.4.4 Intersection and open end point features [Arora *et al.*, 2008]

We have also extracted the intersection and open end points for a character. An intersection point is the pixel that has more than one pixel in its neighbourhood and an open end point is the pixel that has only one pixel in its neighbourhood.

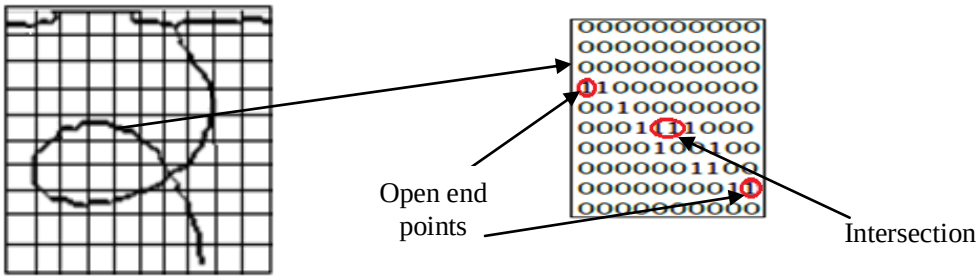


Figure 2.4: Intersection and open end point feature extraction

Following steps have been implemented for extracting these features.

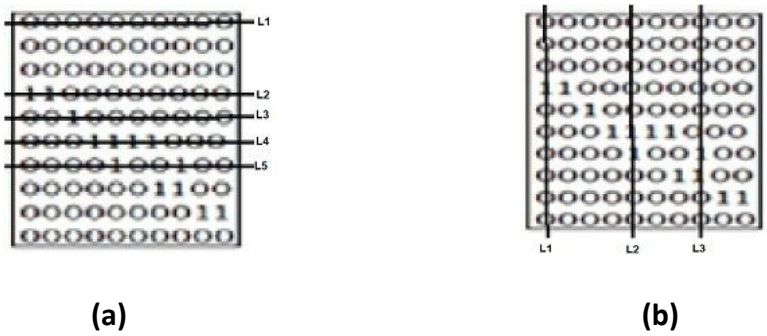
Step I: Divide the thinned image of a character into *n* (=100) zones, each of size 10×10 pixels (Figure 2.4).

Step II: Calculate the number of intersections and open end points for each zone.

This will give 2*n* features for a character image.

2.3.4.5 Transition features [Gader *et al.*, 1997]

Extraction of transition features is based on the calculation and location of transitions from background to foreground pixels and vice versa in the vertical and the horizontal directions. To calculate transition information, image is scanned from left to right and top to bottom. This process, as described below, will give 2*n* features for a character image.



(a) **(b)**

Figure 2.5: Transition feature extraction, (a) Transitions in horizontal direction, (b) Transitions in vertical direction.

As shown in Figure 2.1, we divide the handwritten character image into *n* (=100) number of zones, each of size 10×10 pixels. Then we calculate number of transitions in each zone. The zone depicted in Figure 2.5 (a), contains {0, 0, 0, 1, 2, 2, 4, 2, 2, 0} number of transitions in horizontal direction and as shown in Figure 2.5 (b), it contains {2, 2, 2, 2, 2, 2, 4, 2, 2, 2} number of transitions in vertical direction.

Following steps have been implemented for extracting these features.

Step I: Divide the thinned image of a character into *n* (=100) zones, each of size 10×10.

Step II: Calculate number of transitions in horizontal and in vertical directions for each zone.

2.3.5 Classification

Classification phase uses the features extracted in the previous phase for making class membership in the pattern recognition system. We have used the following classification methods in this research work.

2.3.5.1 NN classifier

In the NN classifier, Euclidean distances from the candidate vector to stored vector are computed. The Euclidean distance between a candidate vector and a stored vector is given by,

$$d = \sqrt{\sum_{k=1}^{N}(x_k - y_k)^2} \tag{2.1}$$

Here, N is the total number of features in feature set, x_k is the library stored feature vector and y_k is the candidate feature vector. The class of the library stored feature producing the smallest Euclidean distance, when compared with the candidate feature vector, is assigned to the input character.

2.3.5.2 SVM classifier

SVM is a very useful technique for data classification. The SVM is a learning machine which has been extensively applied in pattern recognition. SVMs are based on the statistical learning theory that uses supervised learning. In supervised learning, a machine is trained instead of being programmed to perform a given task on a number of inputs/outputs pairs. SVM classifier has also been considered with three different kernels, namely, linear kernel, polynomial kernel and RBF kernel in this work. Also, C-SVC type classifier in Lib-SVM tool has been used for classification purpose in this research work.

2.3.5.3 HMM classifier

Hidden Markov Model (HMM) was introduced in the 1960s and became widespread in the 1980s. HMMs are probabilistic pattern matching techniques that have the ability to absorb both the variability and similarities between stored and inputted feature values. A HMM is a finite state machine that can move to a next state at each time unit. With each move, an observed vector is generated. Probabilities in HMM are calculated utilizing observation vector extracted from samples of handwritten *Gurmukhi* characters. Recognition of unknown character is based on the probability that an unknown character is generated by HMM.

2.3.5.4 Bayesian classifier

The Bayesian classifier is also based on statistical approach that allows designing of the optimal classifier if complete statistical model is known. In this classifier, a character is assigned to the class for which it has the highest probability conditioned on X, where X is the test feature vector. This probability is given by:

$$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)} \tag{2.2}$$

2.3.5.5 MLP classifier

Multi-Layer Perceptron (MLP) has also been used in the present work for classification. Back Propagation (BP) learning algorithm with learning rate (γ) = 0.3 and momentum term (α) = 0.2 has been used in this work for training of MLP based classifiers. *Weka* tool has been used for MLP based classification purpose in this work.

2.4 Recognition accuracy achieved for different scripts

This section presents a brief report on the recognition accuracies achieved by researchers for numerals and character recognition. We have presented their results in Table 2.1 for numerals, Table 2.2 for non-Indian scripts and in Table 2.3 for Indian scripts. As depicted in Table 2.1, one may note that a recognition accuracy of 99.6% has been achieved for handwritten numerals by Pal *et al*. (2008). In Table 2.2, the results of non-Indian scripts are presented. As shown in this table, one may note that a recognition accuracy of 99.4%, 99.9%, 92.8% and 99.2%, has been achieved for *Arabic*, *French*, *Japanese* and *Roman* scripts, respectively.

Table 2.1: Recognition results of handwritten numerals

Author	Test data size	Feature extraction technique	Classifier	Accuracy
Bhattacharya and Chaudhuri (2003)	5,000	Wavelet	MLP	97.2%
Bhattacharya *et al*. (2004)	5,000	Wavelet	MLP	98.0%
Roy *et al*. (2004b)	12,410	Structural, Topological	NN	94.2%
Roy *et al*. (2005a)	3,850	Directional	Quadratic	94.8%
Bhowmik *et al.* (2006)	5,970	Strokes	HMM	95.9%
Kunte and Samuel (2006)	11,500	Wavelet	MLP	92.3%
Pal *et al*. (2006b)	12,000	Water reservoir	Binary tree	92.8%
Rajput and Hangrage (2007)	1,250	Image fusion	NN	91.0%
Pal *et al.* (2007b)	2,690	Directional	Modified quadratic	98.5%
Pal *et al*. (2008)	5,638	Curvature	Modified quadratic	99.6%
Lu *et al*. (2008)	16,000	Directional and Density	SOM	97.3%
Desai (2010)	3,260	Profiles	FFNN	81.7%
Purkait and Chanda (2010)	23,392	Morphological	MLP	97.8%

Table 2.2: Recognition results of handwritten non-Indian scripts

Author	Script	Data set	Number of classes	Feature extraction technique	Classifier	Accuracy
Alaei *et al.* (2009)	*Arabic*	20,000	Not mentioned	Chain code direction	SVM	99.4%
Alaei *et al.* (2010b)	*Arabic*	20,000	Not mentioned	Modified chain code direction	SVM	98.1%
Tran *et al.* (2010)	*French*	400	Not mentioned	Statistical and structural	SVM	99.9%
Zhu *et al.* (2010)	*Japanese*	35,686	790	Geometric features	SVM	92.8%
Pal *et al.* (2010)	*Roman*	11,875	Not mentioned	Chain code	MQDF	99.0%
Park *et al.* (2000)	*Roman*	10, 000	Not mentioned	Gradient and moments based projections	Hierarchical	96.0%
Roy *et al.* (2010)	*Roman*	5,000	Not mentioned	Fractal dimension, Topological	MLP, SVM, *k*-NN, MQDF	99.2%

In Table 2.3, the results on Indian scripts have been presented. It can be noticed that a lot of work has been done on *Bangla*, *Devanagari* and *Kannada* scripts. Some work has also been done to recognize the *Gurmukhi, Malayalam, Oriya and Tamil* scripts as given in this table. As depicted in Table 2.3, for *Bangla* script, maximum recognition accuracy of 89.2% has been achieved by Bhowmik *et al*. (2004). For *Devanagari* script, maximum recognition accuracy of 99.0% has been achieved by Pal *et al*. (2009a). They have used directional features and MQDF classifier for recognition. Kunte and Samuel (2007) have achieved a maximum recognition accuracy of 96.8% for *Kannada* characters. They have tested their technique with 1,000 samples of 50-class problem. For all classes of *Kannada* script, maximum recognition accuracy of 92.6% has been achieved by Venkatesh and Ramakrishnan (2011). They have considered 26,926 samples for testing data set. Arora and Namboodiri (2010) have been a recognition accuracy of 95.8% for *Malayalam* character recognition. They have tested their technique with 7,348 samples of *Malayalam* characters. Joshi *et al*. (2004a) have achieved a maximum recognition accuracy of 91.5% for *Tamil* character recognition. They have considered 4,860 samples of 156 classes for testing data set.

For offline handwritten *Gurmukhi* script, a recognition accuracy of 72.0% has been achieved by Sharma and Jhajj (2010). Nonetheless till now, there is no complete recognition system available for recognition of offline handwritten *Gurmukhi* script. As such, there is a need for an offline handwritten *Gurmukhi* script recognition system that can help people to convert the handwritten *Gurmukhi* text to a computer processable format.

Table 2.3: Recognition results of handwritten Indian scripts

Author	Script	Test data size	Number of classes	Feature extraction technique	Classifier	Accuracy
Bhowmik *et al.* (2004)	*Bangla*	25,000	50	Stroke	MLP	84.3%
Bhowmik *et al.* (2009)	*Bangla*	27,000	45	Wavelet	SVM	89.2%
Belhe *et al.* (2012)	*Devanagari*	10,000	140	Histogram of oriented gradients	HMM and Tree	89.0%
Hanmandlu *et al.* (2007)	*Devanagari*	4,750	Not mentioned	Normalized distance	Fuzzy set	90.7%
Joshi *et al.* (2005)	*Devanagari*	1,487	441	Gaussian low pass filters	Feature based	94.5%
Pal *et al.* (2009a)	*Devanagari*	36,172	Not mentioned	Dimensional	MQDF	99.0%
Sharma and Jhajj (2010)	*Gurmukhi*	5,125	34	Zoning	SVM	73.0%
Rampalli and Ramakrishnan (2011)	*Kannada*	6,195	295	Directional distance distribution, transitions, projection profiles	SVM	89.7%
Venkatesh and Ramakrishnan (2011)	*Kannada*	26,926	295	Quantized slope, quartile features	DTW	92.6%
John *et al.*(2007)	*Malayalam*	4,950	33	Wavelet transform	MLP	73.8%
Lajish (2007)	*Malayalam*	15,752	44	Fuzzy zoning	Class modular NN	78.9%

Raju (2008)	*Malayalam*	12,800	33	Wavelet	MLP	81.3%
Moni and Raju (2011)	*Malayalam*	19,800	44	Gradient	MQDF	95.4%
Chacko and Anto (2010a)	*Malayalam*	Not mentioned	33	Structural	MLP	90.2%
Chacko and Anto (2010b)	*Malayalam*	3,000	30	Zonal	MLP	95.2%
Arora and Namboodiri (2010)	*Malayalam*	7,348	90	Strokes	HMM, DTW	95.8%
Pal *et al.* (2007d)	*Oriya*	5,638	51	curvature features	Quadratic	94.6%
Deepu *et al.* (2004)	*Oriya*	21,840	156	Gaussian low pass filters	PCA and NN	95.3%
Joshi *et al.*(2004a)	*Tamil*	4,860	156	*x-y* co-ordinates	DTW	91.5%
Hewavitharana and Fernando (2002)	*Tamil*	800	26	Pixel density	Statistical	80.0%
Shanthi and Duraiswamy (2010)	*Tamil*	6048	34	Pixel density	SVM	82.0%
Sundaram and Ramakrishnan (2008)	*Tamil*	1,560	156	2-D PCA global features	Modified Mahalanobis distance measure	83.4%
Sastry *et al.* (2010)	*Telugu*	Not mentioned	Not mentioned	3D	Decision tree	93.1%

2.5 Recognition accuracy achieved for complete set of *aksharas*

There are very few reports are available on recognition of complete set of *aksharas*. These reports mainly deal with the recognition of *Kannada* and *Tamil* scripts. Venkatesh and Ramakrishnan (2011) have presented a technique for fast recognition of online handwritten *Kannada* characters. Using this technique, they obtained a recognition accuracy of 92.6% for complete set of *aksharas* of *Kannada* script. Sundaram and Ramakrishnan (2013) have proposed script-dependent approach to segment online handwritten isolated *Tamil* words into its constituent symbols. They tested their proposed scheme on a set of 10, 000 isolated handwritten words. Ramakrishnan and Shashidhar (2013) have addressed the challenges in

segmentation of online handwritten isolated *Kannada* words. They achieved 94.3% segmentation accuracy using attention feed-based segmentation technique. Sundaram and Ramakrishnan (2014) reduced the error rate of the *Tamil* symbol recognition system by reevaluate certain decisions of the SVM classifier.

2.6 Chapter summary

In this chapter, we have surveyed the numeral and character recognition work that has been done on non-Indian and Indian scripts. We have analyzed the work done for pre-processing, segmentation, feature extraction and classification for various Indian scripts, *i.e.*, *Bangla*, *Devanagari*, *Gujarati*, *Gurmukhi*, *Kannada*, *Malayalam*, *Oriya*, *Tamil* and *Telugu*. Also, we have presented the work done for recognition of various phases of non-Indian scripts, *i.e.*, *Arabic*, *French*, *Japanese* and *Roman*. We have also discussed in detail various feature extraction techniques used in this thesis for extracting the features of the characters and classifiers used in this thesis for character recognition. Finally, in this chapter, we have presented recognition accuracies achieved for numerals, non-Indian and Indian scripts.

Chapter 3

Data Collection, Digitization, Pre-processing and Segmentation

==========

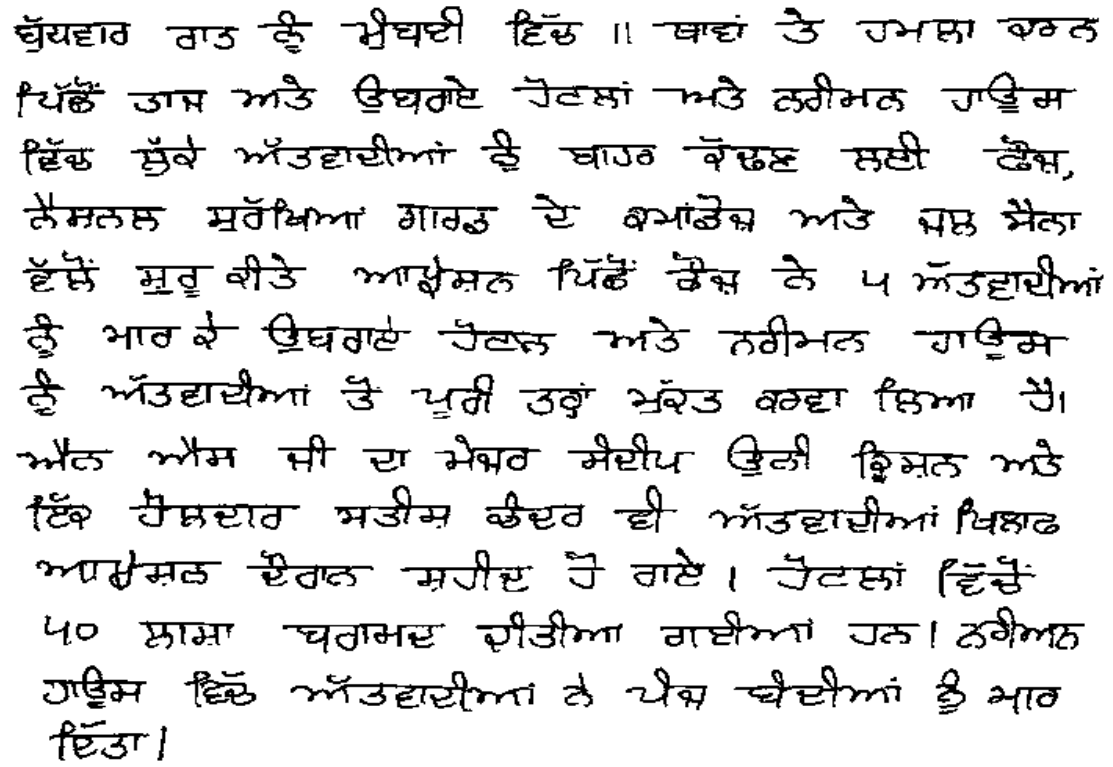
Data collection, digitization, pre-processing and segmentation are preliminary phases of an offline Handwritten Character Recognition (HCR) system. Subsequent sections explain the work done on data collection of offline handwritten *Gurmukhi* script documents, their digitization, pre-processing and segmentation in this thesis. Section 3.1 focuses on data collection; section 3.2 includes digitization process; section 3.3 discusses pre-processing phase and section 3.4 consists of work done in the segmentation phase of the offline HCR system developed in this study.

3.1 Data collection

In this study, we have collected 300 samples of handwritten documents written in *Gurmukhi* script. These samples have been taken for three different categories. Category 1 consists of one hundred samples of offline handwritten *Gurmukhi* script documents where each *Gurmukhi* script document is written by a single writer. Category 2 contains one hundred samples where each *Gurmukhi* script document is written ten times by ten different writers. In category 3, one *Gurmukhi* script document is written by one hundred different writers. As such, this category also consists of one hundred samples as shown in Table 3.1. As such, a sufficiently large database has been built for offline handwritten *Gurmukhi* script documents. These samples of offline handwritten *Gurmukhi* script documents of different writers were collected from various organizations, offices and public places. A sample of an offline handwritten *Gurmukhi* script document is enunciated in Figure 3.1.

Table 3.1: Metadata for data collected

Category	Number of writers	Number of documents written by each writer	Number of samples
Cat-1	1	100	100
Cat-2	10	10	100
Cat-3	100	1	100



ਬੁੱਧਵਾਰ ਰਾਤ ਨੂੰ ਮੁੱਢਲੀ ਵਿੱਚ ।। ਥਾਵਾਂ ਤੇ ਹਮਲਾ ਕਰਨ ਪਿੱਛੋਂ ਤਾਜ਼ ਅਤੇ ਉਬਰਾਏ ਹੋਟਲਾਂ ਅਤੇ ਨਰੀਮਨ ਹਾਊਸ ਵਿੱਚ ਚੁੱਕੇ ਅੱਤਵਾਦੀਆਂ ਨੂੰ ਬਾਹਰ ਕੱਢਣ ਲਈ ਫੌਜ, ਨੈਸ਼ਨਲ ਸੁਰੱਖਿਆ ਗਾਰਡ ਦੇ ਕਮਾਂਡੋ ਅਤੇ ਮੁਬ ਮੈਨਾ ਵੱਲੋਂ ਸੁਰੂ ਕੀਤੇ ਆਪ੍ਰੇਸ਼ਨ ਪਿੱਛੋਂ ਫੌਜ ਨੇ ੫ ਅੱਤਵਾਦੀਆਂ ਨੂੰ ਮਾਰ ਕੇ ਉਬਰਾਏ ਹੋਟਲ ਅਤੇ ਨਰੀਮਨ ਹਾਊਸ ਨੂੰ ਅੱਤਵਾਦੀਆਂ ਤੋਂ ਪੂਰੀ ਤਰ੍ਹਾਂ ਮੁੱਕਤ ਕਰਵਾ ਲਿਆ ਹੈ। ਐਨ ਐਸ ਜੀ ਦਾ ਮੇਜਰ ਸੰਦੀਪ ਉਨੀ ਕ੍ਰਿਸ਼ਨਨ ਅਤੇ ਇੱਕ ਹੌਲਦਾਰ ਸਤੀਸ਼ ਬੰਦਰ ਵੀ ਅੱਤਵਾਦੀਆਂ ਖਿਲਾਫ ਆਪ੍ਰੇਸ਼ਨ ਦੌਰਾਨ ਸ਼ਹੀਦ ਹੋ ਗਏ। ਹੋਟਲਾਂ ਵਿੱਚੋਂ ੫੦ ਲਾਸ਼ਾ ਬਰਾਮਦ ਕੀਤੀਆ ਗਈਆਂ ਹਨ। ਨਰੀਮਨ ਹਾਊਸ ਵਿੱਚੋ ਅੱਤਵਾਦੀਆਂ ਨੇ ਪੰਜ ਬੰਦੀਆਂ ਨੂੰ ਮਾਰ ਦਿੱਤਾ।

Figure 3.1: Offline handwritten *Gurmukhi* script document

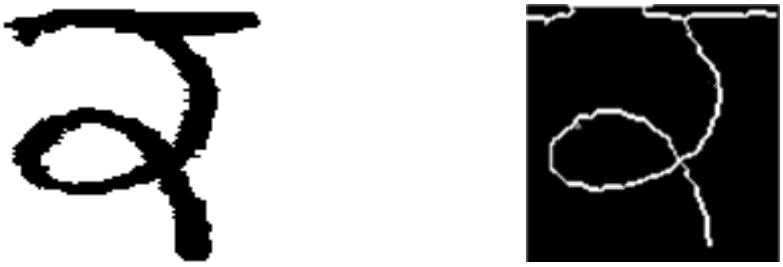
3.2 Digitization

Digitization is the process of converting the paper based handwritten document into electronic form. All three hundred documents, as illustrated in the above section, are scanned

at 300 dots per inch resolution. Digitization produces the digital image, which is fed to the pre-processing phase of the offline HCR system developed in this thesis.

3.3 Pre-processing

In this phase, the size of character image is normalized using Nearest Neighbourhood Interpolation (NNI) technique. After normalization, we construct a bitmap image of the normalized image. Now, the bitmap image is changed into a thinned image. This process of pre-processing is shown in Figure 3.2 for *Gurmukhi* character ਕ.



(a) **(b)**

Figure 3.2 A sample handwritten *Gurmukhi* character (ਕ): (a) Digitized image, (b) Thinned image

3.4 Segmentation

Segmentation is an important step for a character recognition system. *Gurmukhi* script documents can be segmented into paragraphs, lines, words and characters. Segmentation is one of the challenging tasks in a handwritten text recognition system. Section 3.4.1 concentrates on line segmentation, section 3.4.2 includes word segmentation, section 3.4.3 discusses zone segmentation and section 3.4.4 presents isolated and touching character segmentation that has been carried out during this study.

3.4.1 Line segmentation

Line segmentation is the initial stage of segmentation phase in a character recognition system. Line segmentation is a complex task and it becomes even more challenging when one needs to segment lines in a skewed offline handwritten document. Improper line segmentation decreases the recognition accuracy considerably. There are a number of issues in segmenting of handwritten documents into lines. One of the issues is different styles of writing a document. Other issues include skewed lines, curvilinear lines, fluctuating lines, touching lines and overlapping lines. Shapiro *et al.* (1993) were able to find the skew angle by using the Hough transformation. To increase the strength of the histogram, they used black run length smearing technique in the horizontal direction. Sulem and Faure (1994) have been pioneer in developing an approach which is based on the perceptual grouping of all the connected components which constitute black pixels. Iterative construction of text lines is made possible by grouping the neighbouring components that are mutually connected in accordance with preconceived perceptual criteria such as similarity, continuity, and proximity. In this way, it is possible to combine the local constraints, which impart neighbouring components, with global quality measures. These methods are, however, not very useful for segmentation of lines in handwritten documents.

A few other techniques have also been offered for the problem of line segmentation. Some of them are based on linear programming (Yanikoglu and Sandon, 1998), fuzzy run length (Shi and Govindaraju, 2004), adaptive local connectivity map (Shi *et al.*, 2005), level set (Li *et al.*, 2006), Kalman filter (Lemaitre and Camillerapp, 2006), local neighbourhood of word (Basu *et al.*, 2007) *etc*. Gatos *et al.* (2007) have presented their handwriting segmentation results in ICDAR. They have presented various methods including ILSP-LWSeg, PARC, DUTH-ARLSA, BESUS and UoA-HT for segmentation of handwritten text. We have also proposed a technique for line segmentation of offline handwritten *Gurmukhi* documents based on a mixture of the smearing technique and the contour tracing technique.

3.4.1.1 Proposed technique for line segmentation

In this section, we have proposed a technique for line segmentation in offline handwritten *Gurmukhi* script documents. In this technique, we smear consecutive black pixels in the horizontal direction. Then we analyze the distance between the white spaces. If this distance is within a certain permissible threshold limit, then the white space is filled with black pixels. Once this is achieved, the boundaries of the components which are connected to each other within the image that has been smeared define the text lines.

We have applied this technique to offline handwritten *Gurmukhi* script document as shown in Figure 3.3(a). The smeared image and the processed document are shown in Figures 3.3(b) and 3.3(c), respectively. This figure shows that the proposed technique has successfully segmented the lines in this offline handwritten *Gurmukhi* script document. Table 3.2 depicts the accuracy of proposed line segmentation technique. The average accuracy achieved for line segmentation for documents is 98.4%, whereas the average accuracy achieved for all lines is 98.3%. We have calculated the average accuracy of line segmentation for documents and lines by:

Let, $ad(i) =$ accuracy of doc(i)

$$\text{Average accuracy for documents} = \frac{1}{N}\sum_{i=1}^{N} ad(i) \tag{3.1}$$

Let, $nol(i) =$ Number of lines in doc(i)

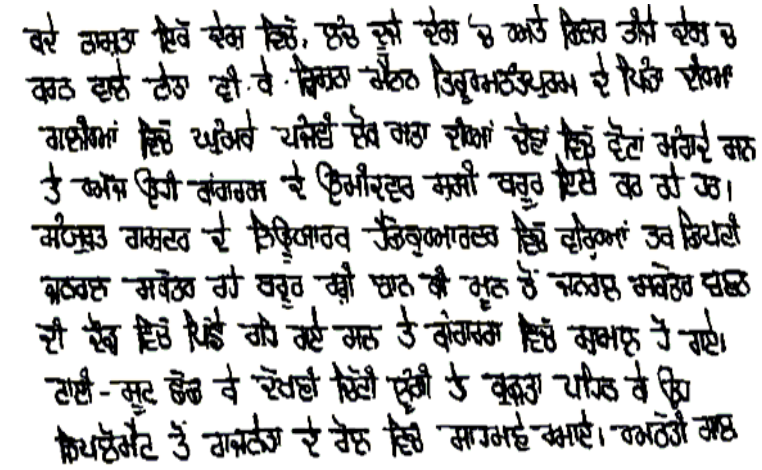
$$\text{Number of lines in all documents } (M) = \sum_{i=1}^{N} nol(i) \tag{3.2}$$

Average accuracy for lines *(A)* =

$$\frac{1}{M}\sum_{i=1}^{N}(nol(i) \times ad(i)) \tag{3.3}$$

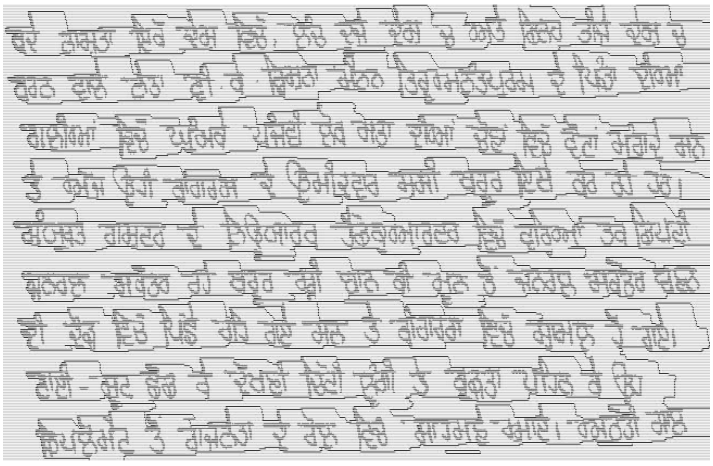
Here, *N* is the total number of documents.

It has been seen that this technique is more effective than projection profiles based technique for handwritten text line segmentation, when the lines are skewed or curved or the space between lines is not uniform. However, we are not able to segment touching lines or overlapping lines very efficiently with this proposed method of line segmentation.



(a)

(b)



(c)

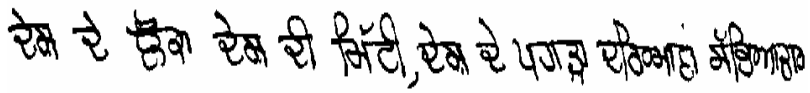
Figure 3.3 Proposed smearing technique for line segmentation: (a) Handwritten *Gurmukhi* script document, (b) Smeared image, (c) Processed document

Table 3.2: Line segmentation accuracy based on proposed technique

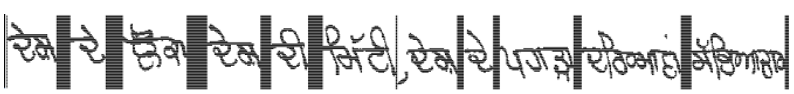
Document	Number of lines	Accuracy
doc(1)	17	100%
doc(2)	17	100%
doc(3)	18	100%
doc(4)	18	94.4%
doc(5)	14	100%
doc(6)	19	94.7%
doc(7)	19	100%
doc(8)	13	100%
doc(9)	18	100%
doc(10)	19	94.7%

3.4.2 Word segmentation

We have used the white space and pitch method for word segmentation in offline handwritten *Gurmukhi* text. The white space and pitch method of detecting the horizontal white space between successive words in a line is a widely used concept for dividing handwritten text line into words. We should not consider this technique for cursive handwriting word segmentation. Figures 3.4(a) and 3.4(b) show a handwritten text line before and after word segmentation, respectively. Word segmentation accuracies from ten example documents considered in this work have been depicted in Table 3.3. We have achieved a word segmentation accuracy of 97.9% for offline handwritten *Gurmukhi* text in this work.



(a)



(b)

Figure 3.4 Word segmentation: (a) *Gurmukhi* text line, (b) Processed *Gurmukhi* text line

Table 3.3: Word Segmentation Accuracy

Document	Number of Words	Accuracy
doc(1)	168	94.0%
doc(2)	180	95.0%
doc(3)	182	91.8%
doc(4)	162	96.9%
doc(5)	156	97.4%
doc(6)	191	97.9%
doc(7)	177	97.2%
doc(8)	172	97.1%
doc(9)	176	97.2%
doc(10)	180	96.7%

3.4.3 Zone segmentation

A line of *Gurmukhi* text can be partitioned into three horizontal zones, namely, upper zone, middle zone and lower zone. Consonants generally occupy the middle zone. The upper zone represents the region above the headline, while the middle zone represents the area just below the headline and above the lower zone. The lower zone is the lowest part which contains some vowels. In the process of *Gurmukhi* script recognition, one needs to find the headline, and the base line, to define the upper, lower, and the middle zones in order to have an efficient recognition system.

3.4.4 Character segmentation

Character segmentation is also a challenging task in an offline handwritten *Gurmukhi* character recognition system. This problem becomes more complex when characters are touching. In this work, we have applied water reservoir based technique (Pal *et al*., 2003) for identification and segmentation of touching characters in offline handwritten *Gurmukhi* words. Touching characters are segmented, based on reservoir base area points. We could

achieve 93.5% accuracy for character segmentation with this method. This section is further divided into three sub-sections. Section 3.4.4.1 concentrates on types of characters in *Gurmukhi* text that are to be tackled while segmenting the characters. Section 3.4.4.2 discusses the segmentation of isolated characters using vertical projection profiles and also the segmentation of touching characters using water reservoir method.

3.4.4.1 Different types of characters

a. Isolated characters

When characters do not touch each other, they are classified as isolated characters. Character segmentation is a straightforward process whenever characters are well spaced as shown in Figure 3.5.

Figure 3.5: A sample *Gurmukhi* word with well-spaced characters

b. Touching characters

In a handwritten *Gurmukhi* script document, touching characters are present frequently. Segmentation of such characters is a complex problem. An example of touching characters is given in Figure 3.6.

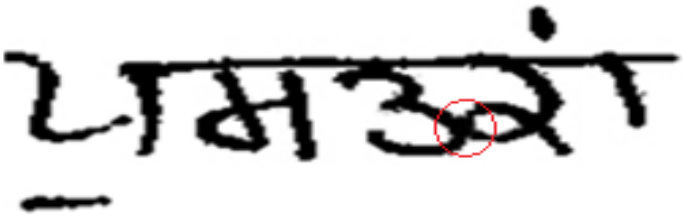


Figure 3.6: A sample *Gurmukhi* word with touching characters

As shown by the red circle in Figure 3.6, the adjacent characters ਤ and ਕ touch each other in the given word.

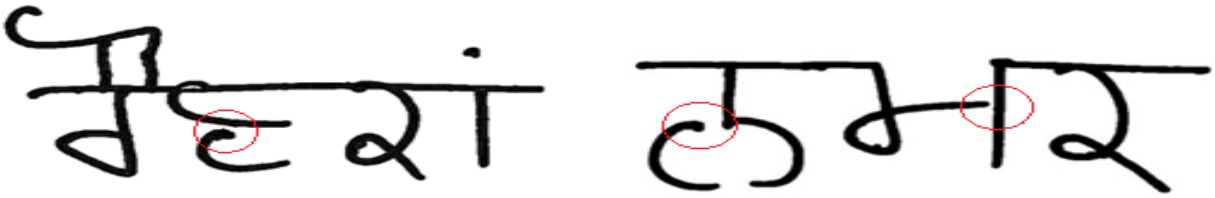
c. Overlapping characters

In offline handwritten *Gurmukhi* script documents, the characters can overlap each other as shown in Figure 3.7. As such, vertical/horizontal projections of these characters shall also be overlapping each other.

Figure 3.7: *Gurmukhi* word with overlapping characters

d. Broken characters

In offline handwritten *Gurmukhi* documents, some portion of the characters in the text may be missing as shown in Figure 3.8. Figure 3.8(a) contains an example case of horizontally broken characters and Figure 3.8(b) contains an example of vertical broken characters. It has been seen that most of the times, each broken character will have an aspect ratio less than that of a single isolated character, making their recognition a difficult task.



(a) **(b)**

Figure 3.8 Broken characters: (a) Horizontally broken characters, (b) Vertically broken characters

3.4.4.2 Segmentation of isolated and touching characters

Segmentation of offline handwritten *Gurmukhi* words into characters is a challenging task primarily because of structural properties of *Gurmukhi* script and various writing styles. We have tested the performance of different algorithms for character segmentation of

collected offline handwritten *Gurmukhi* script documents. In *Gurmukhi* script, most of the characters contain a horizontal line at the upper end of the middle zone which is called the headline. The headline helps in the recognition of script line positions and character segmentation. Segmentation of individual characters in offline handwritten *Gurmukhi* script recognition is a straightforward process when characters are well spaced as shown in Figure 3.9.

(a)

(b)

Figure 3.9 *Gurmukhi* word (ਮਨੀਸ਼): (a) With well-spaced characters, (b) Processed word

The processed word is the outcome of the segmentation process. Segmentation process extracts constituent images from a *Gurmukhi* word and performs the following tasks:

(i) It finds the headline. This is accomplished by finding the maximum number of black pixels in a row.

(ii) Headline is now removed.

(iii) Now, sub-images that are vertically separated from their neighbours are extracted. These sub-images may contain more than one connected component.

It will be done into two stages, identification of touching characters and segmentation of touching characters. To identify the touching characters by measure the width of each character, then the average width of character is found. If, any character, whose width is more than 150% of average width of each character, it is considered as touching characters.

In a handwritten *Gurmukhi* document, it is highly probable that the characters can touch each other. Separation of such touching characters is a complex problem, as explained in Figures 3.10(a) and 3.10(b). We have used water reservoir based method for touching character segmentation (Pal *et al.*, 2003).

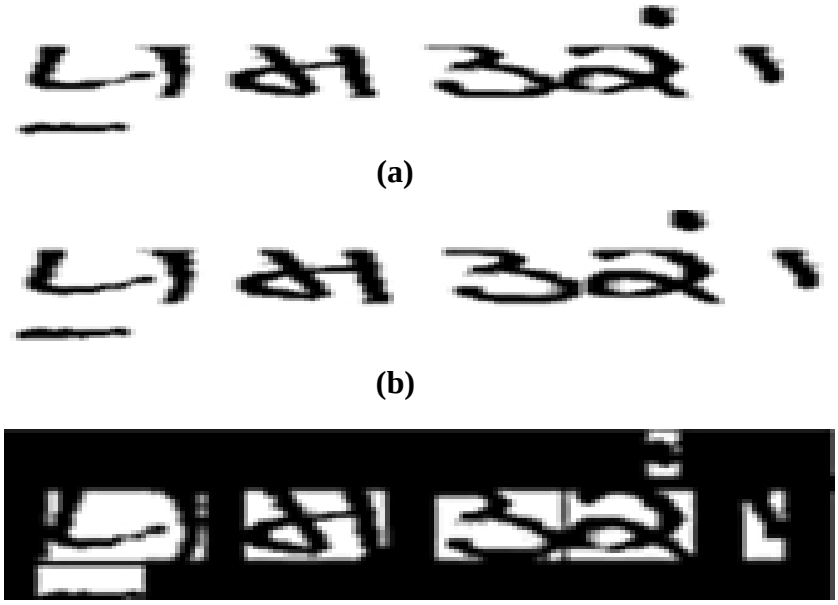
(a) **(b)**

Figure 3.10 *Gurmukhi* word (ਪੁਸਤਕਾਂ): (a) Touching characters, (b) Vertical projection profiles

In this method, the headline of a character is removed. Thereafter, water is poured on top of the character. The water is stored in reservoirs, which are actually the cavity regions of the characters whose headlines have been removed. Figure 3.11 illustrates this approach. Those reservoirs which get formed in the 'cavity regions' of the character when water is poured from the top are called "top reservoirs". However, all the reservoirs which get formed when water is poured in this manner cannot be considered for processing. For a reservoir to be considered for processing, it has to be of a height that is greater than a specified threshold. Normally, the specified threshold value of a 'top reservoir' is 1/10 of the height of the character.

Figure 3.11: A reservoir obtained from water flow from the top marked by dots

As illustrated in Figure 3.10 (a), in the given word, we can clearly see that the two characters ਤ and ਕ are touching each other. Consequently, we see that vertical projection profiles of these characters are also touching each other. To segment the characters which touch each other in such a way, we have used this method. In this technique, the first step is to identify the characters which are well isolated and the characters which touch each other in a given word. Once this is done, the characters that touch each other are segmented in accordance with the reservoir base area. This technique has been illustrated in Figures 3.12(a), 3.12(b) and 3.12(c).



(a)

(b)

(c)

Figure 3.12 Offline handwritten *Gurmukhi* word: (a) Without headline, (b) Touching character segmentation, (c) Complete segmentation of word

The horizontal and vertical projection profile techniques, along with water reservoir method have been applied on all the handwritten *Gurmukhi* documents which have been collected in this study for three different categories. Category wise results of character segmentation accuracy are given in Tables 3.4-3.6.

Table 3.4: Character segmentation accuracy of Cat-1 documents

Document	Total number of characters	Segmented characters	Accuracy
doc(11)	528	486	92.1%
doc(12)	482	440	91.3%
doc(13)	477	451	94.5%
doc(14)	572	526	91.9%
doc(15)	427	406	95.5%
doc(16)	458	447	97.7%
doc(17)	464	438	94.5%
doc(18)	430	401	93.4%
doc(19)	448	413	92.2%
doc(20)	387	374	96.7%
Average accuracy			93.9%

Table 3.5: Character segmentation accuracy of Cat-2 documents

Document	Total number of characters	Segmented characters	Accuracy
doc(21)	497	467	93.9%
doc(22)	418	385	92.2%
doc(23)	436	399	91.4%
doc(24)	528	489	92.6%
doc(25)	412	388	94.2%
doc(26)	429	417	97.5%
doc(27)	452	431	95.4%
doc(28)	424	392	92.7%
doc(29)	429	405	94.4%
doc(30)	446	429	96.1%
Average accuracy			94.0%

Table 3.6: Character segmentation accuracy of Cat-3 documents

Document	Total number of characters	Segmented characters	Accuracy
doc(31)	485	431	88.9%
doc(32)	427	389	91.3%
doc(33)	408	368	90.2%
doc(34)	410	370	90.4%
doc(35)	368	343	93.3%
doc(36)	370	362	97.9%
doc(37)	382	349	91.4%
doc(38)	355	327	92.3%
doc(39)	375	344	91.9%
doc(40)	340	314	97.6%
Average Accuracy			92.5%

As such, we could achieve an average accuracy of 93.5% for the segmentation of isolated and touching characters in this work.

3.5 Chapter summary

In this chapter, we have discussed the data collection, digitization, pre-processing and segmentation phases of offline handwritten *Gurmukhi* character recognition system that has been developed in this thesis. The data has been collected in three different categories. Each category consists of one hundred documents. In this chapter we have proposed a novel technique based on the combination of smearing technique and contour tracing technique for line segmentation. For word segmentation and isolated character segmentation, white space and pitch method has been used, for segmentation of touching characters water reservoir based method has been used. We achieved an accuracy of 98.4%, 97.9% and 93.5% for line, word and character segmentation, respectively.

Chapter 4
A Framework for Grading of Writers

==========

This chapter presents a framework for grading the writers based on their handwriting. This process of grading shall be helpful in organizing handwriting competitions and then deciding the winners on the basis of an automated process. Grading of writers based on their handwriting is an intricate task owing to a variety of writing styles of different individuals. In this chapter, we have attempted to grade the writers based on offline handwritten *Gurmukhi* characters written by them. Selecting the set of features is an important task for implementing a handwriting grading system. In this work, the features used for classification are based on zoning that has shown the capability of grading the writers. Further, samples of offline handwritten *Gurmukhi* characters, from one hundred different writers, have been considered in this work. In order to establish the correctness of our approach, we have also considered these characters, taken from five *Gurmukhi* fonts. We have used zoning; diagonal; directional; intersection and open end points feature extraction techniques in order to find the feature sets and have used Hidden Markov Model (HMM) and Bayesian classifiers for obtaining a classification score. This chapter is divided into three sections. Section 4.1 introduces the handwriting grading system, section 4.2 presents the experimental results of handwriting grading system, based on HMM and Bayesian classifiers. Finally, section 4.3 presents the conclusion of this chapter.

4.1 Handwriting grading system

A handwriting grading system consists of the activities, namely, digitization, pre-processing, segmentation, features extraction, classification and final grading based on the

classification score, as shown in Figure 4.1. This handwriting grading system can be used to grade the participants in a handwriting competition and can also be used, with suitable modifications, for signature verification. The activities of such a system have a close relationship with characters recognition system.

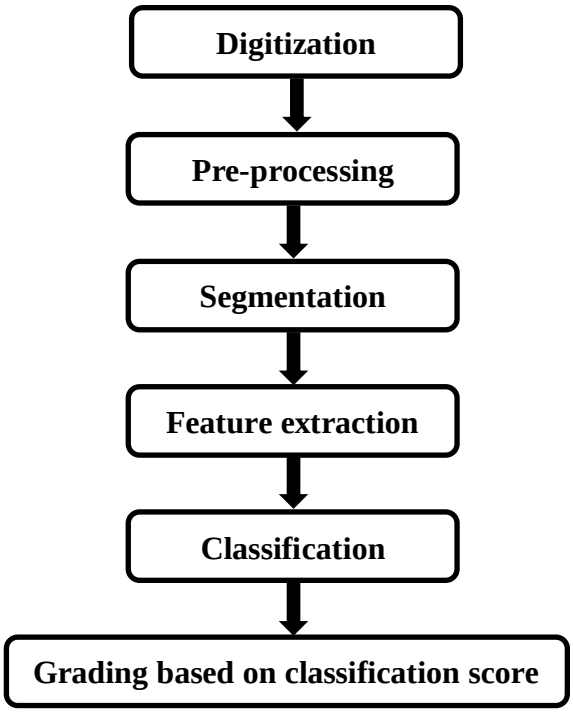


Figure 4.1: Block diagram of handwriting grading system

The phases, namely, digitization, pre-processing and segmentation have been discussed in Chapter 2. We have used digitization phase as discussed in section 2.3.1; pre-processing activities have been applied as discussed in section 2.3.2. We have segmented the handwritten *Gurmukhi* script document into characters using the techniques as discussed in section 2.3.3.

As we have already discussed, feature extraction stage analyzes a handwritten character image and selects a set of features that can be used for grading the writers. In this work, for grading of the writers, we have used various structural feature extraction techniques, namely, zoning, diagonal, directional, intersection and open end points features *etc.* as discussed in section 2.3.4.

Classification phase uses the features extracted in the feature extraction phase, for obtain classification score in the handwriting grading system. For classification, we have used HMM and Bayesian decision making classifiers. A HMM is a finite state machine that may move to a next state at each time unit. With each move, an observed vector is generated. Probabilities in HMM are calculated by utilizing an observation vector extracted from samples of handwritten *Gurmukhi* characters. Recognition of an unknown character is based on the probability that an unknown character is generated by HMM. The Bayesian classifier is a statistical approach that allows designing the optimal classifier if the complete statistical model is known. In this classifier, a character is assigned to the class for which it has the highest probability conditioned on X, where X is the observed feature vector.

4.1.1 Grading based on classification score

As mentioned above, the writers are graded based on their classification score. The score is obtained on the basis of the process illustrated in Figure 4.1.

In order to build the proposed grading system, we have collected data from one hundred different writers. These writers were requested to write each *Gurmukhi* character. A sample of this handwritten character data set, written by ten different writers (W_1, W_2, …, W_{10}) is given in Figure 4.2.

In this work, besides the handwritten characters from these one hundred writers, printed *Gurmukhi* characters from five different fonts have also been considered. This has, primarily, been done for establishing the correctness of the approach considered in this chapter. These five fonts are: *Amrit* (F_1), *GurmukhiLys* (F_2), *Granthi* (F_3), *LMP_TARAN* (F_4) and *Maharaja* (F_5). Sample of a few characters in these fonts is given in Figure 4.3.

Gurmukhi Script Character	W_1	W_2	W_3	W_4	W_5	W_6	W_7	W_8	W_9	W_{10}
ੳ										
ਅ										
ੲ										
ਸ										

Figure 4.2: Samples of a few handwritten *Gurmukhi* characters

Script Character	F_1	F_2	F_3	F_4	F_5
ੳ					
ਅ					
ੲ					
ਸ					

Figure 4.3: A few samples of printed characters from five *Gurmukhi* fonts

In the next section, classifier-wise results of grading have been included.

4.2 Experimental results of handwriting grading system

As discussed in section 4.1.1, the gradation results, based on the values obtained by two classifiers, namely, HMM and Bayesian classifiers are presented in this section. The

probabilities obtained with HMM classification and Bayesian classification are normalized to [0, 100] in order to give the grade in percentage form. Classifier-wise results of grading are presented in the following sub-sections. In the training data set of handwriting grading system, we have used printed *Gurmukhi* font *Anandpur Sahib*. The shape of various *Gurmukhi* characters *in Anandpur Sahib* font has been shown in Figure 4.4.

In the testing data set, we have considered data set of handwritten characters written by one hundred different writers and printed characters of five different *Gurmukhi* fonts as discussed in section 4.1.1.

The experimental results in this chapter have been presented in the form of graphs. These graphs present grading scores obtained for one hundred writers (W_1, W_2, …, W_{100}) and five *Gurmukhi* fonts (F_1, F_2, …, F_5). For the sake of better space usage, the calibration on x-axis does not include all the values. However, graphs contain the data for all 105 points.

S. No.	Character	S. No.	Character	S. No.	Character	S. No.	Character
1	ੳ	2	ਅ	3	ੲ	4	ਸ
5	ਹ	6	ਕ	7	ਖ	8	ਗ
9	ਘ	10	ਙ	11	ਚ	12	ਛ
13	ਜ	14	ਝ	15	ਞ	16	ਟ
17	ਠ	18	ਡ	19	ਢ	20	ਣ
21	ਤ	22	ਥ	23	ਦ	24	ਧ
25	ਨ	26	ਪ	27	ਫ	28	ਬ
29	ਭ	30	ਮ	31	ਯ	32	ਰ
33	ਲ	34	ਵ	35	ੜ		

Figure 4.4: Shape of characters in *Gurmukhi* font *Anandpur Sahib*

4.2.1 Grading using HMM classifier

In order to perform the grading using HMM classifier, the features, namely, zoning, directional, diagonal, intersection and open end points have been taken as input to HMM classifier. In the following sub-sections, the experimental results for these features are presented.

4.2.1.1 HMM based grading using zoning features

In this sub-section, gradation results of writers based on zoning features, using HMM classifier, are presented. Using this feature, it has been noted that font F_4 (with a score of 100) is the best font and font F_5 (with a score of 94.55) is the second best font. On similar lines, it has also been observed that writer W_{14} (with a score of 68.88) is the best writer and writer W_{50} (with a score of 52.19) is the second best writer. The results of this classification process are presented in Figure 4.5.

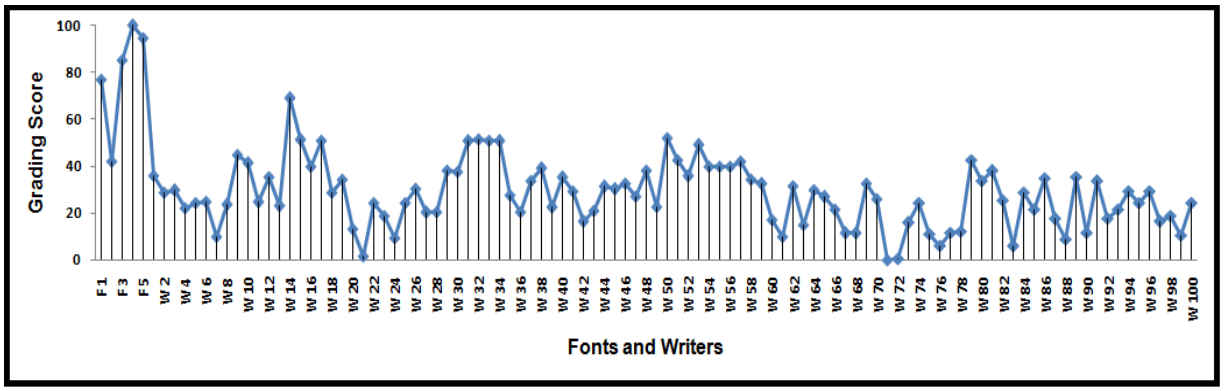


Figure 4.5: Grading of writers using zoning feature and HMM classifier

4.2.1.2 HMM based grading using directional features

When we use directional features as an input to HMM classifier, font F_1 (with a score of 100) comes out to be the best font and font F_2 (with a score of 80.75) comes out to be the next best font. It has also been observed that writer W_{79} (with a score of 83.20) is the best writer and writer W_{80} (with a score of 78.92) is the next best writer amongst the hundred writers taken in this study. Results of HMM based classification using directional feature are

given in Figure 4.6.

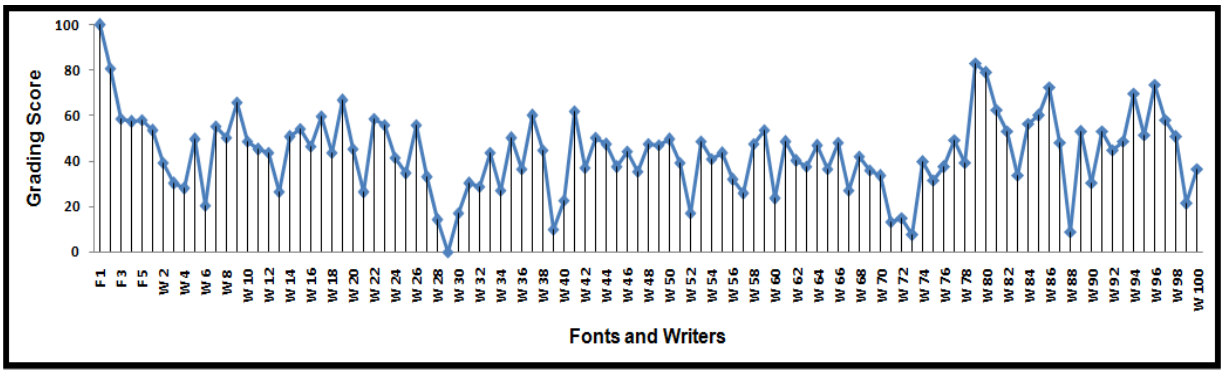


Figure 4.6: Grading of writers using directional features and HMM classifier

4.2.1.3 HMM based grading using diagonal features

In this sub-section, we have presented grading results based on diagonal features using HMM classification. Using this feature, we have seen that font F_4 (with a score of 100) is the best font and the second rank goes to font F_1 (with a score of 95.58). It has also been seen that writer W_{53} (with a score of 62.80) is the best writer and writer W_{15} (with a score of 61.35) is the next best writer from among the hundred writers taken in this work. Results of HMM based classification, using diagonal features, are given in Figure 4.7.

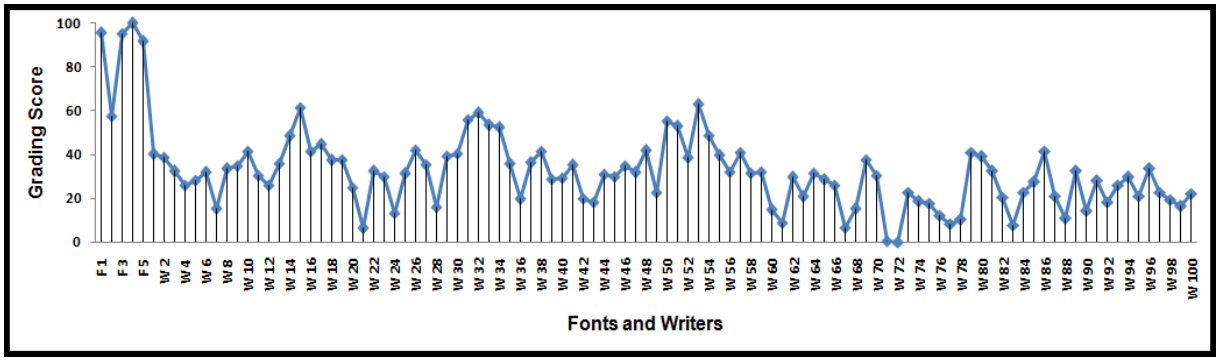


Figure 4.7: Grading of writers using diagonal features and HMM classifier

4.2.1.4 HMM based grading using intersection points based features

When we use intersection points based features as input to HMM classifier, it has been seen that font F_3 (with a score of 100) is the best font and font F_4 (with a score of 99.59) is

the next best font. It has also been seen that writer W_{16} (with a score of 67.79) is the best writer and writer W_{50} (with a score of 65.50) is the second best writer. The results of HMM based classification, using intersection points based features, are discussed in Figure 4.8.

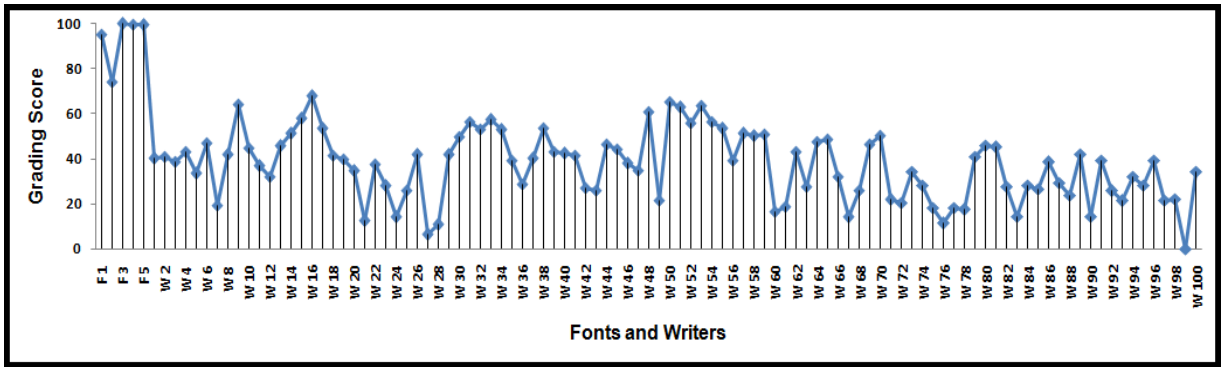


Figure 4.8: Grading of writers using intersection points based features and HMM classifier

4.2.1.5 HMM based grading using open end points based features

In this sub-section, gradation results of writers, based on the open end points based features using HMM classifier, are presented. Using this feature, it has been seen that font F_1 (with a score of 100) is the best font and font F_5 (with a score of 91.79) is the second best font. It has also been discovered that writer W_{33} (with a score of 71.79) is the best writer and writer W_{31} (with a score of 69.25) is the second best writer. The results of this classification process are presented in Figure 4.9.

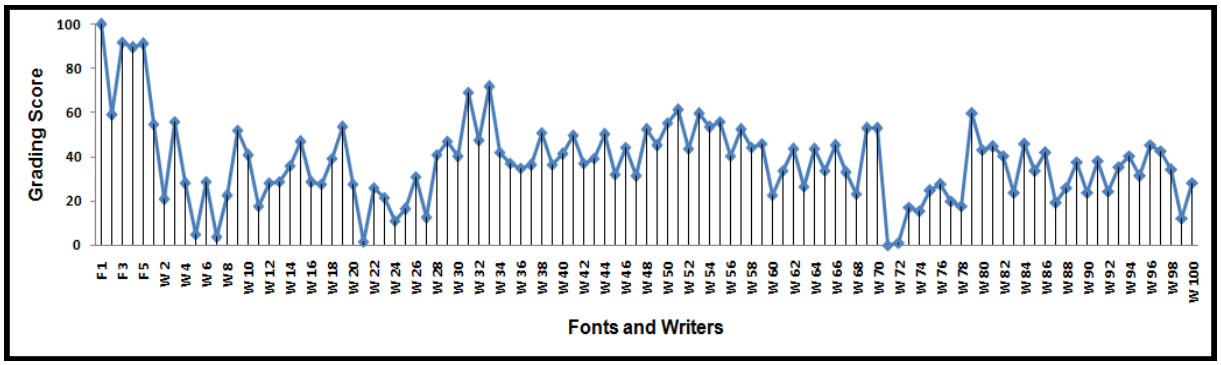


Figure 4.9: Grading of writers using open end points based features and HMM classifier

4.2.1.6 Average grading of writers with HMM classifier

Here, average grading, based on all five features considered in sections 4.2.1.1 to 4.2.1.5 is presented. It has been observed that if we use HMM classifier then font F_1 (with an average score of 93.50) is the best font and font F_4 (with an average score of 89.26) is the next best font. Similarly, it has also been observed that writer W_{53} (with an average score of 56.79) is the best writer and writer W_{50} (with an average score of 55.62) is the second best writer. The average scores of the fonts and writers considered in this study are given in Figure 4.10.

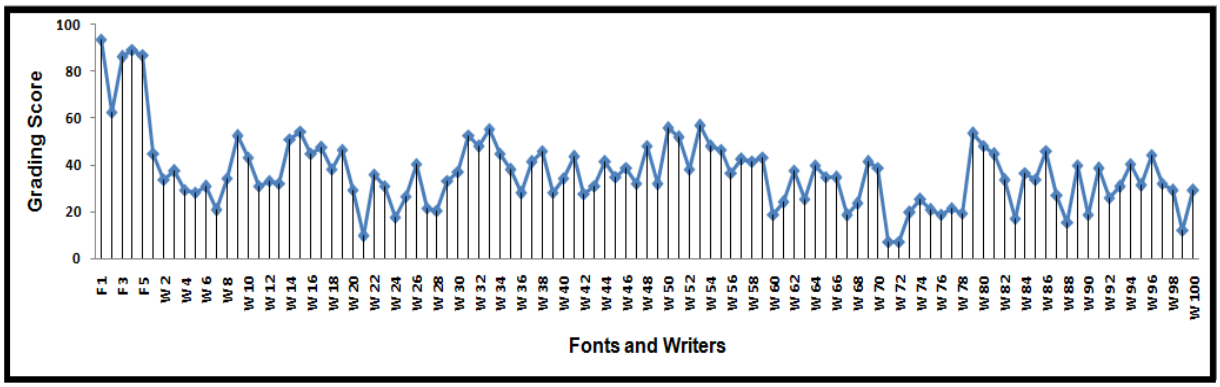


Figure 4.10: Average grading of writers using HMM classifier

Table 4.1: Average grading of writers using HMM classifier

Feature type	Best font and writer	Second best font and writer	Third best font and writer	Fourth best font and writer	Fifth best font and writer
Zoning features	F_4 (100); W_{14} (68.88)	F_5 (94.55); W_{50} (52.19)	F_3 (85.13); W_{32} (51.55)	F_1 (76.79); W_{15} (51.23)	F_2 (42.03); W_{17} (51.01)
Directional features	F_1 (100); W_{79} (83.20)	F_2 (80.75); W_{80} (78.92)	F_3 (58.55); W_{96} (73.52)	F_5 (58.24); W_{86} (72.40)	F_4 (57.33); W_{94} (69.76)
Diagonal features	F_4 (100); W_{53} (62.80)	F_1 (95.58); W_{15} (61.35)	F_3 (95.44); W_{32} (58.91)	F_5 (91.62); W_{31} (55.74)	F_2 (57.72); W_{50} (55.34)
Intersection points based features	F_3 (100); W_{16} (67.79)	F_4 (99.59); W_{50} (65.50)	F_5 (99.37); W_9 (64.35)	F_1 (95.11); W_{53} (63.74)	F_2 (74.08); W_{51} (62.77)
Open end points based features	F_1 (100); W_{33} (71.79)	F_3 (91.79); W_{31} (69.25)	F_5 (91.02); W_{51} (61.20)	F_4 (89.36); W_{53} (59.55)	F_2 (58.95); W_{79} (59.50)
Average with all features	F_1 (93.50); W_{53} (56.79)	F_4 (89.26); W_{50} (55.62)	F_5 (86.96); W_{33} (55.39)	F_3 (86.18); W_{15} (54.28)	F_2 (62.71); W_{79} (53.45)

Table 4.1 depicts the average grading of writers using HMM classifier. As shown in this table, we have seen that Font F_1 is close to *Anandpur Sahib* font and writer W_{53} is the best writer, if we use HMM classifier.

4.2.2 Grading using Bayesian classifier

In this section, the experimental results of grading are presented for the case when we used Bayesian classifier. Here, features, namely, zoning, directional, diagonal, intersection and open end points features have again been considered to be taken as input to Bayesian classifier.

4.2.2.1 Bayesian based grading using zoning features

In this sub-section, gradation results of writers, based on zoning features using Bayesian classifier, are presented. Using this feature, it has been observed that font F_4 (with a score of 100) is the best font and font F_5 (with a score of 94.60) is the second best font. On similar lines, it has also been found that writer W_{14} (with a score of 71.37) is the best writer and writer W_{17} (with a score of 58.52) is the second best writer. The results of this classification process are presented in Figure 4.11.

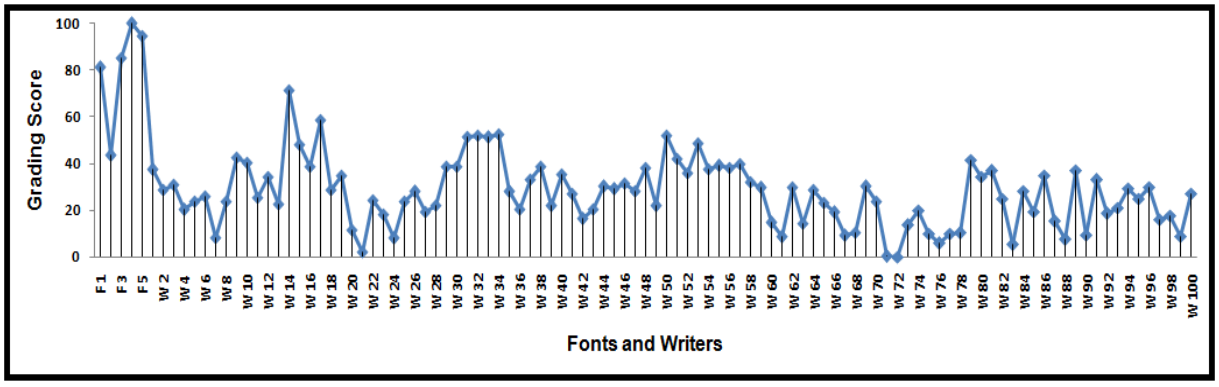


Figure 4.11: Grading of writers using zoning feature and Bayesian classifier

4.2.2.2 Bayesian based grading using directional features

When directional features are used as input to Bayesian classifier, font F_1 (with a score of 73.76) comes out to be the best font and font F_2 (with a score of 60.38) comes out to be the second best font. Also, writer W_1 (with a score of 100) comes out to be the best writer and writer W_{79} (with a score of 58.86) comes out to be the second best writer. The results of this classification process are presented in Figure 4.12.

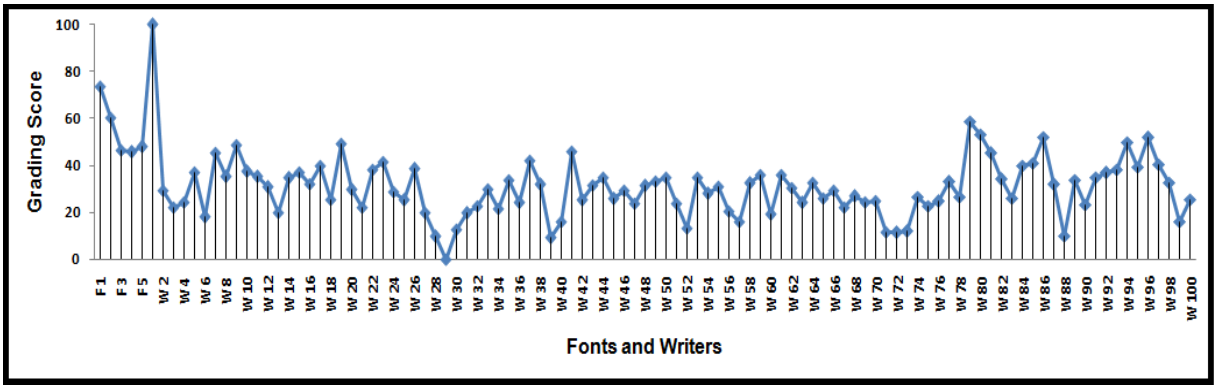


Figure 4.12: Grading of writers using directional feature and Bayesian classifier

4.2.2.3 Bayesian based grading using diagonal features

Results of Bayesian based classification, using diagonal feature, are given in Figure 4.13.

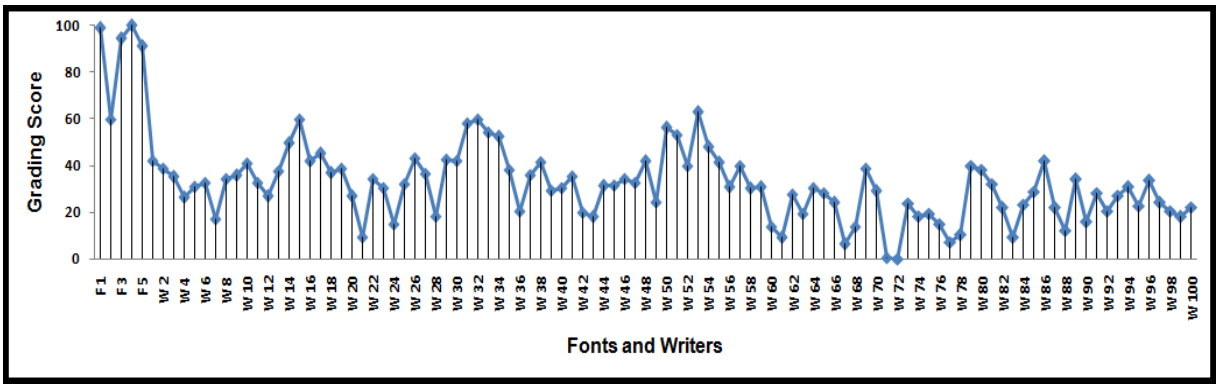


Figure 4.13: Grading of writers using diagonal features and Bayesian classifier

The use of diagonal features as input to Bayesian classifier results into a situation where font F_4 (with a score of 100) comes out to be the best font and font F_1 (with a score of 98.92)

comes out to be the second best font; writer W_{53} (with a score of 62.81) is the best writer and writer W_{15} (with a score of 59.78) is the second best writer amongst the hundred writers engaged in this work.

4.2.2.4 Bayesian based grading using intersection points based features

When we use intersection points based features as input to Bayesian classifier, font F_3 (with a score of 100) comes out to be the best font and font F_4 (with a score of 99.40) comes out to be the second best. Writer W_{16} (with a score of 67.66) comes out to be the best writer and writer W_9 (with a score of 64.97) comes out to be the second best writer. Results of Bayesian based classification, using intersection points based features are given in Figure 4.14.

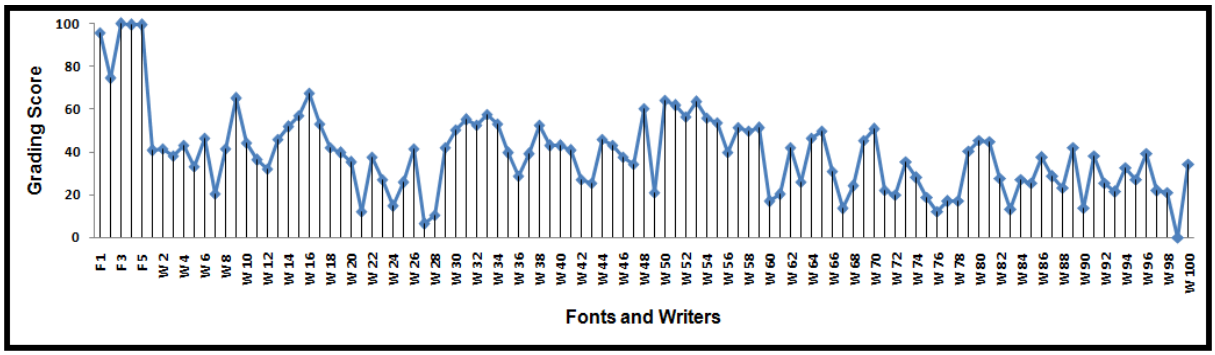


Figure 4.14: Grading of writers using intersection points based features and Bayesian classifier

4.2.2.5 Bayesian based grading using open end points based features

If we consider open end points based features as input to Bayesian classifier, font F_1 (with a score of 100) comes out to be the best font and font F_3 (with a score of 92.11) comes out to be the second best font; writer W_{33} (with a score of 71.18) comes out to be the best writer and writer W_{31} (with a score of 67.08) comes out to be the second best writer.

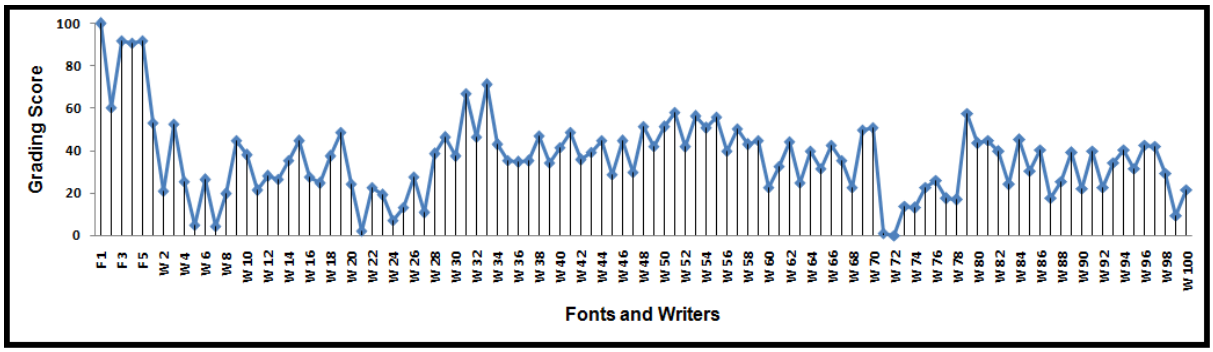


Figure 4.15: Grading of writers using open end points based features and Bayesian classifier

The results of this classification process are presented in Figure 4.15.

4.2.2.6 Average grading of writers with Bayesian classifier

Average grading, based on all five features taken in sections 4.2.2.1 to 4.2.2.5 is presented in this sub-section. It has been seen that if we use Bayesian classifier then font F_1 (with an average score of 89.93) is the best font and font F_4 (with an average score of 87.15) is the next best font. On similar lines, it has also been depicted that writer W_1 (with an average score of 54.65) is the best writer and writer W_{53} (with an average score of 53.26) is the second best writer. The results of this classification process are presented in Figure 4.16.

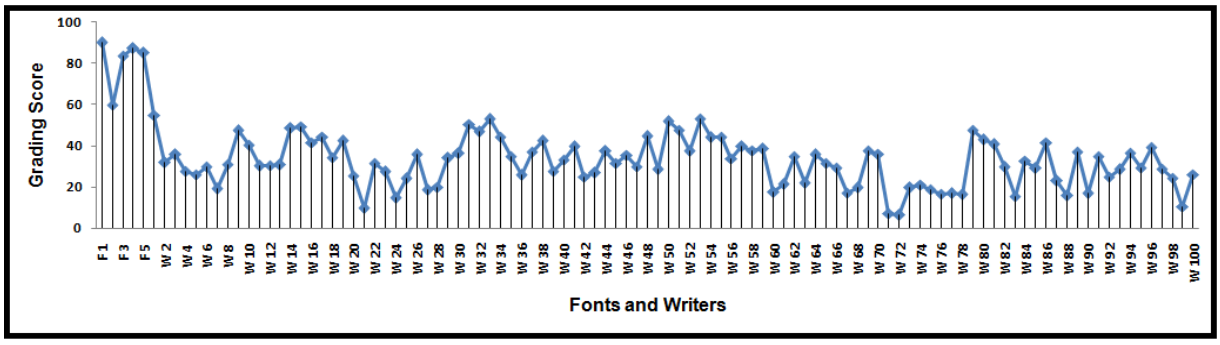


Figure 4.16: Average grading of writers using Bayesian classifier

Table 4.2 presents the average grading of writers using Bayesian classifier. One can note from this table that Font F_1 is the best font and writer W_1 is the best writer, if we consider Bayesian classifier for obtaining the classification score.

Table 4.2: Average grading of writers using Bayesian classifier

Feature type	Best font and writer	Second best font and writer	Third best font and writer	Fourth best font and writer	Fifth best font and writer
Zoning features	F_4 (100); W_{14} (71.37)	F_5 (94.60); W_{17} (58.52)	F_3 (85.15); W_{34} (52.60)	F_1 (81.16); W_{32} (52.20)	F_2 (43.41); W_{50} (52.13)
Directional features	F_1 (73.76); W_1 (100)	F_2 (60.38); W_{79} (58.86)	F_5 (47.91); W_{80} (53.22)	F_3 (46.33); W_{86} (52.03)	F_4 (45.60); W_{96} (51.69)
Diagonal features	F_4 (100); W_{53} (62.81)	F_1 (98.92); W_{15} (59.78)	F_3 (94.38); W_{32} (59.68)	F_5 (91.13); W_{31} (57.95)	F_2 (59.78); W_{50} (56.10)
Intersection points based features	F_3 (100); W_{16} (67.66)	F_4 (99.40); W_9 (64.97)	F_5 (99.40); W_{50} (64.37)	F_1 (95.81); W_{53} (63.47)	F_2 (74.55); W_{51} (61.68)
Open end points based features	F_1 (100); W_{33} (71.18)	F_3 (92.11); W_{31} (67.08)	F_5 (91.96); W_{51} (58.30)	F_4 (90.77); W_{79} (57.41)	F_2 (60.39); W_{53} (56.22)
Average with all features	F_1 (89.93); W_1 (54.66)	F_4 (87.15); W_{53} (53.26)	F_5 (85.00); W_{33} (52.82)	F_3 (83.59); W_{50} (51.71)	F_2 (59.70); W_{31} (50.24)

4.2.3 Average grading with five features and two classifiers

We have also calculated the average grading when all five features and the two classifiers are considered simultaneously in the experimentation. It has been seen in this experiment that font F_1 (with an average score of 91.71) is the best font and font F_4 (with an average score of 88.21) is then the next best font. Writer W_{53} (with an average score of 55.02) is the best writer and writer W_{33} (with an average score of 54.11) is the second best writer. The results of this average grading are presented in Figure 4.17.

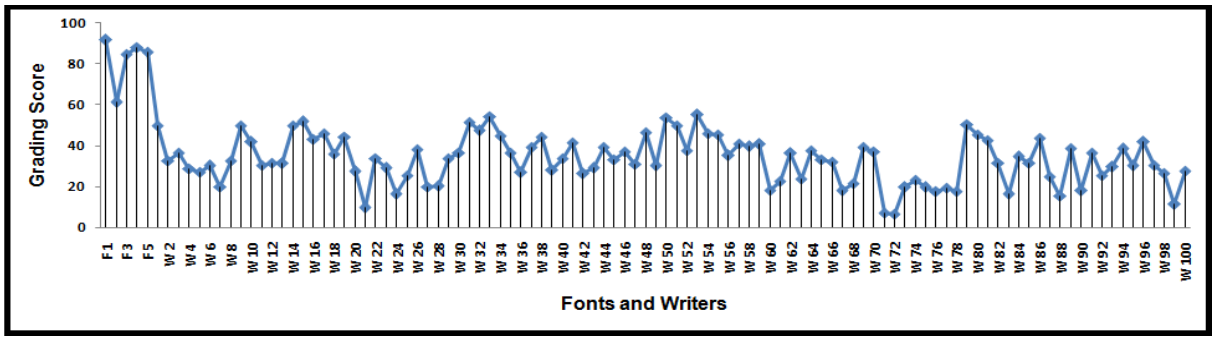


Figure 4.17: Average grading of writers using all features and classifiers

It is worth mentioning here that the two classifiers considered in this study have a good agreement while grading the fonts as well as the human writers. This is supported by the graph, almost a straight line as depicted in Figure 4.18 in which the grading score obtained by Bayesian classifier is taken as a function of grading score obtained by HMM classifier. The data points in green colour represent the grading scores of printed fonts.

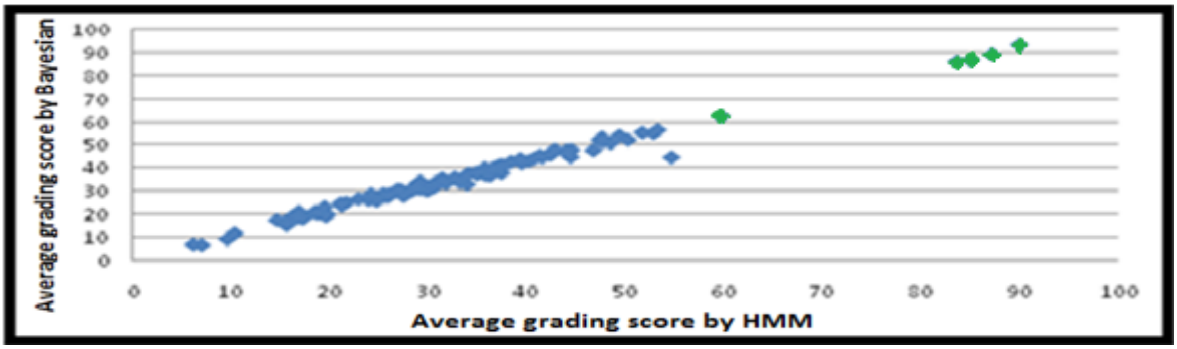


Figure 4.18: Agreement between two classifiers

Table 4.3 shows the performance of five best writers for different features and classifiers, considered in this study.

Table 4.3: Classifier wise performance of the five best writers

Feature Type	HMM Classifier	Bayesian Classifier
Zoning features	W_{14} (68.88); W_{50} (52.19); W_{32} (51.55); W_{15} (51.23); W_{17} (51.01)	W_{14} (71.37); W_{17} (58.52); W_{34} (52.60); W_{32} (52.20); W_{50} (52.13)
Directional features	W_{79} (83.20); W_{80} (78.92); W_{96} (73.52); W_{86} (72.40); W_{94} (69.76)	W_{1} (100); W_{79} (58.86); W_{80} (53.22); W_{86} (52.03); W_{96} (51.69)
Diagonal features	W_{53} (62.80); W_{15} (61.35); W_{32} (58.91); W_{31} (55.74); W_{50} (55.34)	W_{53} (62.81); W_{15} (59.78); W_{32} (59.68); W_{31} (57.95); W_{50} (56.10)
Intersection points based features	W_{16} (67.79); W_{50} (65.50); W_{9} (64.35); W_{53} (63.74); W_{51} (62.77)	W_{16} (67.66); W_{9} (64.97); W_{50} (64.37); W_{53} (63.47); W_{51} (61.68)
Open end points based features	W_{33} (71.79); W_{31} (69.25); W_{51} (61.20); W_{53} (59.55); W_{79} (59.50)	W_{33} (71.18); W_{31} (67.08); W_{51} (58.30); W_{79} (57.41); W_{53} (56.22)
Average with all features	W_{53} (56.79); W_{50} (55.62); W_{33} (55.39); W_{15} (54.28); W_{79} (53.45)	W_{1} (54.66); W_{53} (53.26); W_{33} (52.82); W_{50} (51.71); W_{31} (50.24)

4.3 Discussions and conclusion

In this chapter, an offline handwriting grading system for *Gurmukhi* script writers has been proposed. The features of offline *Gurmukhi* characters that have been considered in this work include zoning; directional; diagonal; intersection and open end points. Two classifiers, namely, HMM classifier and Bayesian classifier, have been used in the classification process. The system, proposed in present study, is tested with the help of five popular printed *Gurmukhi* fonts. As expected, fonts have a better score of gradation in comparison with mortal writers, establishing the effectiveness of the proposed system. The proposed grading system can be used as a decision support system for grading the handwritings in a competition.

Chapter 5

Parabola and Power Curve Based Novel Feature Extraction Methods for Offline Handwritten Gurmukhi Character Recognition

==========

Recent advances in optical character recognition have been supported by innovative techniques for handwritten character recognition. These techniques require extraction of good quality features as their input for recognition process. In this chapter, we have presented two efficient feature extraction techniques, namely, parabola curve fitting based features and power curve fitting based features for offline handwritten *Gurmukhi* character recognition. In order to assess the quality of features in offline handwritten *Gurmukhi* character recognition, we have compared the performance of other recently used feature extraction techniques, namely, zoning features, diagonal features, intersection and open end points features, transition features and directional features with these proposed feature extraction techniques. This chapter is divided into four sections. Section 5.1 consists of parabola curve fitting based feature extraction technique, section 5.2 includes power curve fitting based feature extraction technique, section 5.3 presents the experimental results based on proposed feature extraction techniques and comparison with other recently used feature extraction techniques and section 5.4 presents the summary of this chapter.

5.1 Parabola curve fitting based feature extraction (Proposed Method I)

Curve fitting is the process of constructing a curve that has the best fit to a series of foreground pixels. A fitted curve can be used as an aid for data visualization. Parabola is a curve that is shaped like the path of something that is thrown forward and high in the air and

falls back to the ground. Its equation is $y = a + bx + cx^2$. Parabolas occur naturally as the paths of projectiles. The shape is also seen in the design of bridges and arches. We have used this type of curve to extract the meaningful information about the strokes of offline handwritten characters.

In parabola fitting based feature extraction method, we have divided the thinned image of a character into *n* (=100) zones. A parabola is then fitted to the series of ON pixels (foreground pixels) in each zone using the Least Square Method (LSM). A parabola $y = a + bx + cx^2$ is uniquely defined by three parameters: *a*, *b* and *c*. Values of *a*, *b* and *c* are calculated by solving the following equations obtained from LSM.

$$\sum_{i=1}^{n} y_i = na + b\sum_{i=1}^{n} x_i + c\sum_{i=1}^{n} {x_i}^2 \quad (1)$$

$$\sum_{i=1}^{n} x_i y_i = a\sum_{i=1}^{n} x_i + b\sum_{i=1}^{n} {x_i}^2 + c\sum_{i=1}^{n} {x_i}^3 \quad (2)$$

$$\sum_{i=1}^{n} {x_i}^2 y_i = a\sum_{i=1}^{n} {x_i}^2 + b\sum_{i=1}^{n} {x_i}^3 + c\sum_{i=1}^{n} {x_i}^4 \quad (3)$$

As such, this will give 3*n* features for a given character image as depicted in Table 5.1 for the *Gurmukhi* character (ਕ).

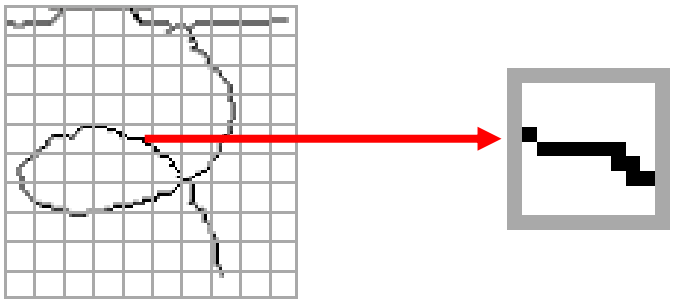


Figure 5.1: Parabola curve fitting based feature extraction technique

The steps that have been used to extract these features are given below.

Step I: Divide the thinned image into *n* (=100) number of equal sized zones.

Step II: For each zone, fit a parabola using the least square method and calculate the values of *a*, *b* and *c* (Figure 5.1).

Step III: Corresponding to the zones that do not have a foreground pixel, set the values of *a*, *b* and *c* as zero.

Step IV: Normalize the feature values in the scale [0, 1] as follows:

$$\text{Normalized feature } NV_i = \frac{(\text{Actual feature } V_i - \text{min of actual feature vector})}{(\text{max of actual feature vector} - \text{min of actual feature vector})}$$

Table 5.1: Parabola fitting based feature values for the *Gurmukhi* character (ਕ) given in Figure 5.1

Zone	*a*	*b*	*c*	Zone	*a*	*b*	*c*
Z_1	0	0	1	Z_2	0.9142	0.9498	0.2532
Z_3	0.9709	0.9799	0.1307	Z_4	1	1	0.0988
Z_5	0.9401	0.9467	0.101	Z_6	0.7085	0.7886	0.1575
Z_7	0.3689	0.5425	0.2814	Z_8	0.3515	0.5123	0.2313
Z_9	0.331	0.4853	0.2065	Z_{10}	0.3074	0.4562	0.1869
Z_{11}	0	0	0	Z_{12}	0	0	0
Z_{13}	0	0	0	Z_{14}	0	0	0
Z_{15}	0	0	0	Z_{16}	0	0	0
Z_{17}	0.2502	0.4164	0.1918	Z_{18}	0.2451	0.4089	0.1738
Z_{19}	0	0	0	Z_{20}	0	0	0
Z_{21}	0	0	0	Z_{22}	0	0	0
Z_{23}	0	0	0	Z_{24}	0	0	0
Z_{25}	0	0	0	Z_{26}	0	0	0
Z_{27}	0	0	0	Z_{28}	0.2106	0.381	0.1794
Z_{29}	0	0	0	Z_{30}	0	0	0
Z_{31}	0	0	0	Z_{32}	0	0	0
Z_{33}	0	0	0	Z_{34}	0	0	0
Z_{35}	0	0	0	Z_{36}	0	0	0
Z_{37}	0	0	0	Z_{38}	0.1946	0.3713	0.1722
Z_{39}	0	0	0	Z_{40}	0	0	0
Z_{41}	0	0	0	Z_{42}	0.2032	0.3778	0.1583
Z_{43}	0.2057	0.3792	0.1659	Z_{44}	0.2127	0.387	0.1557
Z_{45}	0.2029	0.3748	0.1456	Z_{46}	0.2038	0.3723	0.1381
Z_{47}	0	0	0	Z_{48}	0.2079	0.3761	0.1416
Z_{49}	0	0	0	Z_{50}	0	0	0
Z_{51}	0.2079	0.3781	0.1397	Z_{52}	0.2058	0.3746	0.131
Z_{53}	0	0	0	Z_{54}	0	0	0
Z_{55}	0	0	0	Z_{56}	0.1991	0.3725	0.1338
Z_{57}	0.2062	0.3768	0.1263	Z_{58}	0.2021	0.3706	0.1175
Z_{59}	0	0	0	Z_{60}	0	0	0
Z_{61}	0.2067	0.377	0.1186	Z_{62}	0.2024	0.3719	0.1173
Z_{63}	0.1927	0.3624	0.1161	Z_{64}	0.1914	0.3617	0.1121
Z_{65}	0.1936	0.3625	0.1075	Z_{66}	0.1922	0.36	0.1102
Z_{67}	0.1874	0.3552	0.1097	Z_{68}	0	0	0
Z_{69}	0	0	0	Z_{70}	0	0	0
Z_{71}	0	0	0	Z_{72}	0	0	0

Z_{73}	0.1937	0.3641	0.1182	Z_{74}	0	0	0
Z_{75}	0	0	0	Z_{76}	0	0	0
Z_{77}	0.1993	0.3722	0.1179	Z_{78}	0.1972	0.3672	0.1169
Z_{79}	0	0	0	Z_{80}	0	0	0
Z_{81}	0	0	0	Z_{82}	0	0	0
Z_{83}	0	0	0	Z_{84}	0	0	0
Z_{85}	0	0	0	Z_{86}	0	0	0
Z_{87}	0	0	0	Z_{88}	0.1937	0.3615	0.1162
Z_{89}	0	0	0	Z_{90}	0	0	0
Z_{91}	0	0	0	Z_{92}	0	0	0
Z_{93}	0	0	0	Z_{94}	0	0	0
Z_{95}	0	0	0	Z_{96}	0	0	0
Z_{97}	0	0	0	Z_{98}	0.1936	0.361	0.1135
Z_{99}	0	0	0	Z_{100}	0	0	0

5.2 Power curve fitting based feature extraction (Proposed Method II)

A power curve of the form $y = ax^b$ is uniquely defined by two parameters: *a* and *b*. In power curve fitting based feature extraction technique, the thinned image of a character is again divided into *n* (=100) zones. A power curve is fitted to the series of ON pixels (foreground pixels) in each zone using LSM. Thus, the values of *a* and *b* are calculated by the following process.

The power curve is given by:

$$y = ax^b$$

This gives,

$$\log y = \log a + b \log x$$

Let us take,

log *y*= *Y*, log *a* = *A* and log *x* = *X*, This gives rise to a linear relationship in *X and Y*. The normal equations are now obtained using LSM. These are solved and the value of *a* is obtained using the above relationship.

As a result of this curve fitting, we will obtain 2*n* features for a character image as shown in Table 5.2 for the *Gurmukhi* character (ਕ).

The steps that have been used to extract these features are given below.

Step I: Divide the thinned image into *n* (= 100) number of equal sized zones.

Step II: In each zone, fit a power curve using the least square method and calculate the

values of *a* and *b*.

Step III: Corresponding to the zones that do not have a foreground pixel, set the value of *a* and *b* as zero.

Step IV: Normalize the feature values in the scale [0, 1] as follows:

$$\text{Normalized feature } NV_i = \frac{(\text{Actual feature } V_i - \min \text{ of actual feature vector})}{(\max \text{ of actual feature vector} - \min \text{ of actual feature vector})}$$

Table 5.2: Power curve fitting based feature values for the *Gurmukhi* character (ਕ) given in Figure 5.1

Zone	*a*	*b*	Zone	*a*	*b*
Z_1	0.5816	0.2922	Z_2	0.1519	0.4594
Z_3	0.0639	0.5629	Z_4	0.01	0.785
Z_5	0.0025	1	Z_6	0.1596	0.4738
Z_7	0.2913	0.3784	Z_8	0.4669	0.3238
Z_9	0.5726	0.304	Z_{10}	0.6607	0.2873
Z_{11}	0	0	Z_{12}	0	0
Z_{13}	0	0	Z_{14}	0	0
Z_{15}	0	0	Z_{16}	0	0
Z_{17}	0.6785	0.2897	Z_{18}	0.7393	0.2726
Z_{19}	0	0	Z_{20}	0	0
Z_{21}	0	0	Z_{22}	0	0
Z_{23}	0	0	Z_{24}	0	0
Z_{25}	0	0	Z_{26}	0	0
Z_{27}	0	0	Z_{28}	0.731	0.2751
Z_{29}	0	0	Z_{30}	0	0
Z_{31}	0	0	Z_{32}	0	0
Z_{33}	0	0	Z_{34}	0	0
Z_{35}	0	0	Z_{36}	0	0
Z_{37}	0	0	Z_{38}	0.767	0.2672
Z_{39}	0	0	Z_{40}	0	0
Z_{41}	0	0	Z_{42}	0.8073	0.2601
Z_{43}	0.798	0.2626	Z_{44}	0.8086	0.2662
Z_{45}	0.8543	0.2602	Z_{46}	0.8842	0.2506
Z_{47}	0	0	Z_{48}	0.8697	0.2508
Z_{49}	0	0	Z_{50}	0	0
Z_{51}	0.8855	0.2511	Z_{52}	0.9151	0.2473
Z_{53}	0	0	Z_{54}	0	0
Z_{55}	0	0	Z_{56}	0.8991	0.2524
Z_{57}	0.9162	0.2436	Z_{58}	0.9487	0.2393
Z_{59}	0	0	Z_{60}	0	0

Z_{61}	0.938	0.2426	Z_{62}	0.9503	0.2387
Z_{63}	0.9687	0.2332	Z_{64}	0.9757	0.2288
Z_{65}	0.9941	0.2252	Z_{66}	0.9954	0.2252
Z_{67}	1	0.2254	Z_{68}	0	0
Z_{69}	0	0	Z_{70}	0	0
Z_{71}	0	0	Z_{72}	0	0
Z_{73}	0	0	Z_{74}	0	0
Z_{75}	0	0	Z_{76}	0	0
Z_{77}	0.9528	0.2338	Z_{78}	0.9636	0.2282
Z_{79}	0	0	Z_{80}	0	0
Z_{81}	0	0	Z_{82}	0	0
Z_{83}	0	0	Z_{84}	0	0
Z_{85}	0	0	Z_{86}	0	0
Z_{87}	0	0	Z_{88}	0.9658	0.2264
Z_{89}	0	0	Z_{90}	0	0
Z_{91}	0	0	Z_{92}	0	0
Z_{93}	0	0	Z_{94}	0	0
Z_{95}	0	0	Z_{96}	0	0
Z_{97}	0	0	Z_{98}	0.9767	0.2252
Z_{99}	0	0	Z_{100}	0	0

5.3 Experimental results

In this section, we have presented the experimental results based on proposed feature extraction techniques and a comparison has also been made with the existing features. Two classifiers, namely, *k*-NN and SVM have been utilized in this work in order to compare the proposed feature extraction techniques with other recently used feature extraction techniques, namely, zoning features, diagonal features, intersection and open end points features, directional features, and transition features. Each technique has been tested by using 5600 samples of isolated offline handwritten *Gurmukhi* characters.

Table 5.3: Five distinct types of partitioning

Type	Training Data	Testing Data
a	50%	50%
b	60%	40%
c	70%	30%
d	80%	20%
e	90%	10%

In order to find the best feature set for a given offline handwritten *Gurmukhi* character, a performance analysis has also been carried out. We have partitioned the data set in five different ways *a*, *b*, *c*, *d* and *e* as depicted below in Table 5.3.

5.3.1 Performance analysis based on *k*-NN classifier

In this sub-section, experimental results of the data set for the partitions (*a, b,…, e*), based on *k*-NN classifier, are presented (Table 5.4). We have carried out experiments using *k*-NN classifier for the values of *k* = 1, 3, 5, and 7. These experiments gave the best recognition accuracy for the value of *k* = 5. It has been noticed that power curve fitting based features with *k*-NN classifier, achieved the maximum recognition accuracy of 97.9% when we used data set partitioning strategy *c*. These results are graphically shown in Figure 5.2.

Table 5.4: Recognition accuracy based on *k*-NN classifier for various feature extraction techniques

Strategy	Feature extraction techniques						
	Zoning	**Diagonal**	**Directional**	**Intersection and open end point**	**Transition**	**Parabola curve fitting**	**Power curve fitting**
a	82.9%	81.9%	79.8%	81.3%	79.8%	**92.9%**	**96.7%**
b	82.2%	82.5%	81.3%	82.2%	81.3%	**93.1%**	**97.9%**
c	79.6%	84.1%	76.5%	82.4%	84.1%	**94.5%**	**97.9%**
d	81.9%	84.0%	81.4%	86.7%	84.0%	**94.1%**	**97.0%**
e	89.7%	93.1%	86.6%	83.7%	83.1%	**95.4%**	**96.3%**

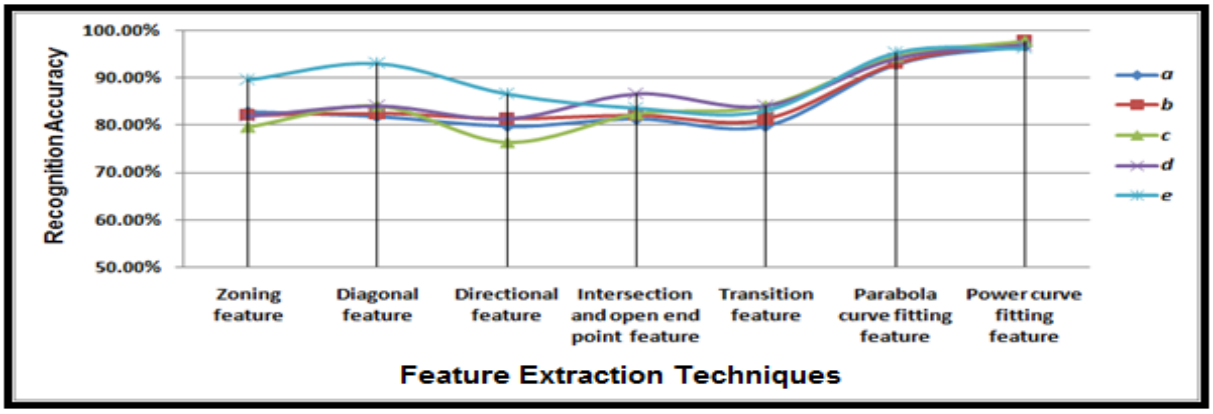


Figure 5.2: Recognition accuracy based on *k*-NN classifier for various feature extraction techniques

5.3.2 Performance analysis based on SVM with linear kernel classifier

In this sub-section, performance for the five data set partitions (*a*, *b*, …, *e*), based on SVM with linear kernel classifier, are presented (Table 5.5). One can see that power curve fitting based features enable us to achieve a recognition accuracy of 94.6% when we use data set partitioning strategy *e* and SVM with linear kernel classifier. These results are graphically shown in Figure 5.3.

Table 5.5: Recognition accuracy based on SVM with linear kernel for various feature extraction techniques

Strategy	Feature extraction techniques						
	Zoning	**Diagonal**	**Directional**	**Intersection and open end point**	**Transition**	**Parabola curve fitting**	**Power curve fitting**
a	64.9%	81.3%	77.4%	81.4%	58.1%	**72.0%**	**82.9%**
b	63.9%	82.2%	79.5%	81.5%	60.2%	**77.1%**	**84.4%**
c	66.2%	84.4%	79.6%	82.4%	62.1%	**77.7%**	**84.9%**
d	69.9%	87.3%	81.9%	86.7%	69.1%	**82.4%**	**88.1%**
e	73.7%	93.1%	89.7%	89.4%	72.0%	**83.7%**	**94.6%**

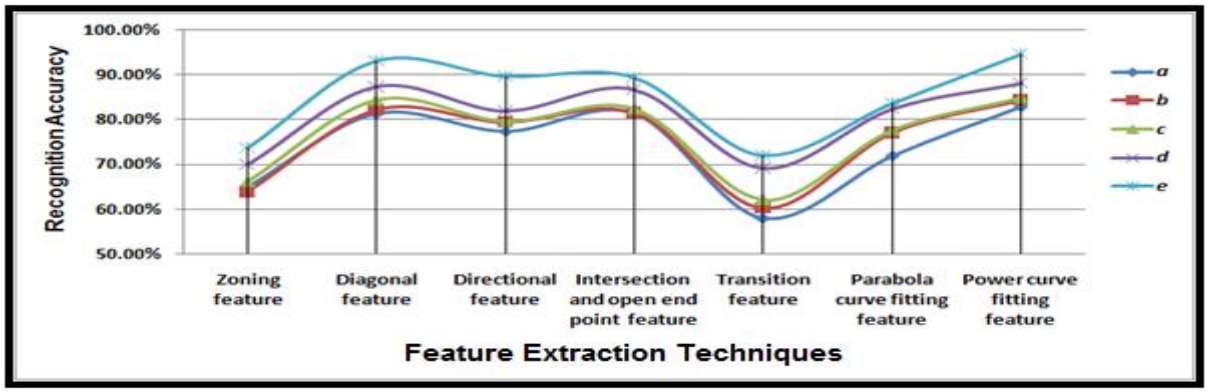


Figure 5.3: Recognition accuracy based on SVM with linear kernel classifier for various feature extraction techniques

5.3.3 Performance analysis based on SVM with polynomial kernel classifier

In this sub-section, performance for the five data set partitions (*a*, *b*, …, *e*), based on SVM with polynomial kernel classifier, are presented (Table 5.6). The degree of polynomial for SVM with polynomial kernel is 3. It has been observed that the power curve fitting based features make it possible to achieve a recognition accuracy of 94.0% when we use data set

strategy *e* and SVM with polynomial kernel classifier. These results are again graphically shown in Figure 5.4.

Table 5.6: Recognition accuracy based on SVM with polynomial kernel for various feature extraction techniques

Strategy	Feature extraction techniques						
	Zoning	**Diagonal**	**Directional**	**Intersection and open end point**	**Transition**	**Parabola curve fitting**	**Power curve fitting**
a	58.9%	76.1%	76.4%	74.2%	57.9%	**80.0%**	**84.2%**
b	59.1%	77.4%	77.4%	78.1%	59.3%	**81.6%**	**84.6%**
c	61.1%	79.7%	79.1%	78.1%	62.6%	**84.0%**	**85.9%**
d	68.1%	83.7%	83.7%	83.4%	69.7%	**86.1%**	**90.9%**
e	72.0%	90.0%	89.4%	84.0%	73.1%	**88.3%**	**94.0%**

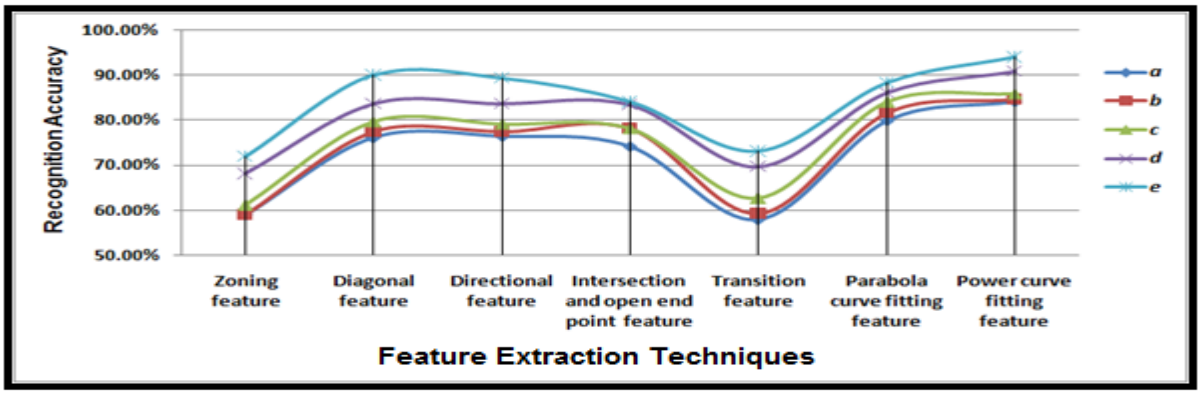


Figure 5.4: Recognition accuracy based on SVM with polynomial kernel classifier for various feature extraction techniques

5.3.4 Performance analysis based on SVM with RBF kernel classifier

In this sub-section, results on the five data set partitions (*a*, *b*, …, *e*), based on SVM with RBF kernel classifier are illustrated (Table 5.7). It has been seen that power curve fitting based features and SVM with RBF kernel classifier achieved a maximum recognition accuracy of 92.3% when we used data set partitioning strategy *e*. These results are graphically shown in Figure 5.5.

Table 5.7: Recognition accuracy based on SVM with RBF kernel for various feature extraction techniques

Strategy	Feature extraction techniques						
	Zoning	**Diagonal**	**Directional**	**Intersection and open end point**	**Transition**	**Parabola curve fitting**	**Power curve fitting**
a	64.9%	76.0%	54.0%	82.4%	58.1%	**66.9%**	**79.9%**
b	63.9%	78.2%	57.9%	84.2%	60.2%	**66.8%**	**81.6%**
c	66.2%	78.7%	60.0%	85.8%	62.1%	**68.3%**	**83.7%**
d	69.9%	84.0%	64.3%	88.7%	69.1%	**74.1%**	**87.4%**
e	73.7%	85.7%	69.1%	91.4%	72.0%	**75.7%**	**92.3%**

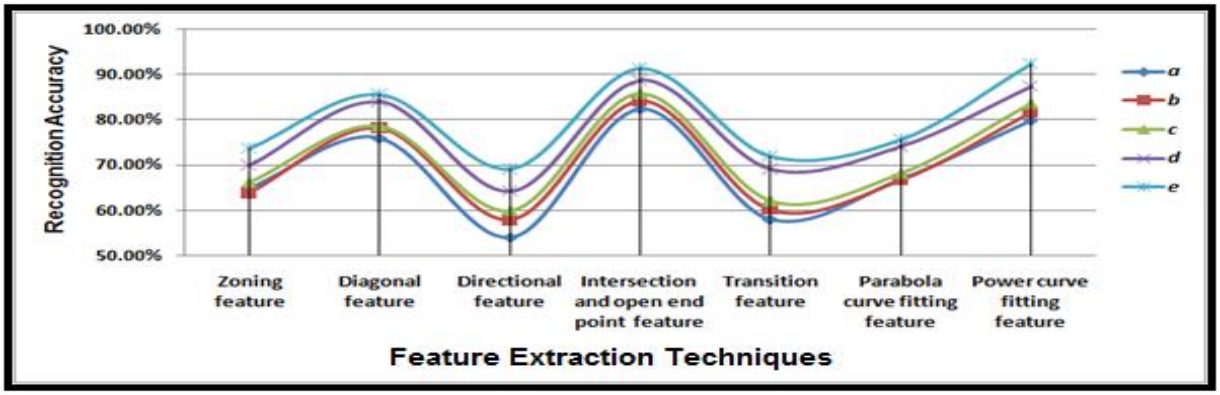


Figure 5.5: Recognition accuracy based on SVM with RBF kernel classifier for various feature extraction techniques

5.4 Discussion and conclusion

In this chapter, we have proposed two efficient feature extraction techniques, namely, parabola fitting based and power curve fitting based for offline handwritten *Gurmukhi* character recognition. The classifiers that have been employed in this study are *k*-NN; and SVM with three categories, namely, Linear-SVM, Polynomial-SVM and RBF-SVM. The system achieves maximum recognition accuracy of 97.9%, 94.6%, 94.0% and 92.3% using *k*-NN, Linear-SVM, Polynomial-SVM and RBF-SVM classifiers, respectively, when power curve fitting based features are used as inputs to the classification process.

It has also been observed that the results achieved using parabola fitting based features are better than recently used feature extraction techniques. Maximum recognition accuracy of 95.4% could be achieved when the parabola fitting based features were used with *k*-NN

classifier and data set partitioning strategy *e*. In the present work, the highest recognition accuracy of 97.9% could be achieved when the power curve fitting based features were used with *k*-NN classifier. In this case, 70% data was taken in training set and 30% data was considered in testing set (strategy *c*). As such, the results obtained using the power curve fitting based features are promising. This technique can further be explored by combining with other techniques for achieving higher recognition accuracy.

Chapter 6

Recognition of Offline Handwritten Gurmukhi Characters using *k*-fold Cross Validation

==========

In this chapter, we have presented an offline handwritten *Gurmukhi* character recognition system using *k*-fold cross validation technique. In general, *k*-fold cross validation technique divides a complete data set into *k* equal sub-sets. Then one sub-set is taken as testing data and remaining *k*-1 sub-sets are taken as training data. In this work, we have used various feature extraction techniques, namely, zoning features, diagonal features, directional features, intersection and open end points features, transition features, parabola curve fitting based features, power curve fitting based features, shadow features, centroid features, peak extent based features and modified division point based features. The peak extent based features and modified division point based features are the new features proposed in this work. For classification, we have considered *k*-NN, Linear-SVM, Polynomial-SVM and MLP classifier. We have used only two flavours of SVM, namely, Linear-SVM and Polynomial-SVM (with degree 3) in this chapter owing to the fact that these gave a reasonably good accuracy when used in previous chapter for measuring the performance of curve based features. This chapter is divided into six sections. Section 6.1 presents the shadow feature extraction technique, section 6.2 contains centroid feature extraction technique, section 6.3 describes peak extent based feature extraction technique and in section 6.4 we have illustrated modified division point based feature extraction technique. Experimental results are depicted in section 6.5 and in section 6.6 we have presented the conclusion of this chapter.

6.1 Shadow feature extraction technique

The lengths of projections of the character images, as shown in Figure 6.1 are considered to extract shadow features with the assistance of character images on the four sides of the minimal bounding boxes that enclose the character image (Basu *et al.*, 2009). Each of the respective values of the shadow feature is divided by the maximum possible length of the projections on each side that has been extracted and that needs to be normalized. The profile counts the number of pixels between the edge of the character and the bounding box of the character image. Shadow features illustrate well the exterior drawing of characters and allow uniqueness between a number of confusing characters, such as "ਬ" and "ਖ".

(a) **(b)**

Figure 6.1 Shadow features: (a) *Gurmukhi* character ("ਬ"), (b) *Gurmukhi* character ("ਖ").

The steps that have been used to extract these features are given below.

Step I: Input the character image of 100×100 size.

Step II: Calculate the length of projections of white pixels of the character image on all the four sides *i.e.* top, bottom, left and right as shown in Figure 6.1.

Step III: Calculate the projection profile as number of background pixels between the edge of the character and bounding box of the character image.

Step IV: Normalize the values of feature vector by dividing each element of the feature vector by the largest value in the feature vector.

These steps yield a feature set with 400 elements.

`

6.2 Centroid feature extraction technique

Centroid is the point that can be considered as the center of a two-dimensional image. Coordinates of the centroid of foreground pixels in each zone of a character image can also be considered as features (Basu *et al*., 2009).

The following steps have been implemented for extracting these features.

Step I: Divide the bitmap image into *n* (=100) number of zones, each of size 10×10 pixels.

Step II: Find the coordinates of foreground pixels in each zone.

Step III: Calculate the centroid of these foreground pixels and store the coordinates of centroid as a feature value.

Step IV: Corresponding to the zones that do not have a foreground pixel, take the feature value as zero.

These steps give a feature set with 2*n* elements.

6.3 Peak extent based feature extraction technique (Proposed method III)

In this chapter, we have proposed a technique for feature extraction, namely, peak extent based feature. The peak extent based feature is extracted by taking into consideration the sum of the peak extents that fit successive black pixels along each zone, as shown in Figure 6.2 (a-c). Peak extent based features can be extracted horizontally and vertically. In the horizontal peak extent features, we consider the sum of the peak extents that fit successive black pixels horizontally in each row of a zone as shown in Figure 6.2 (b), whereas in vertical peak extent features we consider the sum of the peak extents that fit successive black pixels vertically in each column of a zone as depicted in Figure 6.2 (c).

The steps that have been used to extract these features are given below.

Step I: Divide the bitmap image into *n* (=100) number of zones, each of size 10×10 pixels.

Step II: Find the peak extent as sum of successive foreground pixels in each row of a zone.

Step III: Replace the values of successive foreground pixels by peak extent value, in each row of a zone.

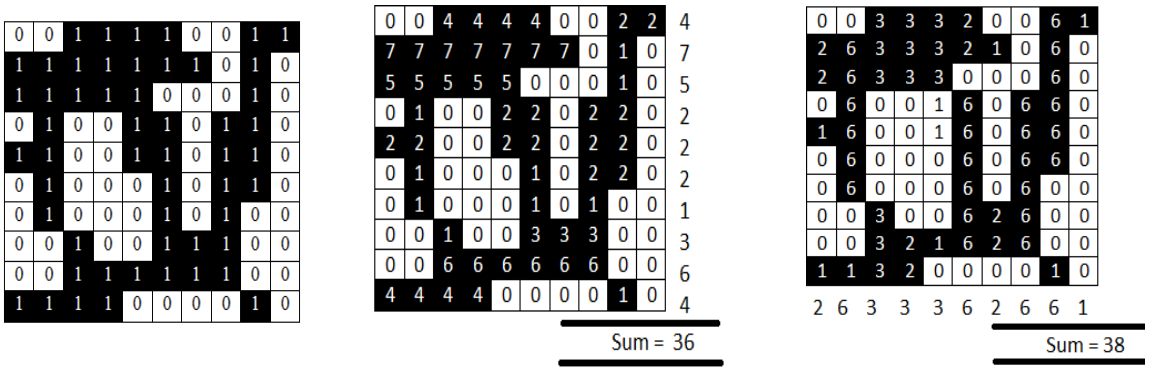
Step IV: Find the largest value of peak extent in each row. As such, each zone has 10 horizontal peak extent features (Figure 6.2(b)).

Step V: Obtain the sum of these 10 peak extent sub-feature values for each zone and consider this as a feature for corresponding zone.

Step VI: For the zones that do not have a foreground pixel, take the feature value as zero.

Step VII: Normalize the values in feature vector by dividing each element of the feature vector by the largest value in the feature vector.

Similarly, for vertical peak extent features, we have considered the sum of the lengths of the peak extents in each column of each zone as shown in Figure 6.2 (c). These steps will give a feature set with 2*n* elements.



(a) (b) (c)

Figure 6.2 Peak extent based features: (a) Zone of bitmap image, (b) Horizontally peak extent based features, (c) Vertically peak extent based features.

6.4 Modified division points based feature extraction technique (Proposed method IV)

In this section, we have presented Modified Division Points (MDP) based feature extraction technique. In this technique, initially, we have divided the character image into *n* (=100) zones, each of size 10×10 pixels. Let Img(x, y) be the character image having 1's representing foreground pixels and 0's representing background pixels. The proposed methodology is based on sub-parts of the character image so that the resulting sub-parts have balanced numbers of foreground pixels. Let $V_p[xmax]$ be the vertical projection and $H_p[ymax]$ be the horizontal projection of the particular zone Z_1 as shown in Figure 6.3.

`

$$V_p = [4, 7, 6, 5, 6, 8, 3, 6, 7, 1]$$

$$H_p = [6, 8, 6, 5, 6, 4, 3, 4, 6, 5]$$

Here, in V_p the division point (d_v) of array is 5 (fifth element), because sum of the left sub-array elements and sum of the right sub-array elements is balanced as far as possible, if we consider fifth element into the left sub-array. Similarly, we have calculated the division point (d_h) of H_p which is taken as 4 (fourth element). The values of division points d_v and d_h of each zone are stored as features in the feature vector.

0	0	1	1	1	1	0	0	1	1
1	1	1	1	1	1	1	0	1	0
1	1	1	1	1	0	0	0	1	0
0	1	0	0	1	1	0	1	1	0
1	1	0	0	1	1	0	1	1	0
0	1	0	0	0	1	0	1	1	0
0	1	0	0	0	1	0	1	0	0
0	0	1	0	0	1	1	1	0	0
0	0	1	1	1	1	1	1	0	0
1	1	1	1	0	0	0	0	1	0

Figure 6.3: Bitmap of zone Z_1

The steps that have been used to extract these features are given below:

Step I: Divide the bitmap image into *n* (=100) number of zones, each of size 10×10 pixels.

Step II: Find the horizontal projection profiles H_p and vertical projection profiles V_p in each zone of a bitmap image.

Step III: Store the horizontal projection profiles values in array *H* and vertical projection profiles values in array *V*.

Step IV: After that, calculate the value of division point (d_h) of array *H* and division point (d_v) of array *V* based on sub-parts of the arrays so that the resulting sub-arrays have balanced numbers of foreground pixels.

Step V: Consider the values of (d_h) and (d_v) in left sub-array for make the possible balance between left sub-array and right sub-array.

Step VI: Calculate the values of (d_h) and (d_v) for each zone and placed in the corresponding zone as its feature.

Step VII: Corresponding to the zones that do not have a foreground pixel, the feature value is taken as zero.

Step VIII: Normalize the values of feature vector by dividing each element of the feature vector by the largest value in the feature vector.

These steps give a feature set with 2*n* elements.

6.5 Experimental results and comparisons with recently used feature extraction techniques

In this section, the results of recognition system for offline handwritten *Gurmukhi* characters with *k*-fold cross validation are presented. In the experimentation work in this thesis, we have considered four example values of *k* (= 3, 4, 5 and 10) in the *k*-fold cross validation. However, we have reported the results for *k* = 5 in this chapter as this yielded the highest accuracy for different classifiers. The recognition results are further based on various feature extraction techniques, namely, shadow features, centroid features, peak extent based features and modified division point based features. Comparison between these feature extraction techniques and other recently used feature extraction techniques, namely, zoning features, diagonal features, directional features, intersection and open end points features and transition features has also been presented. Classifiers, namely, *k*-NN, Linear-SVM, Polynomial-SVM and MLP have been considered in this work for recognition purpose and in order to compare the recognition results of the proposed feature extraction techniques. For the present work, we have used 5,600 samples of isolated offline handwritten *Gurmukhi* characters written by one hundred different writers.

Classifier-wise experimental results of testing are presented in following sub-sections.

6.5.1 Recognition results based on *k*-NN classifier

In this sub-section, experimental results based on *k*-NN classifier are presented. It has been seen that peak extent features, with *k*-NN classifier, achieved an average recognition accuracy of 95.5%. Recognition results based on *k*-NN classifier are depicted in Table 6.1.

`

Table 6.1: Recognition results based on *k*-NN classifier

***k*-fold cross validation**	**Feature Extraction Techniques**										
	Zoning	**Diagonal**	**Directional**	**Transition**	**Intersection**	**Parabola curve fitting**	**Power curve fitting**	**Shadow**	**Centroid**	**Modified division point**	**Peak extent**
Fold 1	70.9%	86.7%	70.7%	89.3%	77.1%	71.9%	68.4%	71.6%	90.0%	91.1%	96.0%
Fold 2	69.4%	86.1%	74.6%	83.1%	88.3%	91.7%	98.2%	69.4%	89.0%	88.7%	96.3%
Fold 3	76.1%	88.0%	77.0%	89.0%	79.4%	92.3%	98.7%	76.0%	94.7%	94.7%	96.6%
Fold 4	63.7%	79.3%	67.7%	78.1%	82.9%	69.9%	67.0%	69.9%	87.4%	81.1%	92.0%
Fold 5	71.0%	88.1%	72.6%	82.4%	72.1%	69.1%	65.0%	73.1%	89.3%	90.3%	96.6%
Average	70.2%	85.6%	72.5%	84.4%	79.9%	78.9%	79.5%	72.0%	90.1%	89.2%	95.5%

6.5.2 Recognition results based on Linear-SVM classifier

In this sub-section, recognition results of Linear-SVM classifier are presented. Using this classifier, we have achieved an average recognition accuracy of 95.6% with proposed peak extent based feature extraction technique. The recognition results of different features considered under this work are given in Table 6.2.

Table 6.2: Recognition results based on Linear-SVM classifier

***k*-fold cross validation**	**Feature Extraction Techniques**										
	Zoning	**Diagonal**	**Directional**	**Transition**	**Intersection**	**Parabola curve fitting**	**Power curve fitting**	**Shadow**	**Centroid**	**Modified division point**	**Peak extent**
Fold 1	63.6%	77.9%	53.3%	54.0%	61.9%	52.7%	48.6%	75.7%	94.9%	84.3%	95.6%
Fold 2	69.7%	78.9%	55.3%	56.6%	67.7%	76.6%	80.3%	84.7%	98.3%	85.1%	98.1%
Fold 3	66.1%	83.9%	56.0%	55.7%	64.7%	68.6%	79.1%	79.4%	98.1%	86.4%	96.6%
Fold 4	60.4%	72.0%	51.0%	60.2%	58.1%	54.6%	50.0%	79.3%	85.4%	79.9%	91.3%
Fold 5	70.0%	81.0%	60.0%	62.0%	70.3%	60.3%	57.9%	90.1%	90.4%	87.1%	96.6%
Average	66.0%	78.7%	55.1%	57.7%	64.5%	62.5%	63.2%	81.8%	93.4%	84.6%	95.6%

6.5.3 Recognition results based on Polynomial-SVM classifier

In this sub-section, recognition results of Polynomial-SVM classifier are presented. Using this classifier, we have achieved an average recognition accuracy of 92.4% with proposed

peak extent based feature extraction technique. The recognition results of different features considered under this work are given in Table 6.3.

Table 6.3: Recognition results based on Polynomial-SVM classifier

***k*-fold cross validation**	**Feature Extraction Techniques**										
	Zoning	**Diagonal**	**Directional**	**Transition**	**Intersection**	**Parabola curve fitting**	**Power curve fitting**	**Shadow**	**Centroid**	**Modified division point**	**Peak extent**
Fold 1	62.9%	72.2%	51.1%	52.4%	62.9%	51.2%	52.3%	72.1%	87.2%	82.2%	91.2%
Fold 2	68.6%	73.2%	52.2%	54.4%	65.3%	78.1%	81.5%	79.6%	85.5%	83.5%	94.4%
Fold 3	67.1%	81.8%	54.7%	54.2%	61.3%	58.3%	78.1%	81.5%	89.1%	87.1%	92.8%
Fold 4	58.2%	68.2%	48.3%	58.7%	52.3%	56.2%	65.1%	72.2%	81.2%	74.5%	90.2%
Fold 5	66.1%	74.6%	57.2%	67.1%	67.1%	64.9%	49.2%	82.2%	90.1%	82.4%	93.5%
Average	64.6%	73.9%	52.7%	57.4%	61.8%	61.7%	65.2%	77.5%	86.6%	81.9%	92.4%

6.5.4 Recognition results based on MLP classifier

In this sub-section, we have presented recognition results of different features considered in this work based on MLP classifier. Using this classifier, we have achieved an average recognition accuracy of 94.7% with proposed peak extent based feature extraction technique. The recognition results of different features are given in Table 6.4.

Table 6.4: Recognition results based on MLP classifier

***k*-fold cross validation**	**Feature Extraction Techniques**										
	Zoning	**Diagonal**	**Directional**	**Transition**	**Intersection**	**Parabola curve fitting**	**Power curve fitting**	**Shadow**	**Centroid**	**Modified division point**	**Peak extent**
Fold 1	86.0%	81.1%	53.1%	79.6%	76.1%	61.0%	60.4%	73.3%	91.0%	86.1%	94.3%
Fold 2	84.7%	82.0%	56.6%	67.7%	63.7%	60.0%	59.4%	71.1%	93.8%	86.9%	96.3%
Fold 3	85.1%	80.0%	56.1%	80.9%	71.0%	61.3%	68.2%	72.6%	82.6%	88.1%	96.3%
Fold 4	84.1%	80.9%	52.4%	65.1%	77.9%	59.6%	59.6%	70.6%	82.1%	84.7%	91.2%
Fold 5	59.6%	80.3%	56.3%	78.7%	67.7%	68.6%	59.3%	72.0%	93.3%	83.4%	95.5%
Average	79.9%	80.9%	54.9%	74.4%	71.3%	62.1%	61.4%	71.9%	88.6%	85.8%	94.7%

`

6.6 Discussions and conclusion

The work presented in this chapter proposes an offline handwritten *Gurmukhi* character recognition system using *k*-fold cross validation technique. The classifiers that have been employed in this work are *k*-NN, Linear-SVM, Polynomial-SVM and MLP. We have used 5600 samples of isolated offline handwritten *Gurmukhi* characters in this study. Two new feature extraction techniques have also been proposed in this chapter. We conclude that peak extent based features are preeminent features as compared to other feature extraction techniques. As depicted in Table 6.5, we could achieve a 5-fold cross validation accuracy with peak extent based features as 95.6%, 92.4%, 95.5% and 94.7% with Linear-SVM, Polynomial-SVM, *k*-NN and MLP classifier, respectively.

Table 6.5: Recognition results based on 5-fold cross validation technique with peak extent based features

Classifier	Recognition Accuracy
Linear-SVM	95.6%
Polynomial-SVM	92.4%
k-NN	95.5%
MLP	94.7%

Chapter 7

PCA Based Analysis and Hierarchical Feature Extraction for Offline Handwritten Gurmukhi Character Recognition System

==========

Principal Component Analysis (PCA) has widely been used for extracting representative features for pattern recognition and has also been used to reduce the dimension of data (Sundaram and Ramakarishnan (2008), Deepu *et al.* (2004)). In the present work, we have explored this technique for the process of recognizing offline handwritten *Gurmukhi* characters and a technique for offline handwritten *Gurmukhi* character recognition based on PCA is presented. The system first prepares a skeleton of the character so that meaningful feature information about the character can be extracted. PCA is then applied to these features for finding the linear combination of relevant features. These combinations are then inputted to classification process. For classification, we have used *k*-NN, Linear-SVM, Polynomial-SVM and RBF-SVM based approaches and also combinations of these approaches. This chapter is divided into four sections. Section 7.1 introduces the concepts of PCA, section 7.2 presents the experimental results based on PCA and section 7.3 presents a hierarchical feature extraction technique for offline handwritten character recognition and section 7.4 concludes the chapter.

7.1 Principal component analysis

PCA is the method that is used to identify correlation among a set of variables for the purpose of data reduction. This powerful exploratory method provides insightful graphical

summaries with an ability to include additional information as well. There are various applications of PCA, namely, summarizing of large sets of data, identifying structure, identifying redundancy, and producing insightful graphical displays of the results.

In pattern recognition field, PCA is used as a mathematical procedure that employs a transformation to convert a set of observations of, possibly, correlated features into a set of values of un-correlated features called as principal components. PCA is a well-established technique for extracting representative features for character recognition and is used to reduce the dimensions of the data. The technique is useful when a large number of variables prohibit effective interpretation of the relationships between different features. By reducing the dimensionality, one can interpret from a few features rather than a large number of features. The number of principal components is generally less than the number of original variables. By selecting top j eigen vectors with larger eigen values for subspace approximation, PCA can provide a lower dimension representation to expose the underlying structures of the complex data sets. Let there be P features for handwritten character recognition. In the next step, the symmetric matrix S of covariance between these features is calculated. Now, the eigen vectors U_i $(i = 1, 2, \dots, P)$ and the corresponding eigen values Δ_i $(i = 1, 2, \dots, P)$ are calculated. From these P eigen vectors only j eigen vectors are chosen, corresponding to the larger eigen values. An eigen vector, corresponding to higher eigen value, describes more characteristic features of a character. Using these j eigen vectors, feature extraction is done using PCA. **In the present work, twelve features for a *Gurmukhi* character have been considered and the experiments have been conducted by taking 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 and 12 principal components extracted with SPSS software tool.** In the next section, the results of these experimentation are presented.

7.2 Experimental results and discussion

In this section, results of offline handwritten *Gurmukhi* character recognition system using PCA are presented. The features that have been considered are zoning features, diagonal features, directional features, transition features, intersection and open end points features, parabola curve fitting based features, power curve fitting based features, shadow features,

centroid features, peak extent based features and modified division points based features. The recognition results are obtained for four classifiers, namely, *k*-NN, Linear-SVM, Polynomial-SVM and RBF-SVM. We have also used combinations of the output of each classifier in parallel, and recognition is done on the basis of the voting scheme. We have considered the following combinations of classifiers:

LPR (Linear-SVM + Polynomial-SVM + RBF-SVM),
PRK (Polynomial-SVM + RBF-SVM + *k*-NN),
LRK (Linear-SVM + RBF-SVM + *k*-NN) and
LPK (Linear-SVM + Polynomial-SVM + *k*-NN)

In this work, we have divided the data set of each category using five partitioning strategies (*a*, *b*, *c*, *d* and *e*) as given in Table 5.3. We have also experimented with 5-fold cross validation technique for partitioning of training and testing data set.

Category-wise results of a recognition system based on PCA are presented in the following sub-sections.

7.2.1 Recognition accuracy for *category* 1 samples

In this section, we have considered each *Gurmukhi* character written one hundred times by a single writer. For the sake of comparisons between the performance of principal components, two principal components (2-PC), three principal components (3-PC), …, twelve principal components (12-PC) have been considered to be taken as input to the classifiers. Partitioning strategy-wise and 5-fold cross validation technique based experimental results of testing are presented in the following sub-sections.

7.2.1.1 Recognition accuracy using strategy *a*

In this sub-section, classifier wise recognition results of partitioning strategy *a* have been presented. LRK is the best classifiers combination for offline handwritten *Gurmukhi* character

`

recognition when this strategy is followed. Maximum accuracy of 98.9% could be achieved for this strategy. Recognition results of classifiers and their combinations are given in Table 7.1 for twelve features (12-feature) and twelve principal components.

Table 7.1: Classifier wise recognition accuracy for *category* 1 samples with strategy *a*

Principal Components	Linear -SVM	Poly.- SVM	RBF- SVM	*k*-NN	LPR	PRK	LRK	LPK	Average
2-PC	94.0%	95.4%	95.4%	94.7%	97.5%	98.2%	98.2%	98.3%	96.5%
3-PC	92.9%	91.3%	93.4%	97.7%	96.7%	97.7%	98.3%	97.9%	95.8%
4-PC	92.8%	69.7%	91.1%	93.2%	96.0%	96.7%	**98.9%**	97.5%	91.9%
5-PC	93.6%	80.5%	92.5%	91.9%	96.5%	96.5%	97.9%	97.0%	93.3%
6-PC	94.4%	89.1%	93.8%	84.7%	97.0%	97.9%	98.2%	98.1%	94.2%
7-PC	94.6%	83.7%	94.3%	82.6%	97.7%	98.3%	98.8%	98.2%	93.5%
8-PC	94.3%	91.1%	94.9%	83.6%	92.8%	97.8%	97.8%	98.2%	93.8%
9-PC	92.8%	92.5%	94.9%	88.7%	93.6%	96.4%	98.3%	98.3%	94.4%
10-PC	93.7%	84.7%	95.2%	92.8%	95.1%	90.1%	98.8%	98.6%	93.6%
11-PC	93.9%	82.6%	91.9%	94.7%	89.2%	98.2%	98.6%	98.5%	93.5%
12-PC	93.7%	86.4%	89.0%	94.8%	86.4%	98.5%	98.3%	95.1%	92.9%
12-Features	94.0%	95.2%	17.8%	70.3%	97.3%	89.0%	88.7%	97.4%	81.2%
Average	93.7%	86.8%	87.0%	89.1%	94.7%	96.3%	97.6%	97.8%	92.9%

7.2.1.2 Recognition accuracy using strategy *b*

We achieved an accuracy of 99.6% with strategy *b* and we have seen that LPR is the best classifiers combination for offline handwritten *Gurmukhi* character recognition for this strategy. Recognition results for twelve features (12-feature) and twelve principal components of partitioning strategy *b* are depicted in Table 7.2.

7.2.1.3 Recognition accuracy using strategy *c*

In partitioning strategy *c*, the maximum accuracy that could be achieved is 99.5%. Using this strategy, we have seen once again that LPR is the best classifiers combination for offline

handwritten *Gurmukhi* character recognition. Recognition results of this partitioning strategy, for twelve features (12-feature) and twelve principal components, are given in Table 7.3.

Table 7.2: Classifier wise recognition accuracy for *category* 1 samples with strategy *b*

Principal Components	Linear -SVM	Poly.- SVM	RBF- SVM	*k*-NN	LPR	PRK	LRK	LPK	Average
2-PC	95.1%	95.4%	95.4%	97.6%	98.2%	98.5%	98.4%	98.9%	97.2%
3-PC	94.9%	93.6%	94.4%	97.4%	98.0%	98.2%	98.7%	98.8%	96.7%
4-PC	95.1%	79.4%	92.6%	96.0%	97.6%	97.5%	98.7%	98.2%	94.4%
5-PC	94.9%	86.6%	93.4%	95.4%	97.7%	98.2%	98.8%	98.3%	95.4%
6-PC	95.7%	91.9%	94.7%	89.2%	98.2%	98.6%	98.9%	98.6%	95.7%
7-PC	95.5%	89.0%	94.8%	86.4%	98.5%	98.9%	99.2%	98.5%	95.1%
8-PC	93.0%	97.4%	98.4%	98.6%	99.4%	93.4%	95.4%	97.7%	96.7%
9-PC	93.9%	92.6%	99.0%	99.3%	**99.6%**	94.7%	89.2%	98.2%	95.8%
10-PC	95.2%	92.8%	95.1%	90.1%	92.8%	97.8%	97.8%	97.9%	94.9%
11-PC	95.2%	90.5%	94.5%	86.5%	93.6%	96.4%	98.3%	98.9%	94.2%
12-PC	92.9%	94.5%	98.5%	93.4%	96.0%	98.8%	95.4%	95.1%	95.6%
12-Features	95.8%	92.2%	20.3%	73.9%	98.1%	88.2%	88.2%	98.9%	81.9%
Average	94.8%	91.3%	89.2%	91.9%	97.3%	96.6%	96.4%	98.2%	94.5%

Table 7.3: Classifier wise recognition accuracy for *category* 1 samples with strategy *c*

Principal Components	Linear -SVM	Poly.- SVM	RBF- SVM	*k*-NN	LPR	PRK	LRK	LPK	Average
2-PC	95.9%	95.2%	95.1%	97.4%	98.7%	99.2%	99.2%	99.0%	97.5%
3-PC	95.1%	93.7%	94.3%	97.3%	98.6%	98.7%	99.1%	99.0%	97.0%
4-PC	94.9%	83.6%	92.8%	97.8%	97.8%	97.9%	99.0%	99.5%	95.4%
5-PC	94.9%	88.7%	93.6%	96.4%	98.3%	98.9%	99.2%	98.9%	96.1%
6-PC	95.2%	92.8%	98.8%	90.1%	98.8%	99.4%	99.5%	99.2%	96.7%
7-PC	95.2%	90.5%	98.9%	86.5%	98.9%	99.0%	99.4%	99.0%	95.9%
8-PC	92.8%	97.8%	97.8%	99.4%	**99.5%**	99.2%	93.6%	96.4%	97.1%
9-PC	93.6%	96.4%	98.9%	99.0%	99.4%	99.0%	95.1%	90.1%	96.4%
10-PC	95.1%	90.5%	94.3%	97.3%	97.8%	97.9%	99.0%	99.5%	96.4%
11-PC	94.9%	97.8%	92.8%	97.8%	98.3%	98.9%	99.2%	98.9%	97.3%
12-PC	92.8%	90.1%	98.6%	97.8%	99.2%	94.3%	94.9%	95.2%	95.4%
12-Features	95.1%	85.8%	25.5%	77.5%	98.0%	92.1%	91.2%	98.9%	83.0%
Average	94.6%	91.9%	90.1%	94.5%	98.6%	97.9%	97.4%	97.8%	95.4%

`

7.2.1.4 Recognition accuracy using strategy *d*

In this sub-section, recognition results using partitioning strategy *d* are presented. Using this strategy, we have achieved maximum recognition accuracy of 99.6% with LRK classifiers combination. Recognition results for the features and the principal components under consideration, using this strategy, are illustrated in Table 7.4.

Table 7.4: Classifier wise recognition accuracy for *category* 1 samples with strategy *d*

Principal Components	Linear -SVM	Poly.- SVM	RBF- SVM	*k*-NN	LPR	PRK	LRK	LPK	Average
2-PC	94.0%	94.0%	94.0%	94.7%	99.1%	99.6%	**99.6%**	99.4%	96.8%
3-PC	94.1%	92.4%	93.3%	96.1%	98.9%	99.1%	99.3%	99.3%	96.6%
4-PC	93.9%	85.3%	92.2%	97.1%	98.1%	98.1%	99.1%	98.6%	95.3%
5-PC	93.7%	87.3%	93.0%	97.4%	98.4%	98.6%	99.4%	98.9%	95.8%
6-PC	93.9%	92.2%	93.9%	92.6%	99.0%	99.3%	99.6%	99.0%	96.2%
7-PC	93.7%	90.4%	87.3%	93.0%	99.1%	99.1%	99.4%	99.1%	95.2%
8-PC	93.0%	97.4%	92.2%	93.9%	99.4%	99.1%	99.3%	98.6%	96.6%
9-PC	93.9%	92.6%	93.3%	96.1%	98.0%	99.1%	99.6%	98.4%	96.5%
10-PC	92.2%	93.9%	98.1%	93.0%	99.1%	98.6%	99.4%	99.1%	96.7%
11-PC	92.6%	99.0%	93.9%	93.9%	99.4%	99.3%	98.6%	99.4%	97.0%
12-PC	93.0%	99.1%	87.3%	94.7%	99.3%	99.1%	98.9%	98.9%	96.3%
12-Features	93.1%	94.4%	36.5%	76.4%	98.6%	92.7%	92.7%	99.0%	85.4%
Average	93.4%	93.2%	87.9%	93.3%	98.9%	98.5%	98.8%	98.9%	95.4%

7.2.1.5 Recognition accuracy using strategy *e*

In this sub-section, classifier wise recognition results of partitioning strategy *e* have been presented. LRK is the best classifiers combination when we follow this strategy. For the features and the principal components under consideration, maximum recognition accuracy of 99.9% could be achieved. Recognition results of different classifiers and their combinations for twelve features (12-feature) and twelve principal components are given in Table 7.5.

Table 7.5: Classifier wise recognition accuracy for *category* 1 samples with strategy *e*

Principal Components	Linear -SVM	Poly.- SVM	RBF- SVM	*k*-NN	LPR	PRK	LRK	LPK	Average
2-PC	89.5%	89.5%	89.7%	96.4%	99.4%	99.4%	99.3%	99.4%	95.3%
3-PC	89.5%	87.2%	89.5%	97.7%	99.1%	99.7%	99.6%	99.4%	95.2%
4-PC	89.5%	80.6%	88.3%	96.6%	99.9%	98.6%	99.6%	99.1%	94.0%
5-PC	89.5%	84.9%	88.6%	97.1%	99.1%	99.1%	99.1%	99.4%	94.6%
6-PC	89.5%	87.7%	98.7%	88.6%	99.7%	99.2%	99.4%	99.7%	95.3%
7-PC	89.5%	86.0%	90.0%	79.1%	99.7%	99.7%	99.6%	99.4%	92.9%
8-PC	83.6%	85.1%	87.1%	88.6%	98.6%	99.1%	99.4%	98.6%	92.5%
9-PC	88.7%	80.6%	88.3%	99.1%	99.1%	96.6%	**99.9%**	98.7%	93.9%
10-PC	92.8%	84.9%	88.6%	99.7%	99.6%	97.1%	99.1%	98.7%	95.1%
11-PC	90.5%	87.7%	91.1%	88.6%	99.7%	99.4%	96.4%	93.6%	93.4%
12-PC	90.0%	89.1%	92.1%	92.2%	98.6%	97.1%	99.4%	99.1%	94.7%
12-Features	89.5%	89.5%	70.1%	69.7%	99.7%	96.6%	96.6%	99.7%	88.9%
Average	89.3%	86.1%	88.5%	91.1%	99.4%	98.5%	99.0%	98.8%	93.8%

7.2.1.6 Recognition accuracy using 5-fold cross validation technique

Table 7.6: Classifier wise recognition accuracy for *category* 1 samples with 5-fold cross validation technique

Principal Components	Linear -SVM	Poly.- SVM	RBF- SVM	*k*-NN	LPR	PRK	LRK	LPK	Average
2-PC	91.8%	92.0%	92.0%	94.2%	96.6%	97.0%	97.0%	97.0%	94.7%
3-PC	91.4%	89.8%	91.1%	95.3%	96.3%	96.7%	97.0%	96.9%	94.3%
4-PC	91.4%	78.1%	89.6%	94.2%	95.9%	95.8%	97.1%	96.6%	92.3%
5-PC	91.5%	83.9%	90.4%	93.7%	96.0%	96.3%	96.9%	96.5%	93.1%
6-PC	91.9%	88.9%	94.1%	87.3%	96.6%	96.9%	97.1%	96.9%	93.7%
7-PC	91.8%	86.2%	91.2%	83.8%	96.8%	97.0%	**97.3%**	96.9%	92.6%
8-PC	89.5%	91.9%	92.2%	91.0%	96.0%	95.8%	95.2%	95.9%	93.4%
9-PC	90.7%	89.1%	93.0%	94.5%	96.0 %	95.2%	94.5%	94.8%	93.5%
10-PC	91.9%	87.6%	92.4%	92.7%	94.9%	94.4%	96.8%	96.8%	93.4%
11-PC	91.6%	89.7%	91.0%	90.5%	94.1%	96.5%	96.3%	95.9%	93.2%
12-PC	90.6%	90.0%	91.2%	92.7%	94.0%	95.6%	95.4%	94.7%	93.1%
12-Features	91.6%	89.6%	33.4%	72.1%	96.4%	89.9%	89.7%	96.8%	82.4%
Average	91.3%	88.1%	86.8%	90.1%	95.8%	95.6%	95.9%	96.3%	92.5%

`

Here, we have seen that, LRK is the best classifier combination when we follow 5-fold cross validation technique for training and testing dataset partitioning. Maximum recognition accuracy of 97.3% could be achieved with this technique. Recognition results of different classifiers and their combinations for twelve features (12-feature) and twelve principal components are given in Table 7.6.

7.2.2 Recognition accuracy for *category* 2 samples

In this section, we have considered each *Gurmukhi* character written ten times by ten different writers. The principal components, two principal components (2-PC), three principal components (3-PC), …, twelve principal components (12-PC) have been considered to be taken as input to the classifiers. Dataset partitioning strategy-wise and 5-fold cross validation technique based experimental results are presented in the following sub-sections.

7.2.2.1 Recognition accuracy using strategy *a*

In this sub-section, classifier wise recognition results of partitioning strategy *a* have been presented. When we consider this strategy, then *k*-NN is the best classifier for offline handwritten *Gurmukhi* character recognition. The maximum accuracy that could be achieved is 94.5% for this strategy. Recognition results of different classifiers and their combinations are given in Table 7.7.

7.2.2.2 Recognition accuracy using strategy *b*

In partitioning strategy *b*, the maximum accuracy that could be achieved is 94.5%. Using this strategy, we have again observed that *k*-NN is the best classifier for offline handwritten *Gurmukhi* character recognition. Recognition results of this partitioning strategy, for twelve features (12-feature) and twelve principal components, are depicted in Table 7.8.

Table 7.7: Classifier wise recognition accuracy for *category* 2 samples with strategy *a*

Principal Components	Linear -SVM	Poly.- SVM	RBF- SVM	*k*-NN	LPR	PRK	LRK	LPK	Average
2-PC	77.8%	75.9%	80.3%	91.4%	83.1%	87.0%	85.8%	88.4%	83.7%
3-PC	76.6%	53.7%	75.3%	**94.5%**	82.1%	86.2%	84.2%	87.9%	80.1%
4-PC	74.8%	25.6%	73.0%	83.7%	79.4%	85.9%	83.6%	87.4%	74.2%
5-PC	75.6%	33.8%	75.7%	75.8%	80.8%	88.0%	83.9%	88.8%	75.3%
6-PC	79.6%	41.9%	78.6%	69.8%	84.1%	90.8%	86.3%	91.1%	77.8%
7-PC	81.4%	45.3%	79.7%	71.4%	85.0%	88.6%	84.1%	90.3%	78.2%
8-PC	77.4%	42.4%	84.3%	84.1%	84.2%	85.3%	86.2%	90.2%	79.3%
9-PC	75.3%	50.0%	50.0%	81.4%	83.6%	84.3%	81.4%	91.5%	74.7%
10-PC	73.0%	59.7%	59.7%	83.9%	86.3%	87.0%	83.9%	89.7%	77.9%
11-PC	75.9%	67.1%	80.8%	86.2%	83.9%	73.3%	84.5%	89.5%	80.1%
12-PC	75.7%	51.3%	79.4%	85.9%	87.3%	83.9%	86.6%	89.7%	80.0%
12-Features	76.5%	53.7%	15.1%	60.5%	84.2%	86.6%	85.7%	85.4%	68.5%
Average	76.6%	50.0%	69.3%	80.7%	83.7%	85.6%	84.7%	89.2%	77.5%

Table 7.8: Classifier wise recognition accuracy for *category* 2 samples with strategy *b*

Principal Components	Linear -SVM	Poly.- SVM	RBF- SVM	*k*-NN	LPR	PRK	LRK	LPK	Average
2-PC	56.2%	55.7%	57.1%	93.1%	86.3%	89.6%	88.8%	91.1%	77.2%
3-PC	79.2%	62.0%	79.4%	**94.5%**	84.1%	88.8%	86.4%	90.2%	83.1%
4-PC	78.4%	34.3%	77.4%	86.2%	84.3%	91.9%	86.6%	92.8%	79.0%
5-PC	79.2%	40.8%	79.7%	77.9%	85.3%	89.5%	87.3%	88.1%	78.5%
6-PC	82.5%	51.3%	81.4%	73.3%	87.4%	90.6%	88.9%	91.5%	80.9%
7-PC	83.9%	56.8%	83.9%	73.0%	87.4%	91.1%	89.7%	93.1%	82.4%
8-PC	82.7%	42.4%	73.3%	77.9%	82.1%	88.6%	87.1%	89.5%	77.9%
9-PC	82.2%	50.0%	62.0%	73.3%	83.2%	87.3%	85.3%	89.7%	76.6%
10-PC	82.4%	42.4%	77.9%	81.4%	82.2%	87.3%	84.2%	81.1%	77.4%
11-PC	81.8%	45.3%	73.3%	81.1%	84.2%	87.1%	84.2%	82.2%	77.4%
12-PC	82.2%	53.7%	56.8%	82.2%	84.2%	85.3%	83.2%	87.1%	76.8%
12-Features	80.0%	82.3%	17.9%	59.9%	87.1%	87.5%	86.7%	88.2%	73.7%
Average	79.2%	51.4%	68.3%	79.5%	84.8%	88.7%	86.5%	88.7%	78.4%

`

7.2.2.3 Recognition accuracy using strategy *c*

We have achieved an accuracy of 95.6% when we used strategy *c* and inferred that *k*-NN is the best classifier combination for offline handwritten *Gurmukhi* character recognition for this strategy. Recognition results for this partitioning strategy are given in Table 7.9.

Table 7.9: Classifier wise recognition accuracy for *category* 2 samples with strategy *c*

Principal Components	Linear -SVM	Poly.- SVM	RBF- SVM	*k*-NN	LPR	PRK	LRK	LPK	Average
2-PC	82.7%	83.1%	84.9%	94.6%	88.1%	90.9%	90.2%	90.4%	88.1%
3-PC	82.2%	70.4%	81.3%	**95.6%**	85.8%	86.4%	88.3%	86.7%	84.6%
4-PC	82.4%	42.4%	79.8%	86.2%	85.6%	87.3%	87.5%	88.3%	79.9%
5-PC	82.5%	50.0%	81.4%	83.0%	86.6%	88.8%	88.1%	89.9%	81.3%
6-PC	84.5%	59.7%	83.9%	79.5%	87.8%	88.5%	90.1%	90.0%	83.0%
7-PC	86.6%	67.1%	85.8%	76.6%	90.3%	90.4%	92.8%	90.6%	85.0%
8-PC	81.4%	83.0%	86.6%	88.8%	88.1%	89.3%	91.1%	89.2%	87.2%
9-PC	83.9%	80.2%	82.4%	88.5%	90.1%	82.1%	89.3%	88.2%	85.6%
10-PC	84.1%	81.1%	82.2%	82.3%	87.2%	89.3%	88.3%	89.1%	85.4%
11-PC	84.5%	82.2%	83.1%	81.8%	87.1%	89.1%	88.2%	89.2%	85.7%
12-PC	82.2%	81.2%	82.1%	83.0%	82.2%	89.3%	90.1%	89.2%	84.9%
12-Features	82.2%	84.9%	22.6%	66.9%	88.2%	85.2%	82.5%	89.2%	75.2%
Average	83.3%	72.1%	78.0%	83.9%	87.3%	88.0%	88.9%	89.2%	83.8%

7.2.2.4 Recognition accuracy using strategy *d*

In this sub-section, classifier wise recognition results of partitioning strategy *d* have been presented. When we consider this strategy, then LRK is the best classifiers combination for offline handwritten *Gurmukhi* character recognition. The maximum accuracy that could be achieved is 99.3% for this strategy. Recognition results for this strategy are depicted in Table 7.10.

Table 7.10: Classifier wise recognition accuracy for *category* 2 samples with strategy *d*

Principal Components	Linear -SVM	Poly.- SVM	RBF- SVM	*k*-NN	LPR	PRK	LRK	LPK	Average
2-PC	90.7%	91.9%	91.6%	93.6%	98.1%	98.4%	98.4%	98.4%	95.1%
3-PC	90.0%	83.0%	88.7%	94.4%	95.1%	97.7%	97.7%	97.7%	93.1%
4-PC	90.6%	55.6%	88.0%	87.4%	96.1%	94.9%	97.1%	94.9%	88.1%
5-PC	91.0%	65.2%	88.9%	82.6%	95.4%	97.0%	97.1%	97.0%	89.3%
6-PC	92.0%	75.9%	90.4%	83.7%	96.6%	98.4%	98.4%	98.4%	91.7%
7-PC	93.0%	80.6%	91.7%	77.9%	97.7%	97.6%	**99.3%**	97.6%	91.9%
8-PC	88.0%	87.4%	96.1%	94.9%	97.1%	90.0%	97.0%	92.2%	92.8%
9-PC	83.0%	88.7%	94.4%	95.1%	91.0%	65.2%	97.0%	94.6%	88.6%
10-PC	90.4%	83.7%	96.6%	92.1%	98.4%	90.0%	83.0%	95.3%	91.2%
11-PC	90.3%	84.3%	88.9%	82.6%	95.4%	90.1%	97.1%	98.3%	90.9%
12-PC	89.1%	84.3%	89.1%	92.1%	83.0%	91.1%	92.2%	97.1%	89.8%
12-Features	88.6%	92.7%	31.7%	67.6%	96.7%	92.4%	91.1%	97.6%	82.3%
Average	89.7%	81.1%	86.3%	87.0%	95.1%	91.9%	95.5%	96.6%	90.4%

7.2.2.5 Recognition accuracy using strategy *e*

In partitioning strategy *e*, the maximum accuracy that could be achieved is 99.7%. Using this partitioning strategy, we have noticed that PRK is the best classifiers combination for offline handwritten *Gurmukhi* character recognition. Recognition results for the features and twelve principal components under consideration, using this strategy, are illustrated in Table 7.11.

7.2.2.6 Recognition accuracy using 5-fold cross validation technique

In this sub-section, 5-fold cross validation technique has been considered for training and testing data set partitioning. Using 5-fold cross validation approach, maximum recognition accuracy of 93.0% with *k*-NN classifier has been achieved. Recognition results of different classifiers and their combinations for twelve features (12-feature) and twelve principal components are given in Table 7.12.

`

Table 7.11: Classifier wise recognition accuracy for *category* 2 samples with strategy *e*

Principal Components	Linear -SVM	Poly.- SVM	RBF- SVM	*k*-NN	LPR	PRK	LRK	LPK	Average
2-PC	89.5%	89.2%	88.6%	92.9%	99.4%	99.4%	99.0%	99.0%	94.6%
3-PC	89.5%	87.2%	87.5%	95.7%	98.0%	98.0%	97.7%	98.1%	94.0%
4-PC	89.5%	67.8%	87.4%	93.7%	98.6%	98.9%	97.0%	96.2%	91.1%
5-PC	89.5%	80.6%	88.0%	85.1%	98.9%	97.7%	97.7%	98.6%	92.0%
6-PC	89.5%	84.9%	88.3%	82.2%	99.1%	98.3%	98.0%	99.7%	92.5%
7-PC	89.7%	84.3%	88.6%	92.2%	99.1%	99.1%	98.1%	99.0%	93.8%
8-PC	87.5%	93.7%	98.6%	95.7%	97.0%	96.3%	98.4%	92.3%	94.9%
9-PC	67.8%	87.5%	93.7%	98.6%	98.9%	97.7%	97.7%	98.6%	92.5%
10-PC	88.3%	78.0%	99.1%	98.3%	98.0%	**99.7%**	96.4%	93.6%	93.9%
11-PC	84.3%	88.0%	85.1%	92.3%	97.1%	97.1%	97.7%	97.1%	92.4%
12-PC	86.2%	87.1%	86.1%	92.2%	94.3%	98.8%	94.3%	95.2%	91.8%
12-Features	88.9%	80.6%	75.5%	48.0%	99.1%	99.4%	99.4%	99.4%	86.3%
Average	86.7%	84.1%	88.9%	88.9%	98.1%	98.4%	97.6%	97.2%	92.5%

Table 7.12: Classifier wise recognition accuracy for *category* 2 samples with 5-fold cross validation technique

Principal Components	Linear -SVM	Poly.- SVM	RBF- SVM	*k*-NN	LPR	PRK	LRK	LPK	Average
2-PC	77.8%	77.6%	78.9%	91.3%	89.2%	91.2%	90.6%	91.6%	86.0%
3-PC	81.8%	69.8%	80.8%	**93.0%**	87.2%	89.6%	89.0%	90.3%	85.2%
4-PC	81.5%	44.2%	79.5%	85.7%	87.0%	89.9%	88.6%	90.1%	80.8%
5-PC	81.9%	53.0%	81.1%	79.3%	87.6%	90.4%	89.0%	90.6%	81.6%
6-PC	83.9%	61.5%	82.8%	76.1%	89.2%	91.5%	90.5%	92.3%	83.5%
7-PC	85.2%	65.5%	84.2%	76.7%	90.1%	91.5%	90.9%	92.2%	84.5%
8-PC	81.7%	68.4%	86.0%	86.5%	87.9%	88.1%	90.1%	88.9%	84.7%
9-PC	76.9%	69.9%	75.0%	85.6%	87.6%	81.7%	88.3%	90.7%	81.9%
10-PC	82.0%	67.6%	81.4%	85.8%	88.6%	88.8%	85.4%	88.0%	83.5%
11-PC	81.7%	71.9%	80.6%	83.1%	87.7%	85.6%	88.5%	89.4%	83.6%
12-PC	81.4%	70.1%	77.1%	85.3%	84.5%	87.9%	87.5%	89.8%	83.0%
12-Features	81.6%	77.3%	31.9%	59.4%	89.2%	88.4%	87.3%	90.1%	75.7%
Average	81.4%	66.4%	76.6%	82.3%	88.0%	88.7%	88.8%	90.3%	82.8%

7.2.3 Recognition accuracy for *category* 3 samples

In this section, we have considered each *Gurmukhi* character written by one hundred different writers. Here, the principal components, two principal components (2-PC), three principal components (3-PC), …, twelve principal components (12-PC) have been considered once again to be taken as input to the classifiers. The results for this case are presented in the following sub-sections.

7.2.3.1 Recognition accuracy using strategy *a*

In this sub-section, we have presented classifier wise recognition results of partitioning strategy *a*. In this strategy, maximum recognition accuracy that could be achieved is 89.2%. Using this strategy, we have observed that LPK is the best classifiers combination for offline handwritten *Gurmukhi* character recognition. Recognition results of different classifiers and their combinations are given in Table 7.13 for twelve features (12-feature) and twelve principal components.

Table 7.13: Classifier wise recognition accuracy for *category* 3 samples with strategy *a*

Principal Components	Linear -SVM	Poly.- SVM	RBF- SVM	*k*-NN	LPR	PRK	LRK	LPK	Average
2-PC	74.9%	75.0%	78.4%	80.5%	81.8%	86.5%	86.1%	**89.2%**	81.5%
3-PC	72.8%	30.2%	69.3%	75.7%	80.1%	85.9%	83.4%	86.5%	73.0%
4-PC	71.6%	14.4%	66.6%	64.1%	77.9%	88.1%	81.9%	85.2%	68.7%
5-PC	72.9%	17.7%	67.1%	58.5%	78.4%	79.6%	82.9%	87.4%	68.0%
6-PC	77.3%	23.9%	72.9%	48.8%	82.5%	83.9%	85.8%	87.7%	70.3%
7-PC	77.8%	34.7%	72.9%	57.5%	82.3%	84.9%	84.9%	87.9%	72.9%
8-PC	73.9%	16.3%	68.0%	65.3%	79.4%	80.6%	84.3%	86.3%	69.3%
9-PC	74.1%	21.8%	68.1%	59.3%	80.8%	79.3%	84.2%	89.1%	69.6%
10-PC	77.7%	30.7%	72.3%	47.1%	82.2%	83.3%	84.8%	89.2%	70.9%
11-PC	75.1%	32.1%	68.1%	79.1%	80.3%	82.2%	83.1%	88.1%	73.5%
12-PC	76.2%	45.1%	69.2%	78.2%	81.2%	81.2%	82.3%	88.1%	75.2%
12-Features	75.8%	69.7%	17.8%	43.9%	81.1%	85.6%	82.7%	87.2%	68.0%
Average	75.0%	34.3%	65.9%	63.2%	80.7%	83.4%	83.9%	87.7%	71.7%

`

7.2.3.2 Recognition accuracy using strategy *b*

In partitioning strategy *b*, the maximum accuracy that could be achieved is 89.7%. Using this strategy, we have seen that again LPK is the best classifiers combination for offline handwritten *Gurmukhi* character recognition. Recognition results for this strategy are illustrated in Table 7.14.

Table 7.14: Classifier wise recognition accuracy for *category* 3 samples with strategy *b*

Principal Components	Linear -SVM	Poly.- SVM	RBF- SVM	*k*-NN	LPR	PRK	LRK	LPK	Average
2-PC	75.8%	76.2%	78.4%	79.7%	82.9%	86.8%	87.4%	**89.7%**	82.1%
3-PC	73.7%	35.7%	70.5%	77.6%	81.9%	87.2%	84.9%	88.7%	75.0%
4-PC	73.9%	16.3%	67.2%	67.0%	79.4%	80.6%	84.3%	86.5%	69.4%
5-PC	74.1%	21.8%	68.5%	59.3%	80.8%	79.8%	84.2%	83.3%	69.0%
6-PC	77.7%	30.7%	73.2%	47.1%	82.2%	83.8%	84.8%	85.4%	70.6%
7-PC	78.1%	40.5%	74.2%	57.6%	83.4%	83.9%	86.1%	87.4%	73.9%
8-PC	78.0%	41.3%	70.1%	76.7%	82.3%	82.3%	82.3%	84.3%	74.7%
9-PC	72.3%	52.3%	71.1%	76.2%	83.1%	82.1%	83.1%	85.1%	75.7%
10-PC	73.2%	43.3%	72.3%	76.3%	80.3%	83.1%	85.1%	87.1%	75.1%
11-PC	74.2%	44.3%	73.3%	78.3%	81.3%	84.3%	84.3%	89.1%	76.1%
12-PC	74.3%	45.1%	74.3%	72.2%	82.2%	84.1%	80.5%	88.2%	75.1%
12-Features	75.4%	51.3%	20.3%	40.7%	82.1%	87.7%	79.5%	81.0%	64.8%
Average	75.1%	41.6%	67.8%	67.4%	81.8%	83.8%	83.9%	86.3%	73.5%

7.2.3.3 Recognition accuracy using strategy *c*

In this sub-section, classifier wise recognition results of partitioning strategy *c* have been presented. Here, LPK again emerged as the best classifiers combination when we followed this strategy. Maximum recognition accuracy of 89.2% could be achieved in this strategy. Recognition results of different classifiers and their combinations for twelve features (12-feature) and twelve principal components are given in Table 7.15.

Table 7.15: Classifier wise recognition accuracy for *category* 3 samples with strategy *c*

Principal Components	Linear -SVM	Poly.- SVM	RBF- SVM	*k*-NN	LPR	PRK	LRK	LPK	Average
2-PC	77.7%	77.4%	80.2%	81.1%	83.9%	86.6%	87.8%	88.7%	82.9%
3-PC	74.5%	43.7%	72.8%	80.3%	81.3%	84.1%	85.6%	86.7%	76.1%
4-PC	74.6%	21.1%	70.0%	68.6%	79.5%	79.2%	83.6%	84.3%	70.1%
5-PC	75.5%	27.4%	70.9%	60.7%	80.9%	79.3%	83.9%	81.8%	70.0%
6-PC	79.9%	38.2%	75.3%	50.8%	83.3%	83.4%	86.0%	86.4%	72.9%
7-PC	80.0%	45.1%	75.7%	59.7%	84.9%	85.1%	86.7%	88.3%	75.7%
8-PC	78.1%	34.6%	73.1%	60.1%	82.3%	82.3%	85.1%	87.3%	72.9%
9-PC	79.3%	54.4%	74.2%	62.1%	83.1%	84.1%	84.2%	87.2%	76.1%
10-PC	80.0%	34.4%	74.3%	62.3%	82.2%	82.2%	84.4%	87.1%	73.4%
11-PC	78.2%	29.3%	75.1%	63.4%	83.1%	84.1%	84.2%	87.5%	73.1%
12-PC	72.5%	40.2%	75.2%	64.1%	80.2%	84.1%	84.5%	**89.2%**	73.8%
12-Features	75.7%	59.7%	25.5%	42.7%	80.1%	69.0%	69.0%	87.3%	63.7%
Average	77.2%	42.1%	70.2%	63.0%	82.1%	81.9%	83.7%	86.8%	73.4%

7.2.3.4 Recognition accuracy using strategy *d*

Table 7.16: Classifier wise recognition accuracy for *category* 3 samples with strategy *d*

Principal Components	Linear -SVM	Poly.- SVM	RBF- SVM	*k*-NN	LPR	PRK	LRK	LPK	Average
2-PC	78.0%	78.6%	80.7%	80.3%	86.6%	89.3%	90.1%	**92.3%**	84.5%
3-PC	75.9%	51.9%	70.9%	77.7%	81.9%	86.6%	86.3%	89.4%	77.6%
4-PC	75.0%	26.2%	69.3%	62.6%	81.7%	79.6%	86.4%	84.9%	70.7%
5-PC	75.0%	29.8%	71.0%	54.9%	81.3%	79.4%	86.3%	81.6%	69.9%
6-PC	79.0%	46.2%	75.7%	42.4%	86.1%	87.6%	88.1%	89.6%	74.4%
7-PC	80.7%	51.2%	76.3%	51.2%	87.9%	88.3%	90.0%	90.3%	77.0%
8-PC	78.2%	51.2%	75.3%	52.3%	82.3%	87.3%	85.2%	88.1%	75.0%
9-PC	79.1%	49.3%	75.3%	58.3%	85.4%	87.1%	87.2%	89.1%	76.4%
10-PC	78.0%	49.2%	75.3%	57.3%	87.3%	80.1%	87.8%	90.3%	75.7%
11-PC	72.3%	40.3%	75.2%	59.3%	87.2%	79.9%	86.3%	91.1%	73.9%
12-PC	73.5%	40.4%	74.3%	58.3%	87.3%	79.3%	87.1%	89.5%	73.7%
12-Features	75.0%	81.5%	32.5%	35.3%	82.1%	73.3%	70.0%	90.3%	67.5%
Average	76.7%	49.7%	70.9%	57.5%	84.8%	83.1%	85.9%	88.9%	74.7%

`

In partitioning strategy *d*, the maximum accuracy that could be achieved is 92.3%. Using this strategy, we have seen that LPK is the best classifiers combination for offline handwritten *Gurmukhi* character recognition. Recognition results for this strategy are given in Table 7.16.

7.2.3.5 Recognition accuracy using strategy *e*

In this sub-section, classifier wise recognition results of partitioning strategy *e* have been presented. Here, LRK is the best classifiers combination for offline handwritten *Gurmukhi* character recognition. We have achieved maximum recognition accuracy of 87.9% in this strategy. Recognition results for this strategy are shown in Table 7.17.

Table 7.17: Classifier wise recognition accuracy for *category* 3 samples with strategy *e*

Principal Components	Linear -SVM	Poly.- SVM	RBF- SVM	*k*-NN	LPR	PRK	LRK	LPK	Average
2-PC	74.6%	76.1%	79.5%	77.1%	84.3%	85.9%	**87.9%**	87.1%	81.6%
3-PC	70.7%	51.9%	65.2%	72.9%	76.0%	76.9%	82.0%	80.6%	72.0%
4-PC	70.9%	28.5%	62.4%	57.1%	75.1%	65.4%	83.1%	66.6%	63.7%
5-PC	70.1%	33.3%	64.9%	51.4%	77.7%	70.6%	83.7%	73.1%	65.6%
6-PC	73.2%	47.9%	69.2%	35.7%	83.1%	76.9%	85.4%	80.6%	69.0%
7-PC	76.4%	54.7%	72.4%	35.4%	85.4%	78.3%	85.4%	77.9%	70.7%
8-PC	72.2%	65.6%	78.1%	56.2%	78.1%	77.3%	76.5%	78.1%	72.8%
9-PC	72.3%	76.1%	65.3%	54.3%	77.3%	78.2%	76.5%	79.1%	72.4%
10-PC	71.5%	70.3%	69.2%	59.1%	78.2%	78.2%	76.5%	78.1%	72.6%
11-PC	72.3%	69.2%	69.4%	57.2%	78.0%	74.6%	77.5%	76.3%	71.8%
12-PC	73.4%	58.1%	68.5%	54.4%	78.9%	76.3%	76.3%	75.3%	70.1%
12-Features	69.2%	67.0%	65.2%	27.7%	82.9%	87.1%	82.3%	83.4%	70.6%
Average	72.2%	58.2%	69.1%	53.2%	79.6%	77.1%	81.1%	78.0%	71.1%

7.2.3.6 Recognition accuracy using 5-fold cross validation technique

In this sub-section, classifier wise recognition results of 5-fold cross validation technique have been presented. For the features and principal components under consideration and 5-fold cross validation technique, maximum recognition accuracy of 87.6% using LPK

classifiers combination has been achieved. Recognition results of different classifiers and their combinations for twelve features (12-feature) and twelve principal components are given in Table 7.18.

Table 7.18: Classifier wise recognition accuracy for *category* 3 samples with 5-fold cross validation technique

Principal Components	Linear -SVM	Poly.- SVM	RBF- SVM	*k*-NN	LPR	PRK	LRK	LPK	Average
2-PC	74.7%	75.1%	77.9%	78.1%	82.2%	85.3%	86.1%	**87.6%**	80.9%
3-PC	72.0%	41.8%	68.3%	75.3%	78.6%	82.5%	82.8%	84.7%	73.2%
4-PC	71.7%	20.9%	65.8%	62.6%	77.1%	77.0%	82.2%	79.9%	67.1%
5-PC	72.0%	25.5%	67.1%	55.8%	78.2%	76.2%	82.5%	79.8%	67.1%
6-PC	75.9%	36.6%	71.8%	44.1%	81.8%	81.5%	84.3%	84.2%	70.0%
7-PC	77.0%	44.3%	72.8%	51.2%	83.1%	82.4%	84.9%	84.6%	72.6%
8-PC	74.6%	41.0%	71.5%	60.9%	79.3%	80.3%	81.0%	83.1%	71.5%
9-PC	73.9%	49.8%	69.4%	60.8%	80.3%	80.5%	81.4%	84.2%	72.6%
10-PC	74.6%	44.7%	71.2%	59.2%	80.4%	79.8%	82.0%	84.6%	72.1%
11-PC	72.9%	42.2%	70.8%	66.1%	80.3%	79.4%	81.4%	84.7%	72.2%
12-PC	72.5%	44.9%	70.9%	64.1%	80.3%	79.4%	80.5%	84.3%	72.1%
12-Features	72.7%	64.5%	31.6%	37.3%	80.0%	78.9%	75.2%	84.1%	65.6%
Average	73.7%	44.3%	67.4%	59.6%	80.2%	80.2%	82.0%	83.8%	71.4%

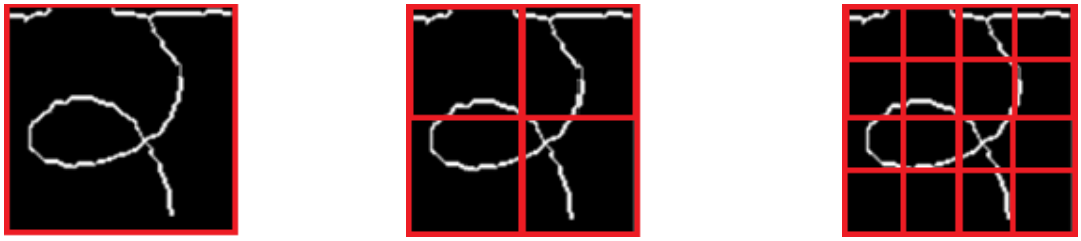
7.3 Hierarchical feature extraction technique for offline handwritten *Gurmukhi* character recognition

We have also proposed a hierarchical feature extraction technique for offline handwritten *Gurmukhi* character recognition. The main aim of the feature extraction phase is to detect preeminent features of digitized character image, which maximize the recognition accuracy in the least amount of time. But, training of the classifier with large number of features acquired is not at all times the best decision, as the unrelated or surplus features can cause harmful impact on a classifier's performance and at the same time, the classifier can become computationally complex.

For optical character recognition, we have to input all the handwritten character images in a uniform size, *i.e.*, the character images should be in standard shape. In order to keep our

`

algorithm easy, here in this work all the character images are reformed in the size of 88×88 pixels, using Nearest Neighborhood Interpolation (NNI) algorithm. After that, we have proposed a feature set of 105 feature elements using four types of topological features, viz., horizontally peak extent features, vertically peak extent features, diagonal features and centroid features. For extracting these features, initially, we have divided the digitized image into number of zones as shown in Figure 7.1. Let L be the current level of image. At this level, the number of the sub-images is $4^{(L)}$. For example, when L = 1 the number of sub-images is 4 and when L = 2 it is 16. So, for every L a $4^{(L)}$- dimensional feature vector is extracted. Here, we have considered L = (0, 1, and 2) in this work.



(a) **(b)** **(c)**

Figure 7.1 Digitized image of *Gurmukhi* character (ਕ) (a) at level L = 0 (b) at level L = 1 (c) at level L = 2

The steps that have been used to extract horizontally peak extent features are given below:

Step 1: Input the initial value of L is 0.

Step II: Divide a bitmap image into $4^{(L)}$ number of zones, each of equal sized (Figure 7.1).

Step III: Find the peak extent as sum of successive foreground pixels in each row of a sub image at each level L.

Step IV: Replace the values of successive foreground pixels by peak extent value, in each row of a zone.

Step V: Find the largest value of peak extent in each row.

Step VI: Obtain the sum of these largest peak extent sub-feature values for each sub-image and consider this as a feature for the corresponding zone.

Step VII: For the zones that do not have a foreground pixel, take the feature value as zero.

Step VIII: IF L<2 then

(a) Set $L=L+1$

(b) Go to step II

Else

Return

Step IX: Normalize the values in the feature vector in scale of 0 to 1 by using:

$$\text{Normalized feature } NV_i = \frac{(\text{Actual feature } V_i - \min \text{ of actual feature vector})}{(\max \text{ of actual feature vector} - \min \text{ of actual feature vector})}$$

Step X: Return

These steps will give a feature set with $4^{(L)}$ elements at each level L.

Similarly, for vertical peak extent features and diagonal features, we have extracted $4^{(L)}$ feature elements. For centroid features, we have extracted $2 \times 4^{(L)}$ feature elements.

As such, we have extracted a feature set of 105 feature elements that include 21 feature elements with horizontally peak extent features, 21 feature elements with vertically peak extent features, 21 feature elements with diagonal features and 42 elements with centroid features. The size of this feature set as 105 elements, being larger in number, can lead to computational complexity of the classifier learning models. Therefore, we reduce the length of feature set using various feature selection techniques. Feature selection is the technique which is used for selecting a subset of relevant features to improve the recognition accuracy and performance of classifier learning model by speeding up the learning process and reducing computational complexities. Three feature selection methods that have been considered in this work are PCA, Consistency Based (CON) and Correlation Feature Set (CFS) as these methods have been widely used in pattern recognition and character recognition for reducing the length of feature set.

7.3.1 Experimental results based on hierarchical feature extraction technique

In this section, experimental results based on hierarchical feature extraction technique for offline handwritten *Gurmukhi* character recognition are illustrated. In the process of evaluating

`

the performance of this technique with SVM classifier, we have considered 5600 samples of isolated offline handwritten *Gurmukhi* characters written by one hundred different writers (*Category* 3). The complete feature set, consisting of 105 feature elements, created on the basis of topological features of the character is given to SVMs for their training. Due to the high dimensionality of feature vector, training a classifier here becomes computationally complex. So, its dimensionality is reduced to 51, 55 and 12 using PCA (Sundaram and Ramakrishnan, 2008), CON (Dash and Liu, 2003) and CFS (Hall, 1998) feature selection techniques, respectively. We have used 5-fold cross validation technique for obtaining recognition accuracy. Using this technique, we have achieved a recognition accuracy of 91.8% with hierarchical feature extraction technique for offline handwritten *Gurmukhi* character and SVM with linear kernel classifier. For this case, confusion matrix for the *Gurmukhi* characters is given in Table 7.19. The recognition results of different features selection techniques and then complete feature set considered under this work are depicted in Table 7.20. These results are graphically shown in Figure 7.2.

Table 7.19: Confusion matrix based upon PCA feature set and SVM with linear kernel classifier

Character	Recognition Accuracy	Confused with characters						
ੳ	ੳ 99%	ਬ 1%						
ਅ	ਅ 100%							
ੲ	ੲ 92%	ਵ 1%	ਟ 2%	ਣ 1%	ਦ 1%	ਬ 1%	ਲ 2%	
ਸ	ਸ 85%	ਗ 2%	ਧ 1%	ਮ 10%	ਲ 2%			
ਹ	ਹ 98%	ਤ 1%	ਰ 1%					
ਕ	ਕ 91%	ਖ 1%	ਘ 1%	ਚ 1%	ਥ 1%	ਬ 1%	ਰ 2%	ੜ 2%
ਖ	ਖ 87%	ਖ 2%	ਛ 1%	ਥ 5%	ਧ 2%	ਪ 1%	ਬ 1%	ਲ 1%
ਗ	ਗ 92%	ਸ 3%	ਮ 3%	ਰ 2%				
ਘ	ਘ 89%	ਅ 2%	ਕ 1%	ਗ 1%	ਛ 1%	ਜ 1%	ਥ 1%	ਪ 4%
ਙ	ਙ 93%	ੳ 1%	ਕ 1%	ਛ 1%	ਝ 1%	ਡ 1%	ਣ 1%	ਥ 1%

ਚ	ਚ 91%	ਢ 8%	ਦ 1%					
ਛ	ਛ 92%	ਗ 1%	ਝ 2%	ਢ 2%	ਫ 1%	ਬ 1%	ਭ 1%	
ਜ	ਜ 97%	ਫ 1%	ਧ 1%	ਵ 1%				
ਝ	ਝ 96%	ੳ 2%	ਝ 1%	ਢ 1%				
ਞ	ਞ 86%	ਏ 1%	ਣ 1%	ਵ 12%				
ਟ	ਟ 95%	ਏ 2%	ਦ 1%	ਬ 1%	ਮ 1%			
ਠ	ਠ 92%	ਢ 1%	ਨ 1%	ਪ 3%	ਬ 1%	ਭ 1%	ਲ 1%	
ਡ	ਡ 96%	ਖ 1%	ਠ 1%	ਰ 2%				
ਢ	ਢ 91%	ਚ 9%						
ਣ	ਣ 94%	ਵ 2%	ਟ 1%	ਬ 1%	ਮ 1%	ਯ 1%		
ਤ	ਤ 97%	ਛ 1%	ਭ 1%	ਯ 1%				
ਥ	ਥ 82%	ਏ 1%	ਖ 8%	ਝ 1%	ਬ 7%	ਯ 1%		
ਦ	ਦ 93%	ਏ 1%	ਚ 2%	ਜ 1%	ਲ 1%	ਵ 2%		
ਧ	ਧ 85%	ਖ 1%	ਘ 1%	ਵ 1%	ਪ 12%			
ਨ	ਨ 95%	ਗ 1%	ਠ 1%	ਲ 3%				
ਪ	ਪ 81%	ਧ 16%	ਛ 2%	ਝ 1%				
ਫ	ਫ 91%	ਝ 1%	ਠ 6%	ਧ 1%	ਨ 1%			
ਬ	ਬ 93%	ੳ 1%	ਖ 1%	ਥ 2%	ਪ 1%	ਮ 1%	ਰ 1%	
ਭ	ਭ 94%	ਝ 1%	ਠ 3%	ਤ 1%	ਥ 1%			
ਮ	ਮ 84%	ਸ 11%	ਗ 2%	ਣ 1%	ਵ 2%			
ਯ	ਯ 95%	ਵ 2%	ਧ 1%	ਸ 1%	ਗ 1%			
ਰ	ਰ 91%	ਕ 3%	ਗ 1%	ਡ 2%	ਭ 1%	ਮ 1%	ੜ 1%	
ਲ	ਲ	ਸ	ਝ	ਨ	ਬ	ੜ		

`

	93%	1%	1%	3%	1%	1%		
ਵ	ਵ 86%	ਏ 1%	ਜ 2%	ਢ 10%	ਦ 1%			
ਸ	ਸ 97%	ਜ 1%	ਰ 2%					

Table 7.20: Recognition results of different feature selection techniques and complete feature set

Classifier	Full Feature Set	PCA	CFS	CON
Linear-SVM	90.1%	**91.8%**	88.9%	66.7%
Polynomial-SVM	62.8%	90.1%	62.9%	22.5%
RBF-SVM	70.2%	90.3%	74.8%	56.1%
Sigmoid-SVM	63.5%	82.6%	68.3%	48.5%

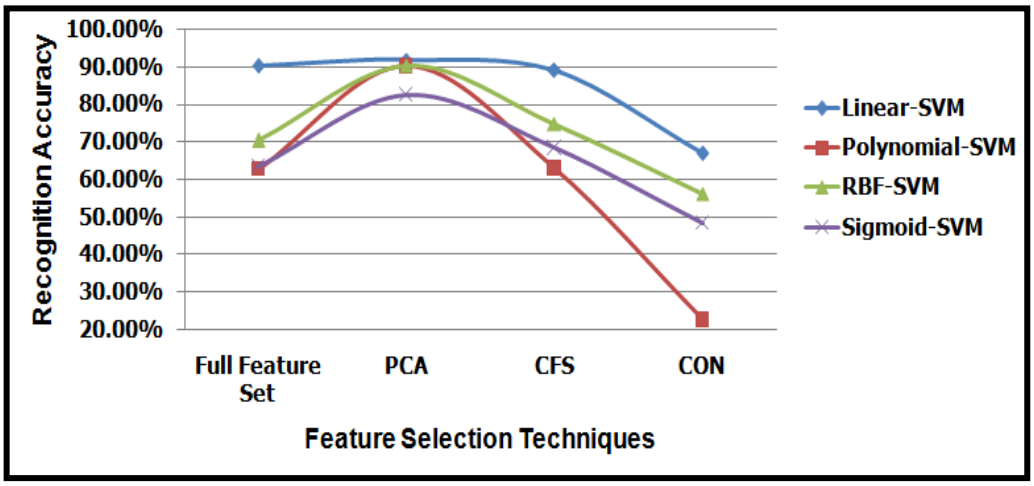


Figure 7.2: Recognition accuracy achieved with various feature selection techniques and using various kernels of SVM

7.4 Chapter summary

In this chapter, we have presented PCA based offline handwritten Gurmukhi character recognition system. A hierarchical feature extraction technique for offline handwritten Gurmukhi character recognition has also been discussed. The features of a character that have been considered in PCA based recognition system include zoning features, diagonal features,

directional features, transition features, intersection and open end points features, parabola curve fitting based features, power curve fitting based features, shadow features, centroid features, peak extent based features and modified division points based features. The classifiers that have been employed in this work are *k*-NN, Linear-SVM, Polynomial-SVM and RBF-SVM and the combinations of these classifiers. The proposed system achieves maximum recognition accuracy of 99.9% for *category* 1 samples, of 99.7% for *category* 2 samples and of 92.3% for *category* 3 samples as shown in Table 7.21. Using 5-fold cross validation approach, maximum recognition accuracy of 97.3% for *category* 1 samples, of 93.0% for *category* 2 samples and of 87.6% for *category* 3 samples have been achieved.

Table 7.21: Category wise recognition accuracy

Samples	Feature	Classifier	Accuracy (%)
category 1,strategy *a*	4-PC	LRK	98.9%
category 1, strategy *b*	9-PC	LPR	99.6%
category 1, strategy *c*	8-PC	LPR	99.5%
category 1, strategy *d*	2-PC	LRK	99.6%
category 1, strategy *e*	9-PC	LRK	**99.9%**
category 1, 5-fold cross validation	7-PC	LRK	97.3%
category 2, strategy *a*	3-PC	*k*-NN	94.5%
category 2, strategy *b*	3-PC	*k*-NN	94.5%
category 2, strategy *c*	3-PC	*k*-NN	95.6%
category 2, strategy *d*	7-PC	LRK	99.3%
category 2, strategy *e*	10-PC	PRK	99.7%
category 2, 5-fold cross validation	3-PC	*k*-NN	93.0%
category 3, strategy *a*	2-PC	LPK	89.2%
category 3, strategy *b*	2-PC	LPK	89.7%
category 3, strategy *c*	12-PC	LPK	89.2%
category 3, strategy *d*	2-PC	LPK	92.3%
category 3, strategy *e*	2-PC	LRK	87.9%
category 3, 5-fold cross validation	2-PC	LPK	87.6%

In hierarchical feature extraction technique, a powerful feature set of 105 feature elements is proposed in this work for recognition of offline handwritten *Gurmukhi* characters using four

`

types of topological features, namely, horizontally peak extent features, vertically peak extent features, diagonal features, and centroid features. It is a well known fact that training of classifiers with large number of features acquired is not at all times the best decision. So, PCA, CFS and CON feature selection techniques are applied to reduce the dimensionality of the feature vector. Recognition accuracy of 91.8% was achieved with the proposed technique while using PCA feature set and SVM with linear kernel classifier. Also, it has been seen that PCA performs better than CFS and CON feature selection techniques for character recognition.

Chapter 8
Conclusions and Future Scope

==========

Offline handwritten character recognition has now been in research for more than four decades. The work done by researchers in this area is admirable. Most of the work in this research area has been done for non-Indian scripts but recent literature shows that researchers have also achieved encouraging results for Indian scripts such as *Bangla, Devanagari, Gurmukhi, Kannada, Oriya, Tamil* and *Telugu*. The main objective of this thesis was to build an offline handwritten *Gurmukhi* script recognition system. This objective has been achieved well as the system so developed recognizes offline handwritten *Gurmukhi* script. This thesis proposes algorithms in various phases of an offline handwritten *Gurmukhi* script recognition system. Related literature on this work has been included in Chapter 2. Chapter 3 includes data collection, digitization, pre-processing and segmentation tasks for this work. A novel technique has been proposed in this work for line segmentation of offline handwritten *Gurmukhi* script documents in Chapter 3. In Chapter 4, we have presented a handwriting grading system based on offline *Gurmukhi* characters. In Chapter 5, we have presented curve fitting based novel feature extraction techniques, namely, parabola curve fitting based features and power curve fitting based features for offline handwritten *Gurmukhi* character recognition. The classifiers that have been used in this work are *k*-NN and SVM, with three flavours, *i.e*., Linear-SVM, Polynomial-SVM and RBF-SVM. Chapter 6 includes an offline handwritten *Gurmukhi* character recognition system using zoning based novel feature extraction methods and *k*-fold cross validation. In this work, we have also used various existing feature extraction techniques, namely, zoning features, diagonal features, directional features, intersection and open end points features, transition features, shadow features, centroid features, peak extent based features and modified division point based features for

offline handwritten *Gurmukhi* character recognition. The peak extent based features and modified division point based features are the new features proposed in this chapter. For classification, different classifiers namely, *k*-NN, Linear-SVM, Polynomial-SVM (with degree 3) and MLPs have been considered in this chapter. Chapter 7 includes an analysis for PCA based offline handwritten *Gurmukhi* character recognition system. A hierarchical feature extraction technique for offline handwritten Gurmukhi character recognition has also been presented in this chapter. We have used *k*-NN, Linear-SVM, Polynomial-SVM and RBF-SVM based approaches and also combinations of these approaches for classification in this chapter. Section 8.1 focuses on the brief contribution of the work and section 8.2 depicts future scope of this work.

8.1 Brief contribution of the work

This work has made the following contributions in the field of offline handwritten *Gurmukhi* script recognition.

8.1.1 Parabola and power curve based novel feature extraction techniques

In this work, we have proposed novel feature extraction techniques, namely, parabola curve fitting based features and power curve fitting based features for offline handwritten *Gurmukhi* character recognition. We have also compared the results of these proposed feature extraction techniques with other recently used feature extraction techniques, namely, zoning features, diagonal features, directional features, intersection and open end points features and transition features. The classifiers that have been employed in this work are *k*-NN and SVM with three flavors, *i.e.*, Linear-SVM, Polynomial-SVM and RBF-SVM. After the comparison of different techniques for character recognition, we have concluded that the results of power curve fitting based features are promising and the system achieves a recognition accuracy of 97.9%, 94.6%, 94.0% and 92.3% using *k*-NN, Linear-SVM, Polynomial-SVM and RBF-SVM classifiers, respectively. These recognition results are presented in Table 5.8. It has also been seen that the results achieved using parabola curve fitting based features are also better than the other recently proposed feature extraction

techniques. A maximum recognition accuracy of 95.4% could be achieved when the parabola fitting based features were used with *k*-NN classifier (*k* = 5).

8.1.2 Handwriting grading system

We have proposed a handwriting grading system based on offline *Gurmukhi* characters. In this work, we have attempted to grade the writers based on offline handwritten *Gurmukhi* characters written by them. We have used zoning; diagonal; directional; intersection and open end points feature extraction techniques in order to find the feature sets and have used HMM and Bayesian decision making classifiers for obtaining a classification score. The proposed handwriting grading system has been tested with the help of five popular printed *Gurmukhi* fonts, namely, *amrit*, *Gurmukhi Lys*, *Granthi*, *LMP_TARAN* and *Maharaja*. In training data set, we have used printed *Gurmukhi* font *Anandpur sahib*. As expected, fonts have a better classification score of gradation in comparison with mortal writers, thereby establishing the effectiveness of the proposed system. The performance of five best mortal writers using this proposed system, are depicted in Tables 4.3. The proposed grading system can be used as a decision support system for grading the handwritings in a competition.

8.1.3 Offline handwritten *Gurmukhi* character recognition system using zoning based novel feature extraction methods and *k*-fold cross validation technique

We have also presented an offline handwritten *Gurmukhi* character recognition system using *k*-fold cross validation technique. In this work, we have used various feature extraction techniques, namely, zoning features, diagonal features, directional features, intersection and open end points features, transition features, shadow features and centroid features. We have also proposed two efficient feature extraction techniques, namely, peak extent based features and modified division point based features in this work. The classifiers that have been employed in this work are *k*-NN, Linear-SVM, Polynomial-SVM and MLPs. We have concluded that peak extent based features have preeminent features when compared to other feature extraction techniques considered in this study. Here, we have used 5600 samples of isolated offline handwritten *Gurmukhi* characters and achieved 5-fold cross validation

`

accuracy with peak extent based features of 95.6%, 92.4%, 95.5% and 94.7% with Linear-SVM, Polynomial-SVM, *k*-NN and MLPs classifier, respectively as depicted in Table 6.5.

8.1.4 PCA based offline handwritten *Gurmukhi* character recognition system

Principal Component Analysis (PCA) has widely been used for extracting representative features for pattern recognition and has also been used to reduce the dimension of data. We have analyzed the performance of features using PCA for offline handwritten *Gurmukhi* character recognition system. The features of a character that have been considered in this work include zoning features, diagonal features, directional features, transition features, intersection and open end points features, parabola curve fitting based features, power curve fitting based features, shadow features, centroid features, peak extent based features and modified division points based features. The classifiers that have been employed here are *k*-NN, Linear-SVM, Polynomial-SVM and RBF-SVM and also different combinations of these classifiers. In this work, we have collected 16,800 samples of offline handwritten *Gurmukhi* characters. These samples have been divided into three categories as discussed in section 7.2. We have divided the data set of each *category* using five partitioning strategies (*a*, *b*, *c*, *d*, and *e*) as given in Table 5.3. The proposed system achieves an average recognition accuracy of 99.9% for *category* 1 samples, of 99.7% for *category* 2 samples and of 92.3% for *category* 3 samples. Using 5-fold cross validation technique, maximum recognition accuracy of 97.3% for *category* 1 samples, of 93.0% for *category* 2 samples and of 87.6% for *category* 3 samples could be achieved. Category wise results of a recognition system based on PCA are depicted in Table 7.21. We have also proposed a hierarchical feature extraction technique for offline handwritten *Gurmukhi* character recognition. In hierarchical feature extraction technique, a powerful feature set of 105 feature elements has been built using four types of topological features, namely, horizontally peak extent features, vertically peak extent features, diagonal features, and centroid features. Using this technique, recognition accuracy of 91.8% was achieved with Linear-SVM classifier. We have also noticed that PCA performs better than CFS and CON feature selection techniques for character recognition.

8.2 Discussion

We have concluded that a hierarchical feature extraction technique performs better than other techniques for offline handwritten *Gurmukhi* character recognition. This hierarchical feature extraction technique, a powerful feature set of 105 feature elements has been built using four types of topological features, namely, horizontally peak extent features, vertically peak extent features, diagonal features, and centroid features. Various feature length reduction techniques, CFS, CON, and PCA are considered in this work and it has also been concluded that PCA performs better than other feature length reduction techniques. We achieved maximum recognition accuracy of 91.8% with Linear-SVM classifier.

8.3 Future scope

The work presented in this thesis can be extended in a number of ways. This section presents some of the directions in which one can extend this work.

These experimentations in this work have been carried out with a design that one writer writes a document n_1 times; m_1 writers write a document n_2 times; and m_2 writers write a document one time. One can think of having a different design of experiments for proposing the offline handwritten character recognition system for *Gurmukhi* script. The number of samples for training and testing can also be experimented for an optimal accuracy of the system.

We could achieve the average line segmentation accuracy of 98.4% in this work. This accuracy can further be increased by proposing more efficient algorithms for line segmentation. The handwriting grading system proposed in this work is based on isolated *Gurmukhi* characters. This work can further be extended to grade the writers based on words, sentences, and paragraphs written by them.

`

In this work, we have proposed four new features to build the offline handwritten character recognition system for *Gurmukhi* script. Proposing of efficient features shall probably, remain an area for future work for quite some more time in this field.

This is well known that *Gurmukhi* shares a number of structural similarities with some other Indian scripts. As such, the work carried out in this thesis can further be extended for these scripts after building a database for these scripts.

References

[1] Abd, M. A. and Paschos, G., 2007. Effective Arabic character recognition using support vector machines. *Innovations and Advanced Techniques in Computer and Information Sciences and Engineering*, pp. 7-11.

[2] Acharya, D., Reddy, N. V. S. and Makkithaya, K., 2008. Multilevel classifiers in recognition of handwritten Kannada numerals. *Proceedings of World Academy of Sciences, Engineering and Technology* (*WASET*), Vol. 42, pp. 278-283.

[3] Ajmire, P. E. and Warkhede, S. E., 2010. Handwritten Marathi character (vowel) recognition. *Advances in Information Mining*, Vol. 2(2), pp. 11-13.

[4] Alaei, A., Pal, U. and Nagabhushan, P., 2009. Using modified contour features and SVM based classifier for the recognition of Persian/Arabic handwritten numerals. *Proceedings of 7th International Conference on Advances in Pattern Recognition* (*ICAPR*), pp. 391-394.

[5] Alaei, A., Nagabhushan, P. and Pal, U., 2010a. A baseline dependent approach for Persian handwritten character segmentation. *Proceedings of 20th International Conference on Pattern Recognition* (*ICPR*), pp. 1977-1980.

[6] Alaei, A., Nagabhushan, P. and Pal, U., 2010b. A new two-stage scheme for the recognition of Persian handwritten characters. *Proceedings of 12th International Conference on Frontiers in Handwriting Recognition* (*ICFHR*), pp. 130-135.

[7] Almuallim, H. and Yamaguchi, S., 1987. A method of recognition of Arabic cursive handwriting. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (*PAMI*), Vol. 9(5), pp. 715-722.

[8] Ananthakrishnan, G., Sen, A., Sundaram, S. and Ramakrishnan, A. G., 2009. Dynamic space warping of sub-strokes for recognition of online handwritten characters. *International Journal of Pattern Recognition and Artificial Intelligence* (*IJPRAI*), Vol. 23(5), pp. 925-943.

[9] Antani, S. and Agnihotri, L., 1999. Gujarati character recognition. *Proceedings of the International Conference on Document Analysis and Recognition* (*ICDAR*), pp. 418-421.

[10] Aparna, K. H., Subramanian, V., Kasirajan, M., Prakash, G. V., Chakravarthy, V. S. and Madhvanath, S., 2004. Online handwriting recognition for Tamil. *Proceedings of 9th International Workshop on Frontiers in Handwriting Recognition* (*IWFHR*), pp. 438-443.

[11] Arivazhagan, M., Srinivasan, H. and Srihari, S., 2007. A statistical approach to line segmentation in handwritten documents. *Proceedings of SPIE*, 6500T

[12] Artieres, T. and Gallinari, P., 2002. Stroke level HMMs for on-line handwriting recognition. *Proceedings of 8th International Workshop on Frontiers in Handwriting Recognition* (*IWFHR*), pp. 227-232.

[13] Artieres, T., Marukatat, S. and Gallinari, P., 2007. Online handwritten shape recognition using segmental Hidden Markov Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (*PAMI*), Vol. 29(2), pp. 205-217.

[14] Arora, A. and Namboodiri, A. M., 2010. A hybrid model for recognition of online handwriting in Indian scripts. *Proceedings of 12th International Conference on Frontiers in Handwriting Recognition* (*ICFHR*), pp. 433-438.

[15] Arora, S., Bhattacharjee, D., Nasipuri, M., Basu, K., D. and Kundu, M., 2008. Combining multiple feature extraction techniques for handwritten Devnagari character recognition. *Proceeding of 10th Colloquium and 3rd International Conference on Industrial and Information Systems* (*ICIIS*), pp. 1-6.

[16] Ashwin, T. V. and Sastry, P. S., 2002. A font and size-independent OCR system for printed Kannada documents using support vector machines. *Sadhana*, Vol. 27(1), pp. 35-58.

[17] Bajaj, R., Dey, L. and Chaudhury, S., 2002. Devnagari numeral recognition by combining decision of multiple connectionist classifiers. *Sadhana*, Vol. 27(1), pp. 59-72.

[18] Bansal, V. and Sinha, R. M. K., 2000. Integrating knowledge sources in Devanagari text recognition system. *IEEE Transactions on Systems, Man and Cybernetics - Part A*, Vol. 30(4), pp. 500-505.

[19] Bansal, V. and Sinha, R. M. K., 2002. Segmentation of touching and fused Devanagri characters. *Pattern Recognition*, Vol. 35(4), pp. 875-893.

[20] Basu, S., Chaudhuri, C., Kundu, M., Nasipuri, M. and Basu, D. K., 2007. Text line extraction from multi-skewed handwritten documents. *Pattern Recognition*, Vol. 40(6), pp. 1825-1839.

[21] Basu, S., Das, N., Sarkar, R., Kundu, M., Nasipuri, M. and Basu, D. K., 2009. A hierarchical approach to recognition of handwritten Bangla characters. *Pattern Recognition*, Vol. 42(7), pp. 1467-1484

[22] Belhe, S., Paulzagade, C., Deshmukh, A., Jetley, S. and Mehrotra, K., 2012. Hindi handwritten word recognition using HMM and symbol tree. *Proceedings of the Workshop on Document Analysis and Recognition* (*DAR*), pp. 9-14.

[23] Bharath, A. and Madhvanath, S., 2011. HMM-based lexicon-driven and lexicon-free word recognition for online handwritten Indic scripts. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (*PAMI*), Vol. 34(4), pp. 670-682.

[24] Bhattacharya, U. and Chaudhuri, B. B., 2003. A majority voting scheme for multi-resolution recognition of handprinted numerals. *Proceedings of 7th International Conference on Document Analysis and Recognition* (*ICDAR*), pp. 16-20.

[25] Bhattacharya, U., Vajda, S., Mallick, A., Chaudhuri, B. B. and Belaid, A., 2004. On the choice of training set, architecture and combination rule of multiple MLP classifiers for multiresolution recognition of handwritten characters. *Proceedings of 9th International Workshop on Frontiers in Handwriting Recognition* (*IWFHR*), pp. 419-424.

[26] Bhattacharya, U., Shridhar, M. and Parui, S. K., 2006. On recognition of handwritten Bangla characters. *Proceedings of International Conference on Computer Vision, Graphics and Image Processing* (*ICVGIP*), pp. 817-828.

[27] Bhattacharya, U., Gupta, B. K. and Parui, S. K., 2007. Direction code based features for recognition of online handwritten characters of Bangla. *Proceedings of 9th*

`

International Conference on Document Analysis and Recognition (*ICDAR*), Vol. 1, pp. 58-62.

[28] Bhowmik, T. K., Bhattacharya, U. and Parui, S. K., 2004. Recognition of Bangla handwritten characters using an MLP classifier based on stroke features. *Proceedings of International Conference on Neural Information Processing* (*ICONIP'04*), pp. 814-819.

[29] Bhowmik, T. K., Parui, S. K., Bhattacharya, U. and Shaw, B., 2006. An HMM based recognition scheme for handwritten Oriya numerals. *Proceedings of 9th International Conference on Information Technology* (*ICIT*), pp. 105-110.

[30] Bhowmik, T. K., Ghanty, P., Roy, A. and Parui, S. K., 2009. SVM-based hierarchical architectures for handwritten Bangla character recognition. *International Journal on Document Analysis and Recognition* (*IJDAR*), Vol. 12, pp. 97-108.

[31] Bishnu, A. and Chaudhuri, B. B., 1999. Segmentation of Bangla handwritten text into characters by recursive contour following. *Proceedings of 5th International Conference on Document Analysis and Recognition* (*ICDAR*), pp. 402-405.

[32] Bunke, H. and Varga, T., 2007. Off-Line Roman cursive handwriting recognition. *Advances in Pattern Recognition*, pp. 165-183.

[33] Chaudhuri, B. B., Pal, U. and Mitra, M., 2002. Automatic recognition of printed Oriya script. *Sadhana*, Vol. 27(1), pp. 23-34.

[34] Chaudhuri, B. B. and Majumdar, A., 2007. Curvelet-based multi SVM recognizer for offline handwritten Bangla: A major Indian script. *Proceedings of 9th International Conference on Document Analysis and Recognition* (*ICDAR*), pp. 491-495.

[35] Chacko, B. P. and Anto, B. P., 2010a. Discrete curve evolution based skeleton pruning for character recognition. *International Journal of Multimedia, Computer Vision and Machine Learning*, Vol. 1(1), pp. 73-81.

[36] Chacko, B. P. and Anto, B. P., 2010b. Pre and Post processing approaches in edge detection for character recognition. *Proceedings of 12th International Conference on the Frontiers of Handwriting Recognition* (*ICFHR*), pp. 676-681.

[37] Cho, S. J. and Kim, J. H., 2001. Bayesian network modeling of strokes and their relationships for on-line handwriting recognition. *Proceedings of 6th International Conference on Document Analysis and Recognition* (*ICDAR*), pp. 86-90.

[38] Dash, M., Liu, H. and Motoda, H., 2003. Consistency based feature selection. *Artificial Intelligence*, Vol. 151, pp. 155-176.

[39] Deepu, V., Sriganesh, M. and Ramakrishnan, A. G., 2004. Principal component analysis for online handwritten character recognition. *Proceedings of 17th International Conference on Pattern Recognition* (*ICPR*), Vol. 2, pp. 327-330.

[40] Dengel, A. R. and Klein, B., 2002. A requirements-driven system for document analysis and understanding. *Proceedings of 5th International Workshop on Document Analysis and Systems* (*IWDAS*), pp. 433-444.

[41] Desai, A. A., 2010. Gujarati handwritten numeral optical character reorganization through neural network. *Pattern Recognition*, Vol. 43(7), pp. 2582-2589.

[42] Dholakia, J., Negi, A. and Mohan, S. R., 2005. Zone identification in the printed Gujarati text. *Proceedings of* 8th *International Conference on Document Analysis and Recognition* (*ICDAR*), Vol. 1, pp. 272-276.

[43] Dimauro, G., Impedovo, S., Modugno, R., Pirlo, G. and Sarcinella, L., 2002. Analysis of stability in hand-written dynamic signatures. *Proceedings of 8th International Workshop on Frontiers in Handwriting Recognition* (*IWFHR*), pp. 259-263.

[44] Dutta, A. and Chaudhury, S., 1993. Bengali alpha-numeric character recognition using curvature features. *Pattern Recognition*, Vol. 26(12), pp.1757-1770.

[45] Fink, G. A., Vajda, S., Bhattacharya, U., Parui, S. K. and Chaudhuri, B. B., 2010. Online Bangla word recognition using sub-stroke level features and Hidden Markov Models. *Proceedings of 12th International Conference on Frontiers in Handwriting Recognition* (*ICFHR*), pp. 393-398.

[46] Gader, P. D., Mohamed, M. and Chiang, J. H., 1997. Handwritten word recognition with character and inter-character neural networks. *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, Vol. 27(1), pp. 158-164.

`

[47] Garain, U., Chaudhuri, B. B. and Pal, T. T., 2002. Online handwritten Indian script recognition: A human motor function based framework. *Proceedings of 16th International Conference on Pattern Recognition* (*ICPR*), Vol. 3, pp. 164-167.

[48] Garg, N. K. and Jindal, S., 2007. An efficient feature set for handwritten digit recognition. *Proceedings of 15th International Conference on Advanced Computing and Communications* (*ADCOM*), pp. 540-544.

[49] Garg, N. K., Kaur, L. and Jindal, M. K., 2010. A new method for line segmentation of handwritten Hindi text. *Proceedings of 7th International Conference on Information Technology: New Generations* (*ITNG*), pp. 392-397.

[50] Gatos, B., Antonacopoulos, A. and Stamatopoulos, N., 2007. ICDAR2007 Handwriting segmentation contest. *Proceedings of 9th International Conference on Document Analysis and Recognition* (*ICDAR*), pp. 1284-1288.

[51] Grosicki, E. and Abed, H. E., 2009. ICDAR 2009 Handwriting recognition competition. *Proceedings of 10th International Conference on Document Analysis and Recognition* (*ICDAR*), pp. 1398-1402.

[52] Hall, M. A., 1998. Correlation-based feature selection for machine learning. *PHD Thesis, Department of Computer Science, Waika to University, Hamilton, New Zealand*

[53] Hanmandlu, M., Murthy, O. V. R. and Madasu, V. K., 2007. Fuzzy model based recognition of handwritten Hindi characters. *Proceedings of 9th Biennial Conference of the Australian pattern recognition society on Digital Image Computing and Techniques and Applications*, pp. 454-461.

[54] Heutte, L., Paquet, T., Moreau, J. V., Lecourtier, Y. and Olivier, C., 1998. A structural/statistical feature based vector for handwritten character recognition. *Pattern Recognition Letters*, Vol. 19(7), pp. 629-641.

[55] Hewavitharana, S. and Fernando, H. C., 2002. A two-stage classification approach to Tamil handwriting recognition. *Proceedings of the Tamil Internet Conference*, pp. 118-124.

[56] Impedovo, S. and Dimauro, G., 1990. An interactive system for the selection of handwritten numeral classes. *Proceedings of 10th International Conference on Pattern Recognition* (*ICPR*), pp. 563-566.

[57] Impedovo, S., Marangelli, B. and Plantamura, V. L., 1976. Real-time recognition of handwritten numerals. *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 6(2), pp. 145-148.

[58] Impedovo, S., Modugno, R. and Pirlo, G., 2010. Membership functions for zoning-based recognition of handwritten digits. *Proceedings of 20th International Conference on Pattern Recognition* (*ICPR*), pp. 1876-1879.

[59] Izadi, S., Sadri, J., Solimanpour, F. and Suen, C. Y., 2006. A review on Persian script and recognition techniques. *Proceedings of Conference on Arabic and Chinese Handwriting*, pp. 22-35.

[60] Jayadevan, R., Pal, U. and Kimura, F., 2010. Recognition of words from legal amounts of Indian bank cheques. *Proceedings of 12th International Conference on Frontiers in Handwriting Recognition* (*ICFHR*), pp. 166-171.

[61] Jindal, M. K., Lehal, G. S. and Sharma, R. K., 2005. Segmentation problems and solutions in printed degraded Gurmukhi script. *International Journal of Signal Processing*, Vol. 2(4), pp. 258-267.

[62] Jindal, M. K., Sharma, R. K. and Lehal, G. S., 2007. Segmentation of horizontally overlapping lines in printed Indian scripts. *International Journal of Computational Intelligence Research*, Vol. 3(4), pp. 277-286.

[63] Jindal, M. K., Sharma, R. K. and Lehal, G. S., 2008. Structural features for recognizing degraded printed Gurmukhi script. *Proceedings of the 5th International Conference on Information Technology: New Generations* (*ITNG*), pp. 668-673.

[64] Jindal, M. K., Lehal, G. S. and Sharma, R. K., 2009a. On segmentation of touching characters and overlapping lines in degraded printed Gurmukhi script. *International Journal of Image and Graphics* (*IJIG*), Vol. 9(3), pp. 321-353.

[65] Jindal, M. K., Lehal, G. S. and Sharma, R. K., 2009b. Segmentation of touching characters in upper zone in printed Gurmukhi script. *Proceedings of 2nd Bangalore*

`

Annual Compute Conference (Bangalore, India, January 09 - 10, 2009). COMPUTE '09. ACM, New York, NY, 1-6. DOI http://doi.acm.org/10.1145/1517303.1517313.

[66] John, R., Raju, G. and Guru, D. S., 2007. 1D wavelet transform of projection profiles for isolated handwritten Malayalam character recognition. *Proceedings of International Conference on Computational Intelligence and Multimedia Applications* (*ICCIMA*), Vol. 2, pp. 481-485.

[67] Joshi, N., Sita, G., Ramakrishnan, A. G. and Madhvanath, S., 2004a. Tamil handwriting recognition using subspace and DTW based classifiers. *Proceedings of 11th International Conference on Neural Information Processing* (*ICONIP*), pp. 806-813.

[68] Joshi, N., Sita, G., Ramakrishnan, A. G. and Madhvanath, S., 2004b. Comparison of elastic matching algorithms for online Tamil handwritten character recognition. *Proceedings of 9th International Conference on Frontiers in Handwriting Recognition* (*ICFHR*), pp. 444-449.

[69] Joshi, N., Sita, G., Ramakrishnan, A. G., Deepu, V. and Madhvanath, S., 2005. Machine recognition of online handwritten Devanagari characters. *Proceedings of 8th International Conference on Document Analysis and Recognition* (*ICDAR*), Vol. 2, pp. 1156-1160.

[70] Kacem, A., Aouiti, N. and Belaid, A., 2012. Structural features extraction for handwritten Arabic personal names recognition. *Proceedings of International Conference on Frontiers in Handwriting Recognition* (*ICFHR*), pp. 268-273.

[71] Karnchanapusakij, C., Suwannakat, P., Rakprasertsuk, W. and Dejdumrong, N., 2009. Online handwriting Thai character recognition. *Proceedings of 6th International Conference on Computer Graphics, Imaging and Visualization* (*CGIV*), pp. 323-328.

[72] Kumar, D., 2008. AI approach to hand written Devanagri script recognition. *Proceedings of IEEE Region 10th International Conference on EC3- Energy, Computer, Communication and Control Systems*, Vol. 2, pp. 229-237.

[73] Kunte, R. S. and Samuel, R. D. S., 2006. Script independent handwritten numeral recognition. *Proceedings of the International Conference on Visual Information Engineering*, pp. 94-98.

[74] Kunte, R. S. and Samuel, R. D. S., 2007. A simple and efficient optical character recognition system for basic symbols in printed Kannada text. *Sadhana*, Vol. 32(5), pp. 521-533.

[75] Lajish, V. L., 2007. Handwritten character recognition using perceptual fuzzy-zoning and class modular neural networks. *Proceedings of 4th International Conference on Innovations in Information Technology* (*ICIIT*), pp. 188-192.

[76] Lajish, V. L., 2008. Handwritten character recognition using gray-scale based state-space parameters and class modular NN. *Proceedings of International Conference on Signal Processing, Communications and Networking* (*ICSCN*), pp. 374-379.

[77] Lajish, V. L. and Kopparapu, S. K., 2010. Fuzzy directional features for unconstrained on-line Devanagari handwriting recognition. *Proceedings of National Conference on Communications* (*NCC*), pp. 1-5.

[78] Lehal, G. S. and Singh, C., 1999. Feature extraction and classification for OCR of Gurmukhi script. *Vivek*, Vol. 12(2), pp. 2-12.

[79] Lehal, G. S., Singh, C. and Lehal, R., 2001. A shape based post processor for Gurmukhi OCR. *Proceedings of 6th International Conference on Document Analysis and Recognition* (*ICDAR*), pp. 1105-1109.

[80] Lehal, G. S. and Singh, C., 2002. A post-processor for Gurmukhi OCR. *Sadhana*, Vol. 27(1), pp. 99-111.

[81] Lehal, G. S. and Singh, C., 2006. A complete machine printed Gurmukhi OCR system. *Vivek*, Vol. 16(3), pp. 10-17.

[82] Lemaitre, A. and Camillerapp, J., 2006. Text line extraction in handwritten document with Kalman filter applied on low resolution image. *Proceedings of 2nd International Conference on Digital Image Analysis for Libraries* (*DIAL*), pp. 38-45.

`

[83] Li, Y., Zheng, Y., Doermann, D. and Jaeger, S., 2006. A new algorithm for detecting text line in handwritten documents. *Proceedings of International Workshop on Frontiers in Handwriting Recognition* (*IWFHR*), pp. 35-40.

[84] Li, Z. C., Li, H. J., Suen, C. Y., Wang, H. Q. and Liao, S. Y., 2002. Recognition of handwritten characters by parts with multiple orientations. *Mathematical and Computer Modelling*, Vol. 35(3-4), pp. 441-479.

[85] Liwicki, M. and Bunke, H., 2007. Combining on-line and off-line systems for handwriting recognition. *Proceedings of International Conference on Document Analysis and Recognition* (*ICDAR*), pp. 372-376.

[86] Lorigo, L. M. and Govindaraju, V., 2006. Offline Arabic handwriting recognition: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (*PAMI*), Vol. 28(5), pp. 712-724.

[87] Lu, S., Tu, X. and Lu, Y., 2008. An improved two-layer SOM classifier for handwritten numeral recognition. *Proceedings of the International Conference on Intelligent Information Technology*, pp. 367-371.

[88] Marukatat, S., Sicard, R., Artieres, T. and Gallinari, P., 2003. A flexible recognition engine for complex on-line handwritten character recognition. *Proceedings of 7th International Conference on Document Analysis and Recognition* (*ICDAR*), pp. 1048-1052.

[89] Mondal, T., Bhattacharya, U., Parui, S. K., Das, K. and Mandalapu, D., 2010. On-line handwriting recognition of Indian scripts-the first benchmark. *Proceedings of 12th International Conference on Frontiers in Handwriting Recognition* (*ICFHR*), pp. 200-205.

[90] Moni, B. S. and Raju, G., 2011. Modified quadratic classifier and directional features for handwritten Malayalam character recognition. *International Journal of Computer Applications*, pp. 30-34.

[91] Mori, S., Suen, C. Y. and Yamamoto, K., 1992. Historical review of OCR research and development. *Proceedings of the IEEE*, Vol. 80(7), pp.1029-1058.

[92] Mozaffari, S., Faez, K., Faradji, F., Ziaratban, M. and Golzan, S. M., 2006. A comprehensive isolated Farsi/Arabic character database for handwritten OCR research. *Pattern Recognition and Image Processing Laboratory Electrical Engineering Department*, Amirkabir University of Technology, Tehran, Iran, pp. 385-389.

[93] Nakagawa, M., Zhu, B. and Onuma, M., 2005. A model of on-line handwritten Japanese text recognition free from line direction and writing format constraints. *IEICE Transactions on Information and Systems*, Vol. E88-D(8), pp. 1815-1822.

[94] Namboodiri, A. M. and Jain, A. K., 2004. Online handwritten script recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (*PAMI*), Vol. 26(1), pp. 124-130.

[95] Nguyen, V., Blumenstein, M., Muthukkumarasamy, V. and Leedham, G., 2007. Off-line signature verification using enhanced modified direction features in conjunction with neural classifiers and support vector machines. *Proceedings of 9th International Conference on Document Analysis and Recognition* (*ICDAR*), pp. 734-738.

[96] Padma, M. C. and Vijaya, P. A., 2009. Monothetic separation of Telugu, Hindi and English text lines from a multi script document. *IEEE International Conference on Systems, Man and Cybernetics*, pp. 4870-4875.

[97] Pal, U. and Chaudhuri, B. B., 1994. OCR in Bangla: an Indo-Bangladeshi language. *Proceedings of 12th International Conference on Pattern Recognition* (*ICPR*), pp. 269-273.

[98] Pal, U. and Chaudhuri, B. B., 2004. Indian script character recognition: a survey. *Pattern Recognition*, Vol. 37, pp. 1887-1899.

[99] Pal, U. and Datta, S., 2003. Segmentation of Bangla unconstrained handwritten text. *Proceedings of 7th International Conference on Document Analysis and Recognition* (*ICDAR*), pp. 1128-1132.

[100] Pal, U., Belaid, A. and Choisy, Ch., 2003. Touching numeral segmentation using water reservoir concept. *Pattern Recognition Letters*, Vol. 24(1-3), pp. 261-272.

`

[101] Pal, U., Roy, K. and Kimura, F., 2006a. A lexicon driven method for unconstrained Bangla handwritten word recognition. *Proceedings of 10th International Workshop on Frontiers in Handwriting Recognition* (*IWFHR*), pp. 601-606.

[102] Pal, U., Belaid, A. and Chaudhuri, B. B., 2006b. A system for Bangla handwritten numeral recognition. *IETE Journal of Research*, Vol. 52(1), pp. 27-34.

[103] Pal, U., Wakabayashi, T. and Kimura, F., 2007a. Handwritten Bangla compound character recognition using gradient feature. *Proceedings of 10th International Conference on Information Technology* (*ICIT*), pp. 208-213.

[104] Pal, U., Sharma, N., Wakabayashi, T. and Kimura, F., 2007b. Handwritten numeral recognition of six popular Indian scripts. *Proceedings of 9th International Conference on Document Analysis and Recognition* (*ICDAR*), Vol. 2, pp. 749-753.

[105] Pal, U., Sharma, N., Wakabayashi, T. and Kimura, F., 2007c. Off-line handwritten character recognition of Devanagari script. *Proceedings of 9th International Conference on Document Analysis and Recognition* (*ICDAR*), pp. 496-500.

[106] Pal, U., Wakabayashi, T. and Kimura, F., 2007d. A system for off-line Oriya handwritten character recognition using curvature feature. *Proceedings of 10th International Conference on Information Technology* (*ICIT*), pp. 227-229.

[107] Pal, U., Roy, K. and Kimura, F., 2008. Bangla handwritten pin code string recognition for Indian postal automation. *Proceedings of 11th International Conference on Frontiers in Handwriting Recognition* (*ICFHR*), pp. 290-295.

[108] Pal, U., Wakabayashi, T. and Kimura, F., 2009a. Comparative study of Devanagari handwritten character recognition using different feature and classifiers. *Proceedings of 10th International Conference on Document Analysis and Recognition* (*ICDAR*), pp. 1111-1115.

[109] Pal, U., Roy, R. K., Roy, K. and Kimura, F., 2009b. Indian multi-script full pin-code string recognition for postal automation. *Proceedings of 10th International Conference on Document Analysis and Recognition* (*ICDAR*), pp. 456-460.

[110] Pal, U., Roy, R. K. and Kimura, F., 2010. Bangla and English city name recognition for Indian postal automation. *Proceedings of 20th International Conference on Pattern Recognition* (*ICPR*), pp. 1985-1988.

[111] Park, J., Govindaraju, V. and Srihari, S. N., 2000. OCR in a hierarchical feature space. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (*PAMI*), Vol. 22(4), pp. 400-407.

[112] Parui, S. K., Chaudhuri, B. B. and Majumder, D. D., 1982. A procedure for recognition of connected handwritten numerals. *International Journal Systems Science*, Vol. 13(9), pp. 1019-1029.

[113] Paulpandian, T. and Ganapathy, V., 1993. Translation and scale invariant recognition of handwritten Tamil characters using a hierarchical neural network. *Proceedings of the IEEE International Symposium on Circuits and Systems* (*ISCAS*), pp. 2439-2441.

[114] Pirlo, G. and Impedovo, D., 2011. Fuzzy zoning based classification for handwritten characters. *IEEE Transactions on Fuzzy Systems*, Vol. 19(4), pp. 780-785.

[115] Plamondon, R. and Srihari, S. N., 2000. On-line and off-line handwriting recognition: A comprehensive survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (*PAMI*), Vol. 22(1), pp. 63-84.

[116] Pradeep, J., Srinivasan, E. and Himavathi, S., 2011. Diagonal based feature extraction for handwritten Alphabets recognition system using neural network. *International Journal of Computer Science and Information Technology*, Vol. 3(1), pp. 27-38.

[117] Prasad, J. R., Kulkarni, U. V. and Prasad, R. S., 2009. Offline handwritten character recognition of Gujarati script using pattern matching. *Proceedings of 3rd International Conference on Anti-counterfeiting, Security, and Identification in Communication* (*ASID*), pp. 611-615.

[118] Prasanth, L., Babu, J. V., Sharma, R. R., Rao, P. G. V. and Manadalapu, D., 2007. Elastic matching of online handwritten Tamil and Telugu scripts using local features. *Proceedings of 9th International Conference on Document Analysis and Recognition* (*ICDAR*), Vol. 2, pp. 1028-1032.

`

[119] Purkait, P. and Chanda, B., 2010. Off-line recognition of handwritten Bengali numerals using morphological features. *Proceedings of the 12th International Conference on the Frontiers of Handwriting Recognition* (*ICFHR*), pp. 363-368.

[120] Rahiman, M. A., Shajan, A., Elizabeth, A., Divya, M. K., Kumar, G. M. and Rajasree, M. S., 2010. Isolated handwritten Malayalam character recognition using HLH intensity patterns. *Proceedings of 2nd International Conference on Machine Learning and Computing* (*ICMLC*), pp. 147-151.

[121] Rajashekararadhya, S. V. and Ranjan, P. V., 2008. Neural network based handwritten numeral recognition of Kannada and Telugu scripts. *Proceedings of 10th IEEE International Conference on TENCON*, pp. 1-5.

[122] Rajashekararadhya, S. V. and Ranjan, P. V., 2009a. Support vector machine based handwritten numeral recognition of Kannada script. *Proceedings of IEEE International Conference on Advance Computing Conference* (*IACC*), pp. 381-386.

[123] Rajashekararadhya, S. V. and Ranjan, P. V., 2009b. Zone based feature extraction algorithm for handwritten numeral recognition of Kannada script. *Proceedings of IEEE International Conference on Advance Computing Conference* (*IACC*), pp. 525-528.

[124] Rajput, G. G. and Hangarge, M., 2007. Recognition of isolated handwritten Kannada numerals based on image fusion method. *Proceedings of International Conference on PReMI*, pp. 153-160.

[125] Raju, G., 2008. Wavelet transform and projection profiles in handwritten character recognition – A performance analysis. *Proceedings of ICADCOM*, pp. 309-314.

[126] Ramakrishnan, A. G. and Shashidhar, J., 2013. Development of OHWR system for Kannada. *VishwaBharat@tdil*, Vol. 39-40, pp. 67-95.

[127] Ramakrishnan, A. G., Bhargava, U. K., Sundaram, S. and Harshitha, P. V., 2013. Development of OHWR system for Tamil. *VishwaBharat@tdil*, Vol. 39-40, pp. 211-224.

[128] Rampalli, R. and Ramakrishnan, A. G., 2011. Fusion of complementary online and offline strategies for recognition of handwritten Kannada characters. *Journal of Universal Computer Science* (*JUCS*), Vol. 17(1), pp. 81-93.

[129] Reddy, G. S., Sharma, P., Prasanna, S. R. M, Mahanta, C. and Sharma, L. N., 2012a. Combined online and offline Assamese handwritten numeral recognizer. *Proceedings of 18th National Conference on Communications* (*NCC-2012*), IIT Kharagpur.

[130] Reddy, G. S., Sarma, B., Naik, R. K., Prasanna, S. R. M and Mahanta, C., 2012b. Assamese online handwritten digit recognition system using Hidden Markov Models. *Proceedings of Workshop on Document Analysis and Recognition* (*DAR'12*), Mumbai, pp. 108-113.

[131] Roy, K., Banerjee, A. and Pal, U., 2004a. A system for word-wise handwritten script identification for Indian postal automation. *Proceedings of 1st IEEE INDICON*, pp. 266-271.

[132] Roy, K., Vajda, S., Pal, U. and Chaudhuri, B. B., 2004b. A system towards Indian postal automation. *Proceedings of 9th International Workshop on Frontiers in Handwriting Recognition* (*IWFHR*), pp. 580-585.

[133] Roy, K., Pal, T., Pal, U. and Kimura, F., 2005a. Oriya handwritten numeral recognition system. *Proceedings of 8th International Conference on Document Analysis and Recognition* (*ICDAR*), pp. 770-774.

[134] Roy, K., Chaudhuri, C., Pal, U. and Kundu, M., 2005b. A study on the effect of varying training set sizes on recognition performance with handwritten Bangla numerals. *Proceedings of 1st IEEE INDICON*, pp. 570-574.

[135] Roy, K. and Pal, U., 2006. Word-wise hand-written script separation for Indian postal automation. *Proceedings of the International Conference on Frontiers of Handwriting Recognition* (*ICFHR*), pp. 521-526.

[136] Roy, K., Alaei, A. and Pal, U., 2010. Word-wise handwritten Persian and Roman script identification. *Proceedings of 12th International Conference on Frontiers in Handwriting Recognition* (*ICFHR*), pp. 628-633.

[137] Sarma, B., Mehrotra, K., Naik, R. K., Prasanna, S. R. M., Belhe, S. and Mahanta, C., 2013. Handwritten Assamese numeral recognizer using HMM & SVM classifiers, *Proceedings of 19th National Conference on Communications* (*NCC 2013*), IIT Delhi.

`

[138] Sastry, P. N., Krishnan, R. and Ram, B. V. S., 2010. Classification and identification of Telugu handwritten characters extracted from palm leaves using decision tree approach. *ARPN Journal of Applied Engineering and Applied Science*, Vol. 5(3), pp. 22-32.

[139] Schomaker, L. and Segers, E. 1999. Finding features used in the human reading of cursive handwriting. *International Journal of Document Analysis and Recognition* (*IJDAR*), Vol. 2, pp. 13-18.

[140] Schomaker, L. 2007. Retrieval of handwritten lines in historical documents. *Proceedings of 9th International Conference on Document Analysis and Recognition* (*ICDAR*), pp. 594-598.

[141] Selamat, A. and Ching, N. C., 2008. Arabic script documents language identifications using Fuzzy ART. *Proceedings of 2nd Asia International Conference on Modelling & Simulation* (*AICMS*), pp. 528-533.

[142] Sethi, K. and Chatterjee, B., 1976. Machine recognition of constrained hand-printed Devanagari numerals. *J. Inst. Elec. Telecom. Engg*., Vol. 22, pp. 532-535.

[143] Shanthi, N. and Duraiswamy, K., 2010. A novel SVM based handwritten Tamil character recognition system. *Pattern Analysis and Applications* (*PAA*), Vol. 13(2), pp. 173-180.

[144] Shapiro, V., Gluhchev, G. and Sgurev, V., 1993. Handwritten document image segmentation and analysis. *Pattern Recognition Letters*, Vol. 14(1), pp. 71-78.

[145] Sharma, D. V. and Lehal, G. S., 2006. An iterative algorithm for segmentation of isolated handwritten words in Gurmukhi script. *Proceedings of 18th International Conference on Pattern Recognition* (*ICPR*), Vol. 2, pp. 1022-1025.

[146] Sharma, A., Kumar, R. and Sharma, R. K., 2008. Online handwritten Gurmukhi character recognition using elastic matching. *Proceedings of Congress on Image and Signal Processing*, pp. 391-396.

[147] Sharma, D. V., Lehal, G. S. and Mehta, S., 2009. Shape encoded post processing of Gurmukhi OCR. *Proceedings of 10th International Conference on Document Analysis and Recognition* (*ICDAR*), pp. 788-792.

[148] Sharma, D. V. and Jhajj, P., 2010. Recognition of isolated handwritten characters in Gurmukhi script. *International Journal of Computer Applications*, Vol. 4(8), pp. 9-17.

[149] Sharma, N., Pal, U. and Kimura, F., 2006a. Recognition of handwritten Kannada numerals. *Proceedings of 9th International Conference on Information Technology* (*ICIT*), pp. 133-136.

[150] Sharma, N., Pal, U., Kimura, F. and Pal, S., 2006b. Recognition of off-line handwritten Devanagari characters using quadratic classifier. *Proceedings of International Conference on Computer Vision, Graphics and Image Processing* (*ICVGIP*), pp. 805-816.

[151] Shelke, S. and Apte, S., 2011. A multistage handwritten Marathi compound character recognition scheme using neural networks and wavelet features. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, Vol. 4(1), pp. 81-94.

[152] Shi, Z. and Govindaraju, V., 2004. Line separation for complex document images using fuzzy run length. *Proceedings of International Workshop on Digital Image Analysis for Libraries* (*DIAL*), pp. 306-312.

[153] Shi, Z., Setlur, S. and Govindaraju, V., 2005. Text extraction from gray scale historical document images using adaptive local connectivity map. *Proceedings of International Conference on Document Analysis and Recognition* (*ICDAR*), Vol. 2, pp. 794-798.

[154] Sreeraj, M. and Idicula, S. M., 2010. *k*-NN based on-line handwritten character recognition system. *Proceedings of 1st International Conference on Integrated Intelligent Computing* (*ICIIC*), pp. 171-176.

[155] Srihari, S. N., 1993. Recognition of handwritten and machine-printed text for postal address interpretation. *Journal on Pattern Recognition Letters - Postal processing and character recognition archive*, Vol. 14(4), pp. 291-302.

[156] Srihari, S. N. and Leedham, G., 2003. A survey of computer methods in forensic handwritten document examination. *Proceedings of 11th Conference of the International Graphonomics Society* (*IGS*), pp. 278-281.

`

[157] Su, B., Lu, S., Phan, T. Q. and Tan, C. L., 2012. Character extraction in web image for text recognition. *Proceedings of 21st International Conference on Pattern Recognition* (*ICPR*), pp. 3042-3045.

[158] Sulem, L. L. and Faure, C., 1994. Extracting text lines in handwritten documents by perceptual grouping. *Proceedings of Advances in handwriting and drawing: a multidisciplinary approach*, pp. 41-52.

[159] Sundaram, S. and Ramakrishnan, A. G., 2008. Two dimensional principal component analysis for online character recognition. *Proceedings of 11th International Conference on Frontiers in Handwriting Recognition* (*ICFHR*), pp. 88-94.

[160] Sundaram, S. and Ramakrishnan, A. G., 2013. Attention-feedback based robust segmentation of online handwritten isolated Tamil words. *ACM Transactions on Asian Language Information Processing* (*TALIP*), Vol. 12(1), Article no. 4.

[161] Sundaram, S. and Ramakrishnan, A. G., 2014. Performance enhancement of online handwritten Tamil symbol recognition with reevaluation techniques. *Pattern Analysis and Applications* (*PAA*), Vol. 17(3), pp. 587-609.

[162] Suresh, R. M. and Arumugam, S., 2007. Fuzzy technique based recognition of handwritten characters. *Image and Vision Computing*, Vol. 25(2), pp. 230-239.

[163] Sutha, J. and Ramaraj, N., 2007. Neural network based offline Tamil handwritten character recognition system. *Proceedings of ICCIMA*, pp. 446-450.

[164] Tapia, E. and Rojas, R., 2003. Recognition of on-line handwritten mathematical formulas in the E-chalk system. *Proceedings of 7th International Conference on Document Analysis and Recognition* (*ICDAR*), pp. 980-984.

[165] Tapia, E. and Rojas, R., 2005. Recognition of on-line handwritten mathematical expressions in the E-chalk system - An Extension. *Proceedings of 8th International Conference on Document Analysis and Recognition* (*ICDAR*), Vol. 2, pp. 1206-1210.

[166] Tran, D. C., Franco, P. and Ogier, J., 2010. Accented handwritten character recognition using SVM-application to French. *Proceedings of 12th International Conference on Frontiers in Handwriting Recognition* (*ICFHR*), pp. 65-71.

[167] Tripathy, N. and Pal, U., 2004. Handwriting segmentation of unconstrained Oriya text. *Proceedings of 9th International Workshop Frontiers in Handwriting Recognition* (*IWFHR*), pp. 306-311.

[168] Venkatesh, N. and Ramakrishnan, A. G., 2011. Choice of classifiers in hierarchical recognition of online handwritten Kannada and Tamil aksharas. *Journal of Universal Computer Science* (*JUCS*), Vol. 17, pp. 94-106.

[169] Wang, X., Govindaraju, V. and Srihari, S. 2000. Holistic recognition of handwritten character pairs. *Pattern Recognition*, Vol. 33(12-33), pp. 1967-1973.

[170] Wong, K. Y., Casey, R. G. and Wahl, F. M., 1982. Document analysis system. *IBM Journal of Research Development*, Vol. 26(6), pp. 647-656.

[171] Yanikoglu, B. and Sandon, P. A., 1998. Segmentation of off-line cursive handwriting using linear programming. *Pattern Recognition*, Vol. 31(12), pp. 1825-1833.

[172] Zhang, T. Y. and Suen, C. Y., 1984. A fast parallel algorithm for thinning digital patterns. *Communications of the ACM*, Vol. 27(3), pp. 236-239.

[173] Zhou, P., Li, L. and Tan, C. L., 2009. Character recognition under severe perspective distortion. *Proceedings of 10th International Conference on Document Analysis and Recognition* (*ICDAR*), pp. 676-680.

[174] Zhu, B., Zhou, X. D., Liu, C. L. and Nakagawa, M., 2010. A robust model for on-line handwritten Japanese text recognition. *International Journal on Document Analysis and Recognition* (*IJDAR*), Vol. 13(2), pp. 121-131.

`