Online Handwritten Gurmukhi Character Recognition using Support Vector Machine

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Technology
in
Computer Science and Application

Submitted By
Dushyant Khurana
(Roll No. 601103005)

Under the supervision of
Dr. R.K. Sharma
Professor, SMCA

SCHOOL OF MATHEMATICS AND COMPUTER APPLICATIONS
THAPAR UNIVERSITY
PATIALA - 147004

July 2013

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled, **"Online Handwritten Gurmukhi Character Recognition using Support Vector Machine"**, in partial fulfillment of the requirements for the award of degree of Master of Technology in Computer Science and Applications submitted in School of Mathematics and Computer Applications, Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of **Dr. R. K. Sharma, Professor, School of Mathematics and Computer Applications.**

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

(Dushyant Khurana)

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

(Dr. R. K. Shrama) 15.7.13
Professor
SMCA

Countersigned by:

25/7/2013

(Dr. Rajesh Kumar)
Head,
School of Mathematics and Computer Applications
Thapar University,
Patiala.

(Dr. S. K. Mohapatra)
Dean (Academic Affairs)
Thapar University,
Patiala

ACKNOWLEDGEMENTS

First of all I would like to thank the Almighty, who has always guided me to work on the right path of the life.

This work would not have been possible without the encouragement and able guidance of my supervisor **Dr. R. K. Sharma**. I thank my supervisor for their time, patience, discussions and valuable comments. Their enthusiasm and optimism made this experience both rewarding and enjoyable.

I am equally grateful to **Dr. Rajesh Kumar**, Associate Professor and Head, School of Mathematics and Computer Applications, for motivation and inspiration that triggered me for the thesis work.

I will be failing in my duty if I don't express my gratitude to **Dr. S. K. Mohapatra,** Senior Professor and Dean of Academic Affairs, Thapar University, for making provisions of infrastructure such as library facilities, computer labs equipped with net facilities, immensely useful for the learners to equip themselves with the latest in the field.

I am also thankful to the entire faculty and staff members of School of Mathematics and Computer Applications Department for their direct-indirect help, cooperation, love and affection, which made my stay at Thapar University memorable.

Last but not least, I would like to thank my parents for their wonderful love and encouragement, without their blessings none of this would have been possible. I would also like to thank my brothers, since they insisted that I should do so. I would also like to thank my close friends for their constant support.

Date: July 14th, 2013

Place: Thapar University, Patiala

(Dushyant Khurana)

ABSTRACT

Communication is always an important part of our life, either in the form of speech or writing. Natural Handwriting is one of the easiest ways to exchange information. In this world of technology exchanging information between user and computer is of immense importance and input devices such as keyboard and mouse have limitations as they cannot provide natural handwriting as input. Online handwriting recognition system can be used as an easiest and natural way of communication between user and human computers. Therefore Pen-Based interfaces are becoming more and more popular and hence a lot of research is being done for recognition. Research work presented in this thesis aims to recognize character with higher accuracy written in Gurumukhi script using Support Vector Machine (SVM) by improving processes of pre-processing phase used for recognition. Gurumukhi is a script of Punjabi Language which is widely spoken across the globe. This thesis is divided into five chapters. A brief outline of each chapter is given in the following paragraphs.

First chapter of this report consists of introduction to online handwritten recognition system, issues in online handwritten recognition system overview of Gurmukhi script and literature review. Issues in online handwriting recognition system includes: handwriting style variations; constrained and unconstrained handwriting; personal, situational and material factors; writer dependent vs. writer independent recognition system. In literature review, a detailed literature survey on each phase of established procedure of online handwriting recognition has been presented.

Second chapter gives the detailed work carried out in three phases. They are data collection, pre-processing and feature extraction. In data collection phase, input handwritten strokes are collected is shown. Phases of pre-processing are discussed and algorithms are presented. In the end feature extraction is explained.

Third chapter describes the recognition techniques that can be used for online handwritten recognition system. In this work Support Vector Machine (SVM) is used as a classifier for the recognition. This chapter also illustrates the use of post-processing phase.

Fourth Chapter contains the results of the algorithm proposed in this thesis work and a comparison is made with the proposed algorithms in Agrawal, (2012). The cross validation testing has been done for calculating the accuracy and 3, 4 and 5 fold cross validation testing is applied on a sample of 30, 50 and 70 of each zone.

Finally, the result of the thesis is concluded in this chapter. Future scope of the work is also discussed.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

Abbreviations	Expanded Forms
PC	Personal Computer
PDA	Personal Digital Assistant
HMM	Hidden Markov Model
SVM	Support Vector Machine
SVC	Support Vector Classifier
SVR	Support Vector Regeressor
OCR	Optical Character Recognition
OHCR	Online Handwritten Character Recognition

CHAPTER 1

INTRODUCTION

In this era, everybody wants to perform one's tasks as easily as possible. Interchanging information between user and the computer can be done by natural handwriting and this seems to be the easiest way. Hence, the research on the topic of online handwriting recognition system is on boom. Most of the users now use touch screen mobile phones, tablets for their needs and ease but still they cannot use their own handwriting for communication with their phones even with this high processing capability of the computers and mobile phone specifically in Indian languages. In order to fully utilize the high processing capability of the computer or tablets, a user interface is needed which should not only be efficient but also natural to the user. This naturality can be there in terms of acceptance of input by computers in user's own handwriting in his own language.

In general it takes good amount of time to input data into a computer through keyboard and the one who is not literate cannot use it. Same is for the output devices that were of huge size. But now both input and output process can be done on a single compact device, where a user gives input through an electronic pen and finds the output simultaneously and hence the data can be recorded. Sometimes we need textual form of data and online handwritten recognition systems got placed which make it very much easier for the user to input the data that can be converted into its digital or text form without using the keyboard and the additional benefit is the data can be entered at a faster rate. Thus, the handwriting recognition field has great potential to improve the communication between the user and the computer. Handwriting recognition is the additional ability of a computer to interpret the data which is intelligibly handwriting input from various resources such as paper documents, electronic pen tablets, touch screen and other devices and then generate a description of the data in a desired format.

Handwriting recognition is in research for many decades and many researches are going on all over the world. Great advance have been made in this field and due to this reason, the usage and reliability of online handwriting based devices such as Tablet PCs and Personal Digital Assistants (PDAs) have increased a lot. There are

many applications of the handwriting recognition systems. Some examples are: this can act as a communication tool for entering information, tool for reading bank cheques fields, converting document into textual form, *etc.*

1.1 Classification of Handwriting Recognition

The classification of handwriting recognition can be done into two major categories shown in Figure 1.1.

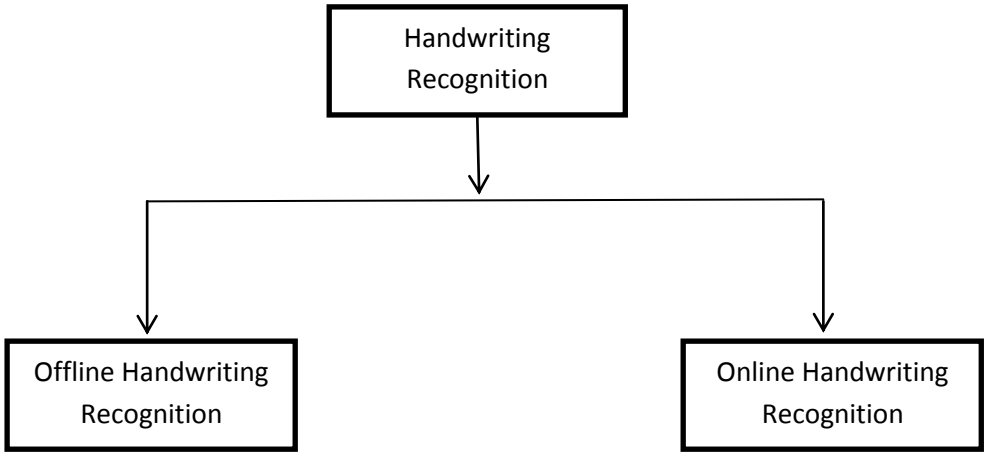


Figure 1.1: Classification of handwriting recognition

1.1.1 Offline Handwriting Recognition

Under this type recognition systems, the writing is available in the form of image which is captured with the help of a scanner or a camera which captures the writing optically. This means that the text is not recognised at the same time when it is produced but after the writer finishes writing, *i.e.*, in this case, the text is written on a surface such as paper and from there it is then passed to computer in the image form. It is then recognised by the handwriting recognition system. This is done in a sequential process: firstly the image is stored digitally in grey scale format, *e.g.*, bitmap image, and then further processing is done on it to get good recognition accuracy. Features for recognition can also be enhanced and then extracted from the stored bitmap image by applying desired functions. This type of recognition is called as Optical Character Recognition. Recognition of machine printed text or characters is also a part of OCR. Offline methods are less suitable for communication between man and machine because it has no real time interactivity. They are more suitable for automatic conversion of paper documents to electric documents which may be intercepted by computers.

1.1.2 Online Handwriting Recognition

In comparison to the offline methods of recognition, online handwriting recognition is done in real time, *i.e.*, at the same time as the handwriting is produced. The surface used for handwriting is usually electronic device called as digitized tablet and is used along with a pen called as "Stylus" with the help of which writer writes on it. As the pen moves on the surface of the digitized pad, the two dimensional coordinates will be captured and stored as a function of time as shown in Figure 1.2.

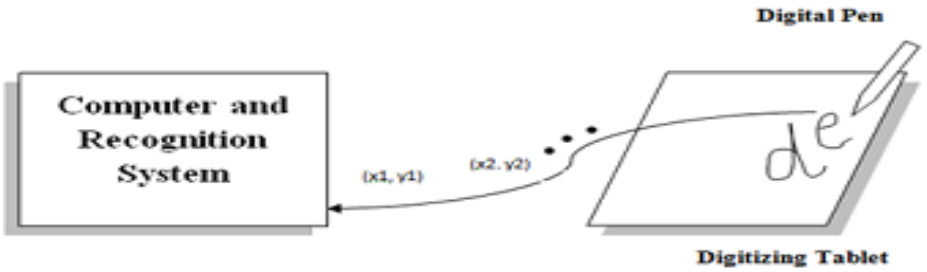


Figure 1.2: A tablet digitizer, input sampling and communication to the computer

1.1.3 Online Recognition vs. Offline Recognition

- Real Time Interactivity: Online Handwriting recognition is real time process, *i.e.*, recognition is done when data is written but in case of offline recognition is done after the writing is completed or at a later time.
- User Adaptation: In case of online methods, recognition can be corrected immediately. If a particular character is not being recognized properly, user can change his writing style then and there, and hence adapts to online method.
- Pre-processing: Pre-processing operations such as smoothing, normalization, detection of loops *etc*., is faster in case of online methods. Thus, lesser time is required for pre-processing operations in case of online handwriting as compared to offline handwriting.
- Easy Segmentation: As dynamic information is collected for each stroke in case of online recognition methods, therefore segmentation operations are much easier in online handwriting recognition as compared to offline.

- Special electronic devices are needed in case of online handwriting recognition *e.g.*, Digital tablets, PDAs, *etc.*, but in case of offline we need just a pen and paper. Offline methods are well suited for recognising printed documents.

In this work, we have focussed on improving of recognition accuracy by improving the processes of pre-processing. The stages of a handwriting recognition system, including the pre-processing stage are described in section 1.4. The present work has also focussed on the improvement of recognition accuracy of strokes that are used to form a Punjabi akshara. Next section introduces, in brief, the Gurmukhi script, which is used to write Punjabi language.

1.2 Introduction to Gurmukhi Script

Gurmukhi is the script used for writing Punjabi language in India and across the world which is the world's 14th most widely spoken language. The name Gurmukhi derived from the old Punjabi term "Gurmukhi", meaning "from the mouth of the Guru". Gurmukhi script is written in left-to-right direction and in top-down approach. Most of the characters have a horizontal line on the upper part by which characters of the word are connected. This is called as the headline. Gurmukhi has 41 consonants, 9 vowel symbols, and two symbols for nasal sound, 1 symbol which duplicates the sound of any consonant (addak), 3 subjoined forms of the consonant Rara, Haha and Vava. Table 1.1 shows these Gurmukhi characters.

The Gurmukhi script contains 35 letters. Out of which the three letters are distinct in Gurmukhi script as they form the basis of vowel and are distinct because they form the basis for vowels and are not consonants.

Table 1.1: Character Set of Gurmukhi Script

Character	**Character Name**
ੳ	URHA
ਅ	ERHA
ੲ	EERHI

ਸ	SUSSA (Sa)
ਹ	HAHA (Ha)
ਕ	KUKKA (Ka)
ਖ	KHUKHA (Kha)
ਗ	GUGGA (Ga)
ਘ	GHUGGA (Gha)
ਙ	UNGGA (Nga)
ਚ	CHUCHA (Ca)
ਛ	CHHUCHHA (Cha)
ਜ	JUJJA (Ja)
ਝ	JHUJHA (Jha)
ਞ	YANZA (Nya)
ਟ	TAINKA (Tta)
ਠ	THUTHA (Ttha)
ਡ	DUDDA (Dda)
ਢ	DHUDDA (Ddha)
ਣ	NAHNHA (Nna)
ਤ	TUTTA (Ta)
ਥ	THUTHA (Tha)
ਦ	DUDA (Da)
ਧ	DHUDDA (Dha)

ਨ	NUNNA (Na)
ਪ	PUPPA (Pa)
ਫ	PHUPHA (Pha)
ਬ	BUBBA (Ba)
ਭ	BHUBBA (Bha)
ਮ	MUMMA (Ma)
ਯ	YAIYYA (Ya)
ਰ	RARA (Ra)
ਲ	LULLA (La)
ਵ	VAVA (Va)
ੜ	RARA (Rha)

Out of these character first three are unique vowels, these characters are never used on their own. Remaining are the consonants of the Gurmukhi script.

Table 1.2: Unique Vowel Characters

ੳ	URHA
ਅ	ERHA
ੲ	EERHI

In addition to these, there are six consonants created by placing a dot (bindi) at the foot (pair) of the consonant these are shown in the Table 1.3.

Table 1.3: Six Special consonants in Gurmukhi

ਸ਼	SUSSA PAIR BINDI (Sha)
ਖ਼	KHUKHA PAIR BINDI (Khha)
ਗ਼	GUGGA PAIR BINDI (Ghha)
ਜ਼	JAJJA PAIR BINDI (Za)
ਫ਼	PHUPHA PAIR BINDI (Faa)
ਲ਼	LALLA PAIR BINDI (Lla)

1.2.1 Vowels in Gurmukhi Script

Gurmukhi follows similar concepts to other Brahmi scripts. All consonants are followed by an inherent ‘a’ sound. This inherent vowel sound can be changed by using dependent vowel signs which attach to bear consonants. In some cases, dependent vowel signs cannot be used at the beginning of a word or syllable for instance and so an independent vowel character is used instead. Table 1.4 shows a list of dependent vowels.

Table 1.4: Dependent vowel lists

ੁ	MUKTA (a)
ਾ	KANNA (aa)
ਿ	SIHARI (i)
ੀ	BIHARI (ii)
ੇ	LAVAN (ee)
ੈ	DULAVAN (ai)
ੂ	ONKAR (u)

◌ੂ	DULANKAR (uu)
◌ੋ	HORA (oo)
◌ੌ	KANAURA (au)

1.2.2 Zones in Gurmukhi Script

A word in Gurmukhi can be portioned into three zones these are:

Upper Zone: It is the region above the headline, where vowels reside

Middle Zone: It represents the area where the consonants and some other parts of vowels reside, *i.e.*, the area below the headline but above the lower zone.

Lower Zone: It represents the area below the middle zone where some vowels, halant or certain half character lie at the foot of the consonants. These zones are shown in Figure. 1.3 for the Gurmukhi word (ਦੁਲੰਕਾਰ).

Figure 1.3: Three zones and headline in Gurmukhi word

Characteristics of Gurmukhi Scripts

- Gurmukhi alphabet is syllabic, *i.e.*, all consonants have an inherent vowel.
- Unlike Roman characters, Gurmukhi is written below the line and it has no concept of lower and upper case.
- The Gurmukhi script, unlike the Greek and Roman alphabets, is arranged in a logical fashion: vowel first, then consonants and semivowels
- Upper zone and middle zone are separated by a line called head line or *siro rekha* as present in other Indian scripts also such as Devnagari.

1.3 Issues in Handwritten Character Recognition

The online handwriting recognition is used for identification of characters and it is used with the devices such as PDAs, touch screen phones, cross pad and tablet PCs, but due to variability in handwriting styles and distortions caused by digitizing process, even the best handwriting character recognizer is unreliable. Distortion can because of the electronic device used which takes the input and send to the recognizer and if the input is distorted good result cannot be gained. In order to use these input devices, the accuracy achieved by the handwriting recognizer must be sufficiently high so that it is acceptable by the user. Some of the other issues are explained, in brief, in next subsections.

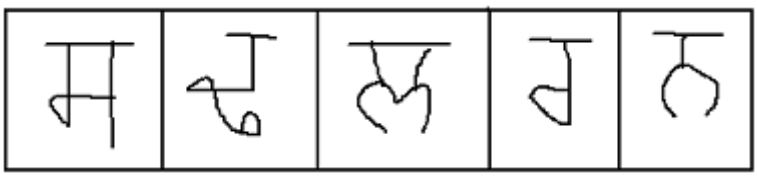
1.3.1 Handwriting Styles Variations

This is the major issue in recognition of a character as different writer has a different way of writing a character, means different writer different handwriting, different styles. Many times a person finds himself/herself unable to recognize his/her own handwriting. Hence, practically it is much difficult for a recognizer recognize handwriting efficiently. Variations like Deformed Geometry, slants, skews, overlapping, distortions are inserted by different writers in different ways. Geometric properties like aspect ratio, position and size vary. Some writer writes very small some writes very big characters. Figure 1.4 shows a sample of variations in five writers writing the same character of the Gurmukhi script although such samples share a high degree of similarities. The character may look similar but the no. of strokes, direction and the drawing may vary considerably.

Figure 1.4: Variation in some of the Gurmukhi Characters by five writers

1.3.2 Constrained and Unconstrained Handwriting

Handwriting styles could be constrained or unconstrained. Constrained handwriting is boxed discrete or spaced discrete in nature. In boxed discrete handwriting, unconstrained handwriting is cursive or mixed cursive Figure 1.4 shows the constrained and unconstrained handwriting as the writing changes from constrained to unconstrained the difficulty level of recognizing also increases. As in constrained writing each character is written separately with spaces and no character touches the other character but in other case character in one word is connected and strokes are used more than once in individual character, it is referred to as cursive writing. As there is no clear boundaries are specified between characters to distinguish them hence recognition is difficult. Also, it is observed that many users write in mixed cursive writing illustrated in Figure 1.5 (Agrawal, 2011).



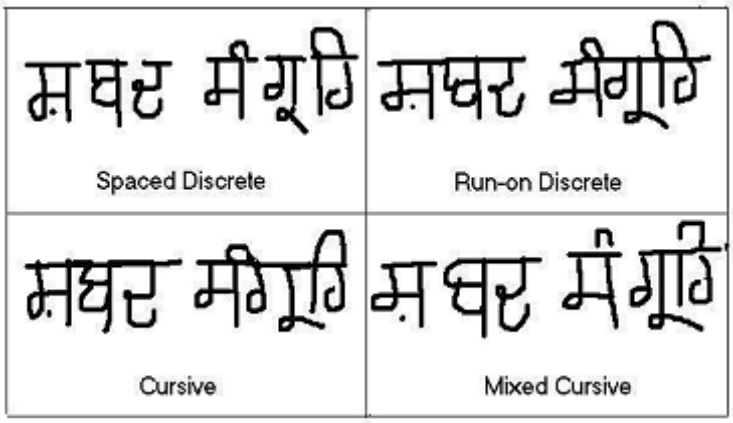


Figure 1.5: Different styles of writing

1.3.3 Personal and Situational Aspects

Personal factors include writer's writing style which might be affected by handedness, either left handed or right handed and many writers are habitual to write random or specific inclined text lines, Some users write very fast some write very slow. A good recognition machine requires a neat and clean handwriting and this writing style also depends on profession of writers to some extent as shown in Figure 1.6 (Tappert, 1984).

The situational aspects depend on the facts either writer is interested or not to write, how much attention or concentration writer is paying, whether there was any interruption while writing. The material used in writing may provide comfort or distortion to writer that result in variations in handwriting. This also includes the position and size of the writing pad.

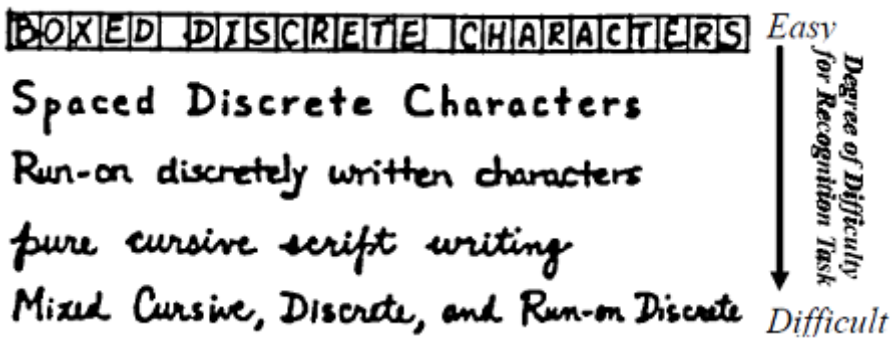


Figure 1.6: Types of writing styles

1.3.4 Writer dependent vs. writer independent recognition system

The writer dependent recognition system is used to recognize the samples of only those writers whose samples are taken to train the recognition system. It is specific to

a group of writers or a single writer, if it is train with the sample taken from a single writer. In writer dependent system, all possible style variations can be trained to the system, hence a higher recognition rate can be achieved.

On the other hand writer independent system is a generalized recognition system which works for the entire writer means, it recognizes the handwriting samples of unknown writers. Hence, it needs to train the system with all possible and commonly used style variations. Therefore it needs to train the system with large number of samples from a large number of different writers, to make the recognition system generalized, therefore the writer independent system has a comparatively lower recognition rate. In practice, the writer independent system is in more demand because of generalized application.

1.4 Online Handwritten Character Recognition System

It involves many steps to completely recognize and produce machine encoded text. These phases are termed as Data acquisition, Pre-processing, Feature extraction, recognition and post-processing. The architecture of these phases is shown in Figure 1.7 and these phases are listed below with a brief description. These phases are elaborated in the next sections.

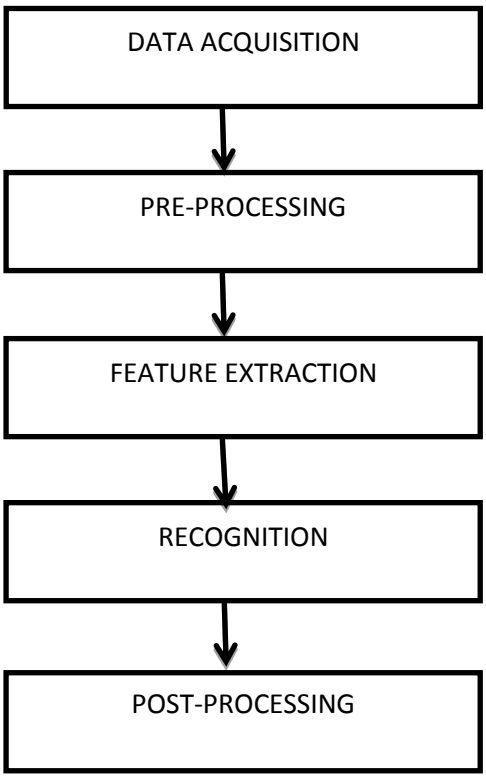


Figure 1.7: Phases of online handwritten character recognition

1.4.1 Data Acquisition

Online handwriting recognition requires a device that captures the writing as it is written by the writer. The most common devices are electronic tablet or digitizer. These devices use a pen that is digital in nature. Data collection is the first phase in online handwriting recognition that collects the sequence of coordinate points a moving pen. A typical pen includes two actions, namely, PenDown and PenUp. The pen traces, between PenDown and PenUp is called as stroke. These pen traces are sampled at a constant rate, therefore these pen traces are evenly distributed in time not in space. The appearances of Personal Digital and Assistants, cross pad and tablet are shown in Figure 1.8 used for data acquisitiom.

Figure 1.8: Commonly used hardware devices for data acquisition

1.4.2 Pre-processing

This phase in handwriting recognition system is applied to remove noise or distortion present in input stroke due to hardware and software limitations like: irregular size of the stroke, missing points of coordinates while capturing pen movement, jitter present in the text. Left or right bend in handwriting and uneven distances of points from neighbouring positions with many more problems like,

- Accidental pen lifts which occur when the writing pressure falls below the tablet activation threshold
- Repetitive or redundant samples when the pen is stationary or moving slowly
- Isolated samples which are distant from the general outline due to the irregular motion of the writer's hand or imperfections in the tablet

To remove these types of error pre-processing is required in online handwriting recognition system, it includes five common steps, namely, removal of duplicate points, size normalization, centering, interpolation missing points, and resampling of points. These steps are shown in Fig 1.9.

1.4.3 Feature Extraction

Feature extraction is extracting information from raw data which is most relevant for classification purpose and that minimizes the variation in class and maximizes the variation between classes.

Selection of a feature extraction method is probably the single most important factor in achieving high recognition performance in recognition system. The computational complexity of a classification problem can also be reduced if suitable features are selected.

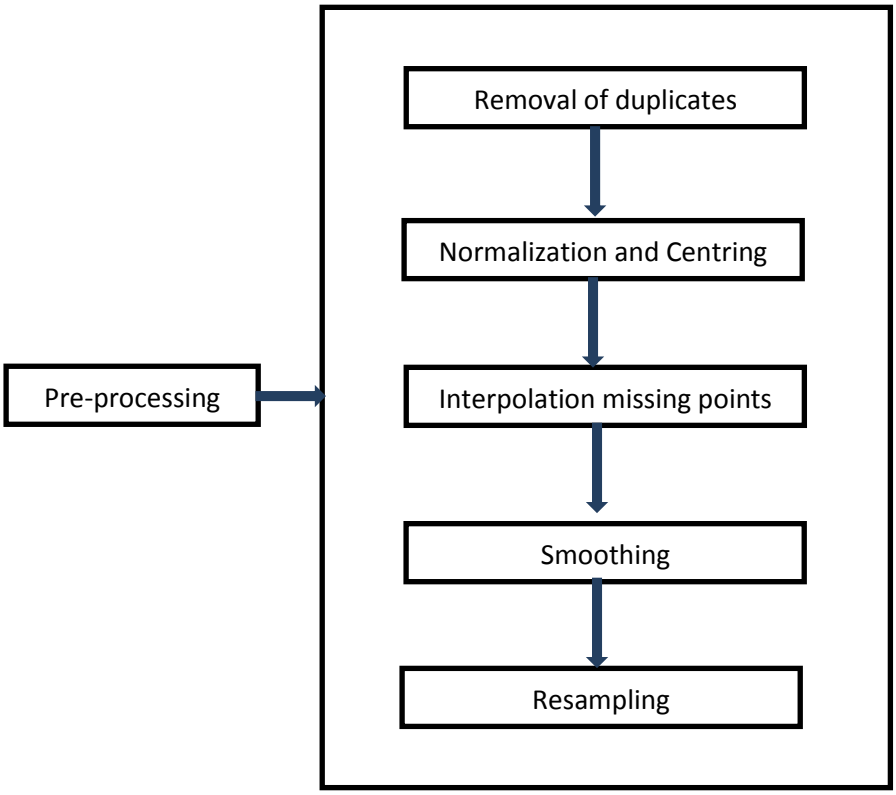


Figure 1.9: Pre-processing phase of recognition system.

1.4.4 Recognition

This is the most important phase of the online handwriting recognition system and uses the features, extracted in the previous stage, to identify the input stroke. In order to obtain the class labels of character or stroke, variety of classification methods can be used like statistical methods, structural and syntactical method, elastic matching method, neural network methods *etc*., a brief introduction of these methods are described below.

- **Statistical Classification Methods**

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

- **Artificial Neural Networks**

  Artificial neural networks are similar to neuron system. A neural network is composed of a number of interconnected neurons, and the manner of interconnection differentiates the network models into feed forward networks, recurrent networks, self-organizing networks, and so on. Single layer neural network and multilayer perception are the example of ANN classifier.

- **Support Vector Machines (SVM)**

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

- **Structural Classification Methods**

Structural methods recognize patterns via elastic matching of strings, graphs, or other structural descriptions. Structural pattern recognition methods are used more often in online character recognition than in offline character recognition. Instead of representing the character pattern as a feature vector of fixed dimensionally, structural methods represent a pattern as a structure (string, tree or graph) of flexible size. The structural representation records the stroke sequence or topological shape of the character pattern, and hence resembles well to the mechanism of human perception. In recognition, each class is represented as one or more structural template, the structure of the input pattern is matched with the templates and is classified in the class of the template of minimum distance or maximum similarity.

- **Multiple Classifier Methods**

By multiple classifier methods, the classifications of multiple classifiers are combined to reorder the class. Different classifier show variable performance, varying classification accuracy and speed, and different errors on concrete patterns. Hence, it is natural to combine the strengths of different classifier to achieve higher performance. This is usually accomplished by combining the decision of multiple classifiers. In research, many combination methods have been proposed, and the applications to practical problems have proven the advantage of the ensemble over individual classifier. The combination of different classifier can be categorized as parallel (horizontal) and sequential (vertical, cascaded). A parallel combination is most often adopted for improving the classification accuracy, whereas sequential combination is mainly used for acceleration the classification of a large category set.

1.5 Literature survey

The basic idea of doing this survey is to find that how many different types of techniques have been applied for the recognition and to see the recognition accuracy in percentage of their respective methods. The survey is done on both online and offline handwriting recognition systems with a special emphasis on the Indian scripts.

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

1.5.1 Work Related to Pre-processing

A survey carried out by Tappert *et al*. (1988) and by Tappert *et al*. (1990) identifies different pre-processing techniques like External and internal segmentation; and noise reduction including smoothing, filtering, wild point correction, dehooking, dot reduction and stroke connection; normalization. The normalization technique consists of deskewing, base line drift correction, size normalization and stroke length normalization.

Pre-processing technique for online recognition given by (Nair and Leedham, 1991). They present solutions to some of the major problems that occur in recognition system. They describe seven steps for pre-processing a stroke before they can move on to recognition phase. These steps are:

- Minimum distance filtering.
- Modified gear backlash smoothing.
- Minimum distance filtering.
- Straight line average smoothing.
- Minimum distance filtering.
- Straight line average smoothing with angle constraint.
- Serif Removal.

Homayoon *et al*. (1994) have mainly discussed two pre-processing steps for an online handwriting recognition system, which are size normalization and slope correction. According to them while performing size normalization, the base-line and the mid-line need to be estimated. The area surrounded by the base-line and the mid-line is the only part of any word which is always non-empty. Once accurate estimates of the base-line and the mid-line are obtained, a magnification factor can be computed from

the ratio of the nominal mid-portion size and that of the input. The entire input data may then be scaled using the obtained magnification factor. Then slope correction is performed which is based on the evaluation of the mean velocities in the x and y directions. The angle between these velocity vectors is used to estimate the angle by which the words should be rotated. For special cases such as in words with all block capital letters, special provisions was taken to avoid wrong estimation of the rotation angle.

Aparna *et al*. (2004) proposed a pre-processing technique that consists of interpolation, smoothing and normalization of strokes. For normalizing a stroke they first determine the bounding box of the entire stroke. Then they divide both and dimensions of the stroke by the 'height' of the horizontal block. This preserves the relative size of strokes in a horizontal block. The strokes are then converted onto curve length base (sampled uniformly along curve length) and then smoothed independently along t-axis using a Gaussian filter.

Nair *et al*. (2008) discussed the problem arises on writing like accident pen lifts which occur when the writing pressure falls below the tablet activation threshold, redundant samples when the pen is stationary, isolated samples due to irregular motion, serifs at the start and end of pen strokes. They have also given the solution for these problems in their work.

Zheng and Zhu (2006) demonstrated that hardware system collects handwriting signature signals, including the position signals of and in real time. The space resolution is 4900x4900 and the sampling interval is 10ms. First, they removed the gaps between strokes, and these gaps reflect the time that the handwriting pen does not contact the handwriting pad and the positions that strokes begin and end. Simultaneously, they normalized the size, the length of the signature

Hosny *et al*. (2011) has given pre-processing technique for Arabic characters using four steps:

- **Duplicate Points Removal**: By checking whether the coordinates of any two points in a stroke are the same. If so, one of them is removed.
- **Interpolating Points**: To add any missing points by linear interpolation.

- **Smoothing**: To eliminate hardware imperfections and trembles in writing each point is substituted with the weighted average of its neighbouring points.
- **Re-sampling**: Due to the variation in writing speed, the acquired points are not distributed evenly along the stroke trajectory. This operation is used to get a sequence of points which is equidistant.

1.5.2 Work Related to Character Recognition

For Indian languages most of the research work is performed on firstly on Devnagri and secondly on Bangla script. Pal and Chaudhary (2004) presented a survey on Indian Scripts Character Recognition. This paper introduces the properties of Indian scripts and work and methodologies approached to different Indian scripts. They have presented the study of the work for character recognition on many Indian languages scripts including Devnagari, Bangla, Tamil, Gurmukhi, Gujarati and Kannada.

Wakabayashi *et al*. (2009) also presented a comparative study of Devnagari and handwritten character recognition using different features and classifier. They used four sets of features based on curvature and gradient information obtained from binary as well as grayscale images and compared result using 12 different classifiers as concluded the best results 94.94% and 95.19% for features extracted from binary and gray image respectively obtained with Mirror Image Learning (MIL) classifier. They also concluded curvature features to use for better results than gradient features for most of classifiers.

A later review of research on Devnagari character recognition is also presented by Dungre *et al*, (2010). They have reviewed the technique the techniques available for character recognition. They have introduced image pre-processing techniques for thresholding, skew detection and correction, size normalization and thinning which are used in character recognition. They have also reviewed the feature extraction using global transformation and series expansion like Fourier transform, Gabor transform, Wavelet moments, Statistical features like zoning, projections, crossing and distance and some geometrical and topological features commonly practiced. They also reviewed the classification using template matching, statically techniques, neural network, SVM and the combination of a classifier for better accuracy is practiced recognition.

Mukherji and Rege, (2009) used shape features and fuzzy logic to recognize offline Devnagari Character recognition. They segmented the thinned character into strokes using structural features like endpoint, cross-point, junction points, and thinning. They classified the segmented shapes or strokes as left curve, right curve, horizontal stroke, vertical stroke, slanted lines *etc*., They used tree and fuzzy classifier and obtained average 86.4% accuracy.

Vamvakas *et al*, (2007) described the statistical and structural features they have used in their approach of Greek handwritten character recognition. The statistical features they have used are zoning, projections and profiling, and crossing and distances features. In direction features they used directional histograms of contour and skeleton images. In addition to normal profile features they described in and out the profiles of contour of images. The structural features they have depicted are end point, crossing point, loop, horizontal and vertical projections histogram, radial histogram, radial out-in and in-out histogram.

Sarabjit *et al*. (2007) described projections based statistical approach for handwritten character recognition. They proposed four sided projections of characters and projections were smoothed by polygon approximation.

Araki *et al*. (2008) proposed a statistical approach for character recognition using Bayesian Filter. They reported good recognition performance in spite of the simplicity of Bayesian algorithm.

Arora *et al*. (2008) used intersection shadow features. Chain code histograms and straight line fitting features and weighted majority voting techniques for combining the classification decision obtained from different Multilayer Perception (MLP) based classifier. They also used chain code histogram ad moment based features to recognize handwritten Devnagari characters. Chain code was generated by detecting the direction of the next in-line pixel in the scaled contour image. Moment features were extracted from scaled and thinned character image.

1.5.3 Work on Gurmukhi Character Recognition

In particular to Gurmukhi script, earliest major contributors founded are Chandan Singh and G. S. Lehal. They proposed Gurmukhi script recognition system (Lehal and Singh, 2000). Later they develop a complete machine printed Gurmukhi OCR system

(Lehal and Singh, 2006). Some of their other research works related to Gurmukhi Script recognition is done (Lehal and Singh, 2002) in which they proposed feature extraction, classification and post-processing approaches for Gurmukhi scripts.

Sukhpreet Singh *et al*. (2012) In this manuscript handwritten Gurmukhi character recognition for isolated characters is proposed. We have used Gabor Filter based method for feature extraction. Our database consists of 200 samples of each of basic 35 characters of the Gurmukhi script collected from different writers. These samples are pre-processed and normalized to 32*32 sizes. The highest accuracy obtained is 94.29% as 5-fold cross validation of whole database with SVM classifier used with RBF kernel.

Sharma *et al*. (2008) and Sharma *et al*. (2009) have presented the implementation of three approaches: elastic matching technique, small line segments and HMM based technique, to recognize online handwritten Gurmukhi character and reported 90.8%, 94.59% and 91.59% recognition accuracies respectively. In elastic machine technique, first they recognized the strokes and then evaluated the character based on the strokes.

CHAPTER 2

DATA COLLECTION, PRE-PROCESSING, AND FEATURE EXTRACTION

The phases of the recognition process have been discussed in Chapter 1. In this chapter we will elaborate each phase and discuss about the algorithm used in this thesis work.

2.1 Data Capturing

The characters are inputted with the help of mouse or digital pen (stylus) for writing on a writing pad. The time sequential signal from the movement of the pen on the writing pad captures the dynamic position of the pen with the help of the sensors attached to the pad, means it captures the sequence of consecutive points on the x-y plane in the form of coordinates.

The example of a sample stroke is given in Figure 2.1. A stroke S = {(X_0, Y_0), (X_1, Y_1), (X_2, Y_2), …, (X_n, Y_n)} is expressed by the time sequence of co-ordinates as shown in Figure 2.1 represents the generation of data for a sample of a Gurmukhi character in which (X_0, Y_0) and (X_n, Y_n) denote the initial and the final point of the generated time sequence of co-ordinates for the particular sample.

Figure 2.1: A sample stroke of Gurmukhi character

A stroke is defined as the sequence of sample points occurring between consecutive PenDown and PenUp transitions. A character could possibly be a uni-stroke character or formed from two or more strokes. In Gurmukhi Script also single stroke as well as multi-stroke both types of characters are present. In Figure 2.2 we have Gurmukhi Character 'ਗ'. We can observe that 'ਗ' is a 3-stroke character.

Speed of writing directly affects the recognition process. When the speed of writing is slow, the sample points are located densely, whereas quick writing produces sparsely located points. One can note that the speed of writing typically slows down on sharp corners, in the beginning of the stroke and at the end of the stroke. The selection of a suitable level of sampling rate and resolution depends on the writing speed and the scale of the meaningful pen trace features. If the sampling rate is too slow, odd corners will be introduced to the sampled pen trace and dome of the real corners and miniscule trace features can be missed.

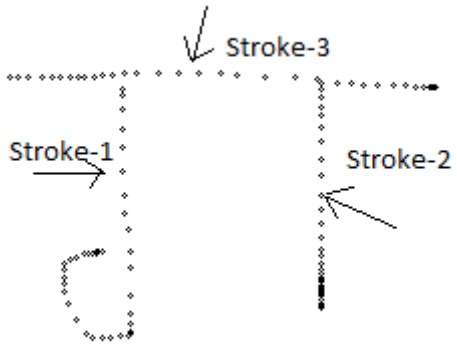


Figure 2.2: A sample Gurmukhi character of 3 strokes.

2.2 Pre-processing Phase

The input stroke contains noise and distortion due to hardware and software limitations. As discussed in chapter 1, the issues in online handwriting in stroke has to be removed before so as to extract efficient feature which can be proven as the higher accuracy of recognition. Pre-processing phase for Gurmukhi character recognition consists of following stages are:

2.2.1 Removal of Duplicate Points

Sometimes while writing, the stroke is overlapped due to the shape of the stroke or the character, and hence while writing the pixels captured contains duplicate points, as shown in Figure 2.3, which has to be removed for getting equal spaced resampling points.

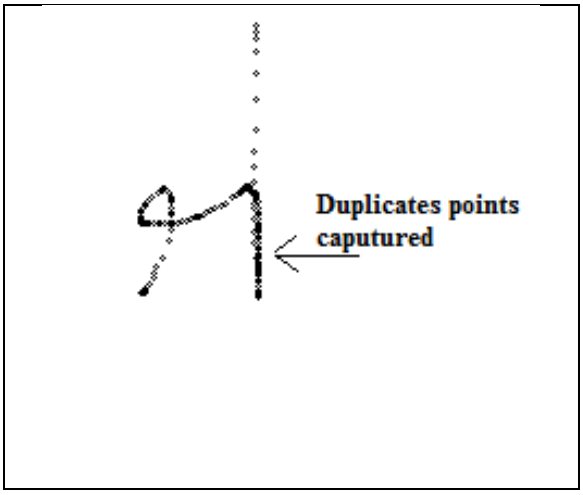


Figure 2.3: A sample stroke showing duplicate or overlapped points

Algorithm 2.1 for removing duplicate points:

1. Store the captured *X* and *Y* co-ordinate value captured in different array.
2. Set *i* = total number of pixels captured
3. Set *j* = 0 and *k* = 0.
4. Repeat step 4 until *j* < *i* and increment *j* by 1.
5. Set *k* = *j* + 1 and repeat step 6-8 until *k* < *i*-1 and increment *k* by 1.
6. If (*X*[*j*] == *X*[*k*] and *Y*[*j*] == *Y*[*k*]).
7. Delete *X*[*k*] and *Y*[*k*] from the array.
8. End if.

2.2.2 Size Normalization and Centring

The size of the input stroke depends on the movement of a pen, the writing pad and also on the writer. Strokes are not of the same size, when we move the pen again and again. Size normalization normalizes every stroke to the same size and centered to a constant frame and places the text at a fixed distance from the origin. This can be achieved by comparing the input character border frame or taking the

maximum value of the co-ordinates of x-axis and y-axis with an already assumed fixed frame and the input character can be moved along with an assumed center location for centring.

The algorithm implemented in this work fixes the size of an input character to a fixed size of 300x300 pixels and places in the center of the fixed frame. Thus, if a stroke is entered, less than 300x300 size. It is transformed into a fixed size of 300x300 pixels. Similarly if the stroke entered, is greater than the size of 300x300 then also it will be transformed to a size of fixed size 300x300. This can be seen in the Figure 2.4 where in the first case, the smaller size stroke is entered and in the second window after transformation normalized and centered stroke is shown.

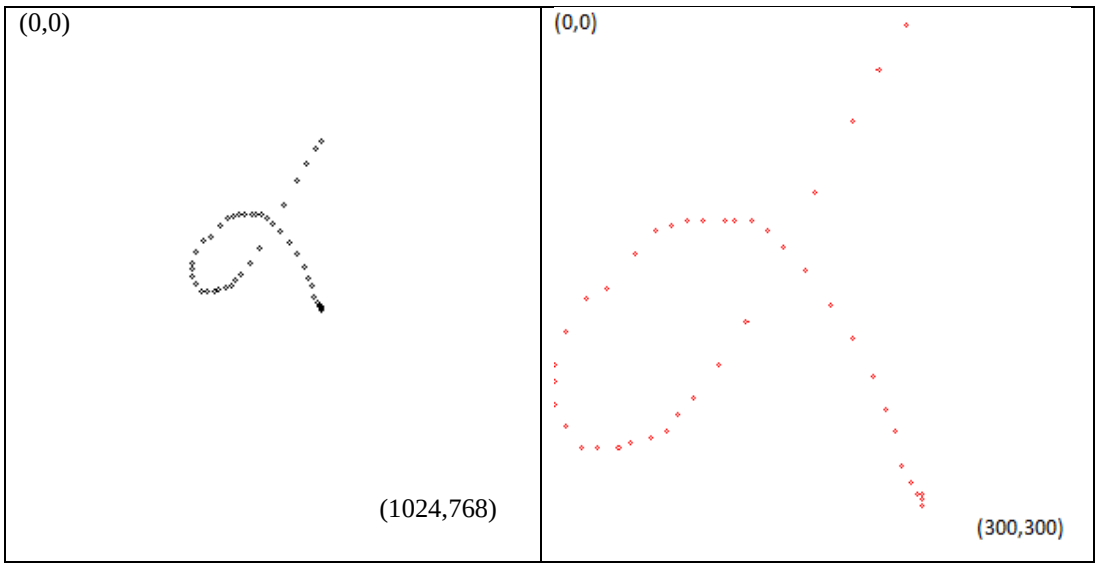


Figure 2.4 Handwriting character after normalization of a sample Gurmukhi character (ka)

Algorithm 2.2 Normalization and centring

1. Set *i* = total no of pixel or points captured.
2. Set *frame-x* = 300 and *frame-y* = 300 (for scaling to a 300x300 area).
3. Find the maximum of *X* and *Y* co-ordinates values, as *x-max* and *y-max* and minimum of co-ordinates value, as *x-min* and *y-min*
4. Calculate *Sx* = *frame-x*/(*xmax* - *xmin*)
5. Calculate *Sy* = *frame-y*/(*ymax* - *ymin*)
6. Take min(*Sx*, *Sy*) as scaling factor, fact.
7. Repeat step 8 to 10 for all the points captured.
8. $P_x[i] = P_x[i] * fact$, where P_x is the X co-ordinate point.

9. $P_y[i] = P_y[i] * fact$, where P_y is the y co-ordinate point.
10. $i = i - 1$.

This algorithm normalizes the stroke in a fixed size of 300x300 pixels.

Centring

1. Set i = total no of pixel or points captured.
2. Find the min of co-ordinates value of x-axis and y-axis and set as *xmin* and *ymin* respectively.
3. Repeat step 4 to 6 untill $i = 0$,
4. $P_x[i] = P_x[i] - xmin$, where P_x is the x co-ordinate point.
5. $P_y[i] = P_y[i] - ymin$, where P_y is the y co-ordinate point.
6. $i = i - 1$.

2.2.3 Missing Point Interpolation

When a character is drawn with high speed, the writing pad sensors may miss some point to capture as hardware issue or it can be a software problem the interface algorithm cannot capture a high speed writing input stroke. These missing points can be interpolated using various techniques such as Bezier and B-Spline curve interpolation. In this thesis work, cubic Bezier interpolation has been used. In this piecewise interpolation technique, a set of consecutive four points is considered for obtaining the curve. The next four points give the next Bezier curve and so on.

Bezier interpolation is applied between the points where the distance is greater than one. First, the algorithm finds the difference between the pair of consecutive points and calls Bezier where the distance between the points is greater than one. Interpolation is done between those points whose distance is greater than only by making a Bezier curve. Figure 2.5 shows how interpolation of missing points is done. Figure consists of three columns where first column shows the inputted stroke second column shows the normalized and third column shows the stroke after interpolation.

Algorithm 2.3 Interpolating Missing Points.

1. Create an empty list L for storing the points generated.
2. Set t = number of strokes in the list and set $k = 1$.

3. Repeat step 4 for each stroke k, until $k <= t$.
4.
   i) Calculate I as the total number of points in the current stroke.
   ii) If ($I >= 4$) then GOTO step 5
5. Set $P = 1$
6. Repeat step 7 until $P = I-4$
7. Repeat step 8 until distance $(P_i, P_{i+1}) > 1$
8. Call Bezier $(P_i, P_{i+1}, P_{i+2}, P_{i+3})$
   i) Update List L by incorporating the new points as the consecutive points obtained through Bezier Function.
   ii) Set $P = P+1$
9. Exit

Function Bezier $(P_i, P_{i+1}, P_{i+2}, P_{i+3})$

1. u is a variable such that $0 \leq u \leq 1$.
2. Set $u = 0.1$.
3. Repeat steps 4 and 5 until $u \leq 1$.
4. Calculate x co-ordinate of the new point as
   $(1-u)^3 P_{ix} + 3u(1-u)^2 P_{(i+1)x} + 3(1-u)u^2 P_{(i+2)x} + u^3 P_{(i+3)x,}$
   And similarly calculate Y co-ordinate
   $(1-u)^3 P_{iy} + 3u(1-u)^2 P_{(i+1)y} + 3(1-u)u^2 P_{(i+2)y} + u^3 P_{(i+3)y,}$
5. $u = u + 0.1$.
6. Return.

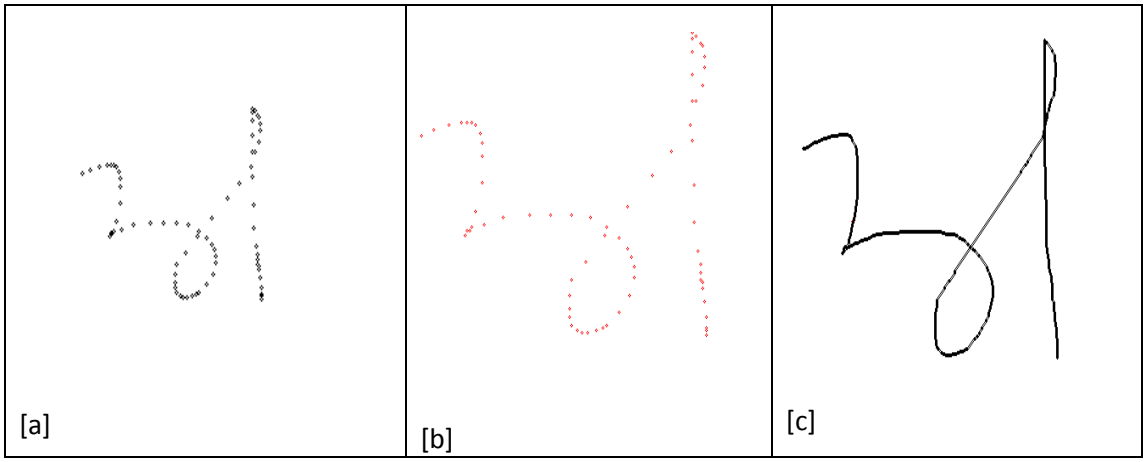


Figure 2.5: A sample stroke of Gurmukhi showing (a) -Inputted stroke (b) normalized stroke (c) Interpolated stroke

2.2.4 Resampling

Resampling of points is done to fix the number of points in the input handwritten character and also retain the original shape of the character. A filter is applied such that only a fixed number of points are selected. According to the literature survey, the results we obtained said that the best results can be obtained when the numbers of resampled points are 64. The points are selected in such a way that the shape of the character will not get affected. Since, the distance between the two points after interpolation is less than one and a half, removal of points shall include two options: remove all points between the pair of points having distance less than one, or, remove points at constant distances, *i.e.*, two or three and so on. Figure 3.4 illustrates the resampling done on the input handwritten character.

Algorithm 2.4 Resampling

1. For all the points in the list,
   If (distance between the two points) >1.5 then
   Call algorithm (interpolation missing points)
   End if
2. For all the points calculate P as a total no of points in a stroke.
3. For getting the 64 points calculate X=P/64
4. If X is absolute then
   For all the points of a stroke in a list, remove points at X distance with respect to total number of points in the stroke

End if

5. Else take X= (int) X and $Y = X+1$

   Two linear equations will be made

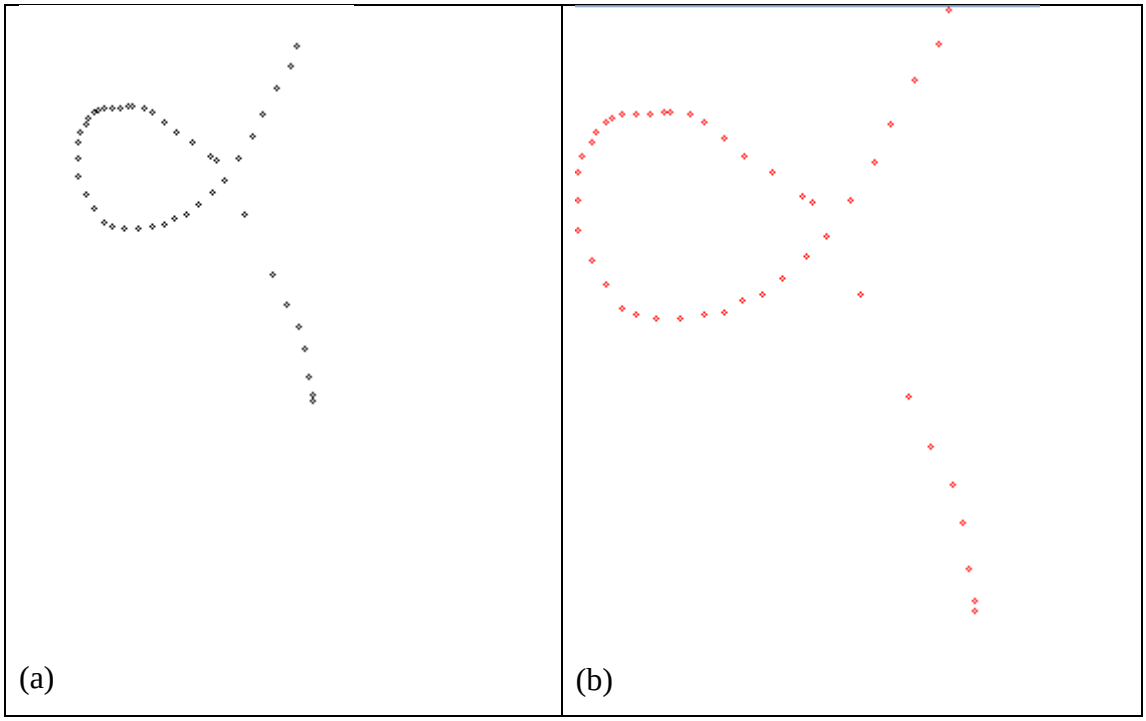
   $X*a + Y*b = P$ (Total no of points in a stroke)

   $a + b = 64$

   Find the value of a and b

   For all the points of a stroke, remove the points at X distance 'a' number of times and Y distance 'b' number of times.

End else.



(a) (b)

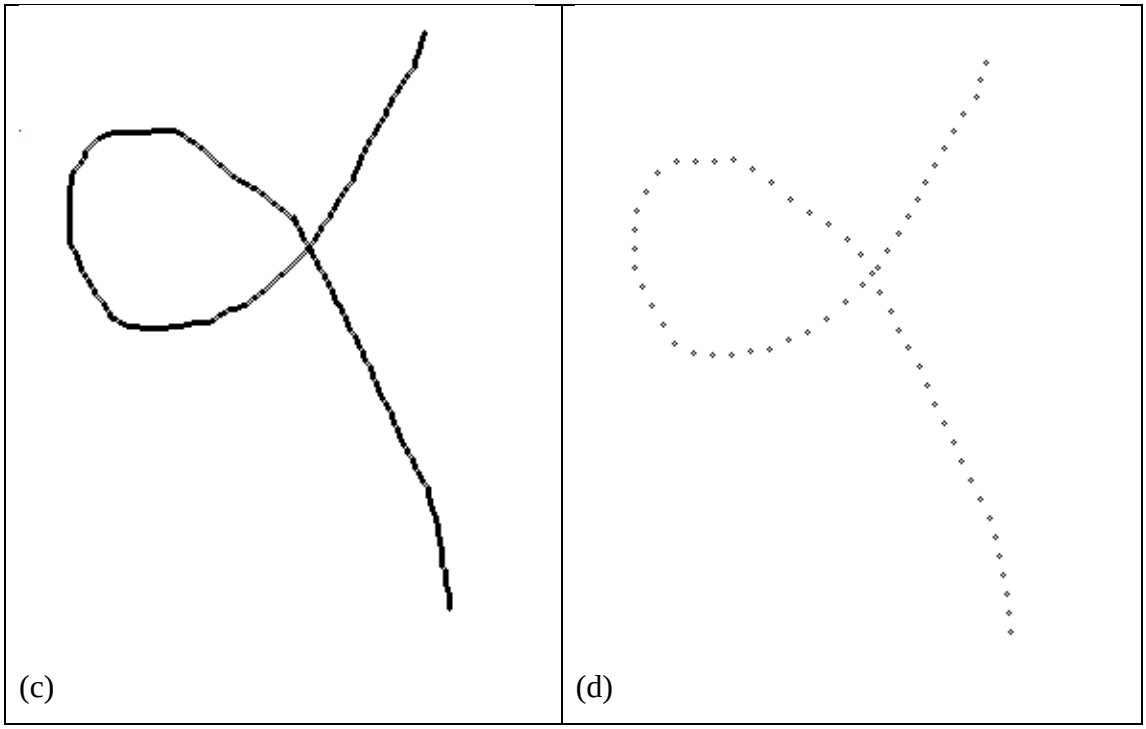


Figure 2.6: Shows the entire algorithm (a) input stroke by the writer (b) normalized and centering (c) interpolation missing points (d) resampling of points.

2.3 Feature Extraction

Feature extraction is a very important step and recognition accuracy largely depends upon the features extracted. Points generated after pre-processing phase is used as a feature for recognition. The number of points for each stroke will be fixed for each stroke and are equidistant as far as possible. In this thesis work the number of points taken is 64. The extracted feature, *i.e.*, the x and y co-ordinates of each stroke are storing it in a format of a used classifier for recognition as it will be passed to a classifier which tells about the class of these points they belongs to in recognition phase.

CHAPTER 3

GURMUKHI CHARCATER RECOGNITION AND POST-PROCESSING

In this chapter the recognition phase of the online handwritten recognition system is discussed. Before recognition of the stroke, it is first, pre-processed as discussed in chapter 2. The pre-processed handwritten stroke points are then recognized using Support Vector Machine (SVM) which returns stroke-id. Introduction to SVM is discussed in section 3.1, the Recognition phase is discussed in section 3.2 and post-processing part is discussed in section 3.3.

3.1 Introduction to SVM

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

Support Vector Machines are based on the concept of decision planes that defines decision boundaries. A decision plane is one that separates between a set of objects having different class membership. The Figure 3.1 shows the classification of objects having class one of the two: either triangle or diamond. The separating line defines a boundary on the right side of which all objects are diamond and to the left of which all objects are triangulated. Any new object falling to the right is labelled, *i.e.*, classified, as diamond (or classified as triangle if its fall to the left of the separating line.

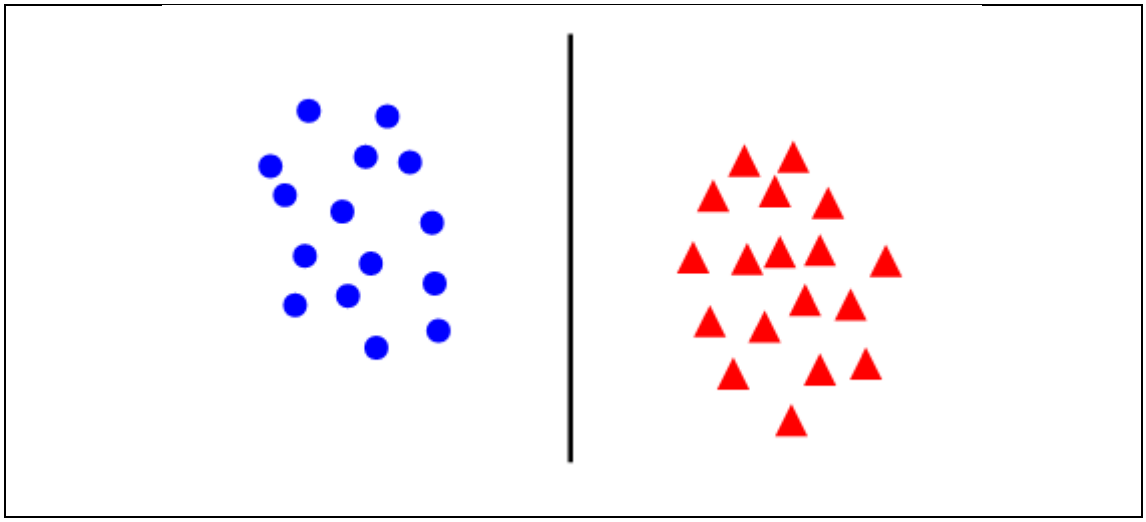


Figure 3.1: Linear classification of objects by SVM into two classes

The above Figure is a classic example of a linear classifier, *i.e.*, a classifier that separates a set of objects into their respective classes (diamond and triangle in this case) with a line. Most classification tasks, however, are not that simple, and often more complex structures are needed in order to make an optimal separation, *i.e.*, correctly classify new objects (test case) on the basis of the examples that are available (train case). This complex separation is represented by a complex function rather than linear function and such classification of objects is shown in Figure 3.2 (a). Compared to the previous schematic, it is clear that a full separation of the diamond and triangle objects would require a curve (which is more complex than a line).

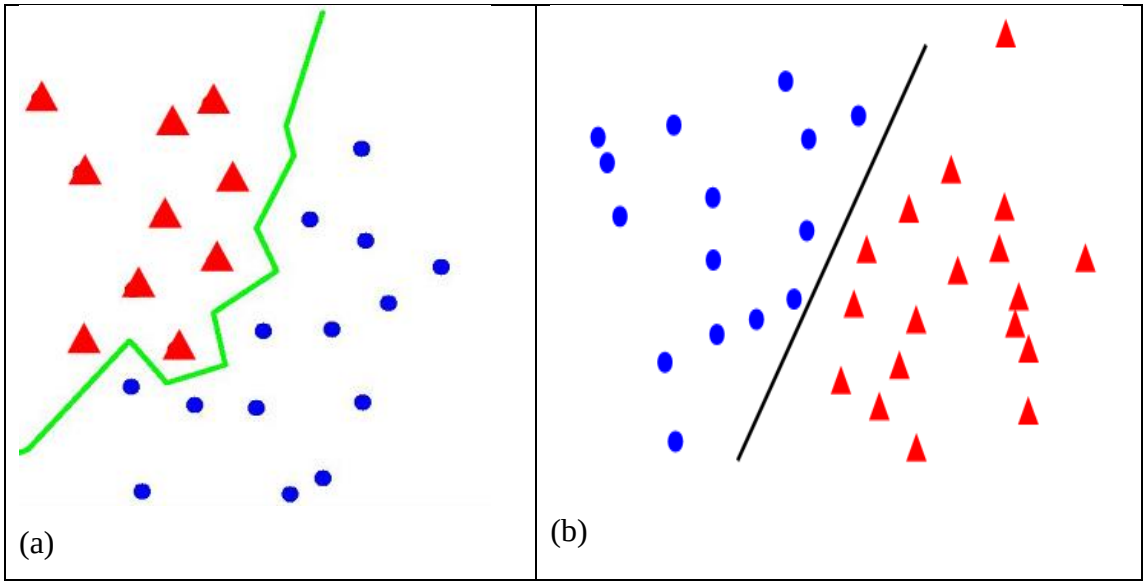


Figure 3.2 Mapping of SVM classifications from complex to linear: (a) classification by a complex kernel function (b) mapping of input space of Figure (a) into feature space

Classification tasks based on drawing separating to distinguish between objects of different class membership are known as hyperplane classifier. It linearly separates any two classes by finding a maximum margin between the two classes. The margin means the minimal distance from the separating hyperplane to the closest data points. SVM learning machines search for an optimal separating hyperplane where the margin is maximal. The outcome of the SVM is based only on the data points that are at the margin and are called 'Support Vectors'. This concept is illustrated in Figure 3.3.

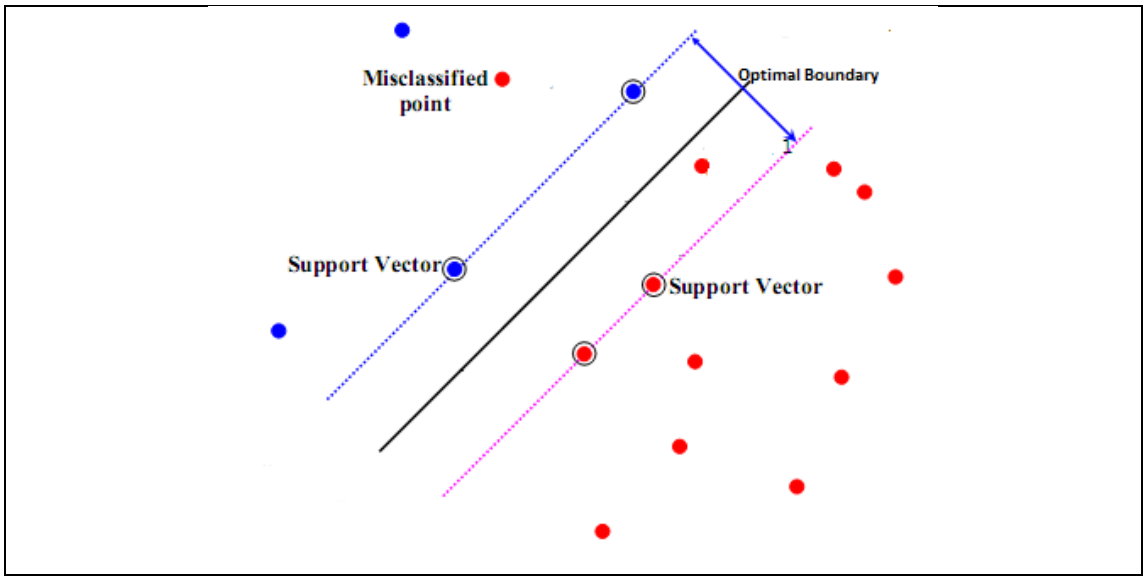


Figure 3.3: Concept of support vectors

The black line in Figure 3.3 determines the mapping done from the input space to the feature space in SVM.

Support Vector Machine classifier has gained immense popularity in recent years providing excellent recognition results in various applications. Some of the applications of SVM are:

- Text (and hypertext) categorization.
- Image classification
- Bioinformatics (protein classification, cancer classification)
- Topic drift in page ranking algorithms

- Handwriting character recognition

SVM in the present dissertation has been used for handwriting character recognition.

3.2 Recognition of Gurmukhi Characters

As discussed earlier, a Gurmukhi character can be written in one stroke or more than one stroke, it depends on the user. A character of Gurmukhi is divided into three zones, *i.e.*, lower zone, middle zone and upper zone, for recognition, stroke id is given to every possible stroke that can make a Gurmukhi character. Ids given to a particular stroke is shown in Table 3.1 for lower zone, Table 3.2 for upper zone and Table 3.3 for middle zone.

With the help of electronic devices and tool developed in C#. Net, samples, which are given in the below tables of each zone of all the strokes are captured. In total 11 users, at an average of 10 has written each stroke of every class. Stroke ids given in the tables has 77 strokes in the middle zone, 12 strokes in upper zone and 7 strokes in the lower zone. 100 samples of each stroke for all the zones is captured and the co-ordinates of every sample stroke, *i.e.*, X, Y co-ordinates are stored in a text file, called as raw data, in a specified format which is discussed in the next section.

Table 3.1: Lower zone stroke ids

S.NO	STROKE ID	SYMBOL	DESCRIPTION
1)	101		Onkar
2)	102		Bindi
3)	103		Ra
4)	104		Ha
5)	105		Va
6)	106		Dulenkar
7)	107		Halant

Table 3.2: Upper zone stroke ids

S.NO	STROKE ID	SYMBOL	DESCRIPTION
1)	121		Upper bar
2)	122		Lawan
3)	123		Dalawan
4)	124		Hora
5)	125		Ghanoura bar
6)	126		Adhak
7)	127		Bindi
8)	128		Tippi
9)	129		Associate bar
10)	130		Dalawan
11)	131		Dalawan
12)	132		Lawan

Table 3.3: Middle zone stroke ids

S.NO	STROKE ID	SYMBOL	DESCRIPTION
1)	141		Urah
2)	142		Urah
3)	143		Urah

4)	144		Aarah
5)	145		Aarah
6)	146		Eedi
7)	147		Eedi
8)	148		Eedi
9)	149		Eedi
10)	150		Eedi
11)	151		Sassa
12)	152		Sassa
13)	153		Sassa
14)	154		Sassa
15)	155		Haaha

16)	156		Haaha
17)	157		Kakka
18)	158		Kakka
19)	159		Khakha, Pappa
20)	160		Khakha,
21)	161		Khakha, Pappa
22)	162		Vertical Bar
23)	163		Khakha, Dhadha
24)	164		Gagga, Rara
25)	165		Gagga, Rara
26)	166		Ghagga

27)	167		Ghagga
28)	168		Aeyaa
29)	169		Aeyaa
30)	170		Chachaa
31)	171		Chachaa
32)	172		Chhachha
33)	173		Chhachha
34)	174		Jajja
35)	175		Jajja
36)	176		Jhajja
37)	177		Jhajja
38)	178		Jhajja

39)	179		Neyaa, Vawa
40)	180		Neyaa, Vawa
41)	181		Vawa
42)	182		Tenqa
43)	183		Thathaa
44)	184		Thathaa
45)	185		Dadda
46)	186		Dadda
47)	187		Dhadhaa
48)	188		Dhadhaa
49)	189		Nahnaa

50)	190		Nahnaa
51)	191		Tatta
52)	192		Tatta
53)	193		Dadhaa
54)	194		Dadhaa
55)	195		Dadhaa
56)	196		Nanhaa
57)	197		Nanhaa
58)	198		Nanhaa
59)	199		Half Vertical bar
60)	200		Faffa
61)	201		Faffa

62)	202		Babba
63)	203		Babba
64)	204		Bhabhaa
65)	205		Bhabhaa
66)	206		Mamma
67)	207		Yaeya
68)	208		Yaeya
69)	209		Yaeya
70)	210		Lalla
71)	211		Lalla
72)	212		Rara

73)	213		Rara
74)	214		Rara
75)	215		Cehari
76)	216		Behari
77)	217		Mamma
78)	218		Jhajja
79)	219		Rara
80)	220		Cehari
81)	222		Jajja
82)	223		Jajja

For recognition of a Gurmukhi character, which is in the form of strokes, first pre-processed and 64 resemble points are recorded, *i.e.*, 64 *X* co-ordinates and 64 *Y* co-ordinates. These points are stored in a specified format of the classifier. Here we have used SVM classifier, therefore, we have stored the resampled points in SVM format. The format of the SVM is given as below:

Stroke ID: 1:X co-ordinate 2:Y co-ordinate 3:X co-ordinate 4:Y co-ordinate,…, 127:X co-ordinate 128:Y co-ordinate.

Hence, all the 100 samples of each stroke are firstly pre-processed and the result is stored in the SVM format as shown in Figure 3.4. As discussed in earlier section the SVM needs to be trained with a set of training data, in the recognition, which gives model file as an output. Before training the training data is first scaled in a lower and upper bound. As discussed earlier, after pre-processing of raw data, co-ordinates values range from (0, 0) to (300, 300), these co-ordinates was scaled with a lower limit of 1 and upper limit of 9, using the tool of LIBSVM.

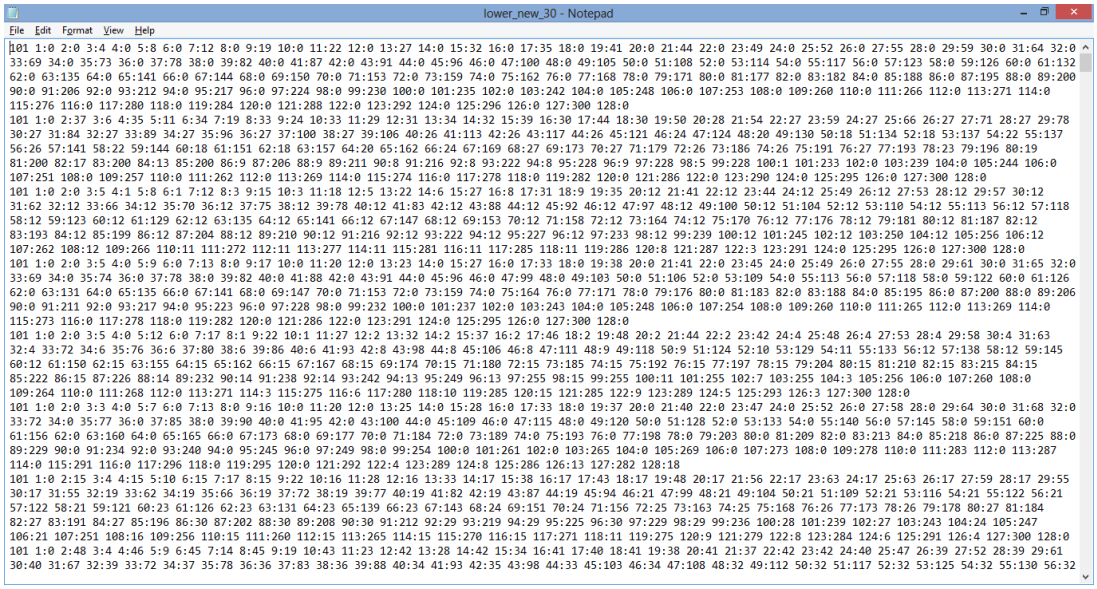


Figure 3.4: Pre-processed points in SVM format

The scaled file was used for training the SVM machine. The scaling can be done with any lower and upper limit depends on any data set values. In this thesis, to scale the Data, lower limit of 1 and upper limit of 9 is taken. The scaled file is shown in Figure 3.5. The training data for training the SVM should be accurate to get correct recognition results, thus respective stroke-id of stroke is given, captured at the time of data collection.

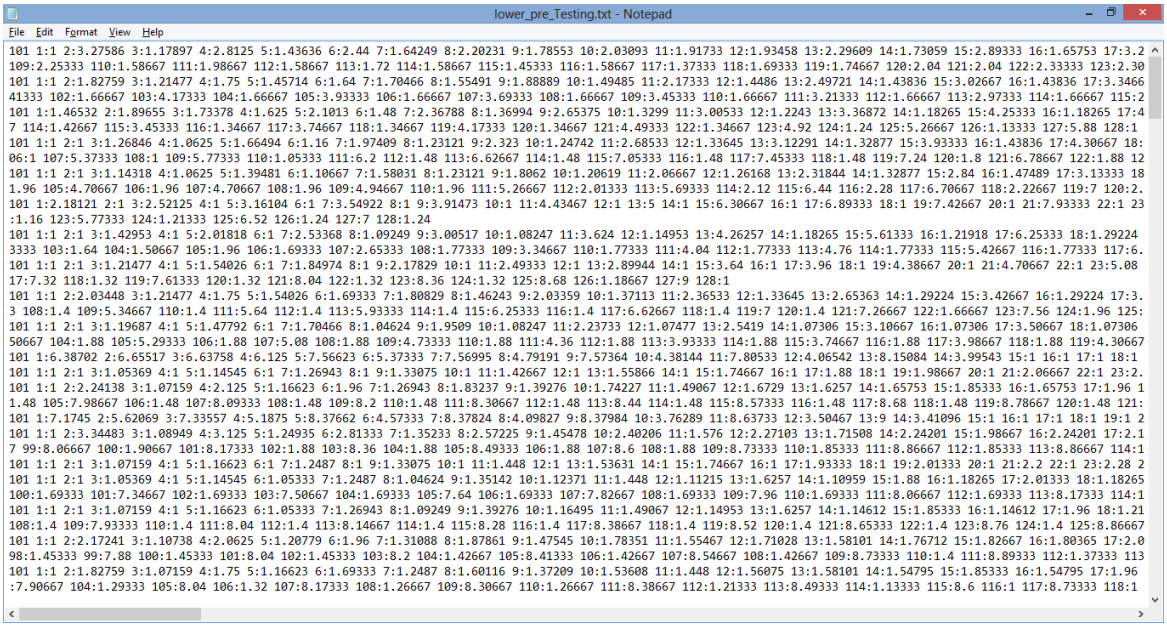


Figure 3.5: Scaled file of pre-processed points

Parameter Selection:

The effectiveness of SVM machine depends on the Parameters used, *i.e.*, the selection of SVM's kernel, kernel parameter and the soft margin parameter C. In this thesis the kernel model used is linear and the kernel parameter's default value of epsilon value is 0.01 and gamma value is 0.005.

These parameters are passed for training the SVM machine with the scaled file of training set to get a model file which is further used for recognition. There are 77 strokes in the middle zone, which are considered in this work, has 100 samples each, they are after pre-processing saved in a text file named as training. txt, means there are total 75*100 = 7500 strokes in a file and each are resampled to 64 points after processing. This file is then scaled to be in upper and lower limit and a new file as training.txt.scale is generated when training.txt file is provided as input to the tool LIBSVM. This scaled file, training.txt.scale, was used for training the SVM machine to generate a model file. The model file was generated, on passing the scaled file with the required parameters, *i.e.*, kernel selection, Gamma value and epsilon value. A sample model file shown in Figure 3.6.

Figure 3.6: Model file of pre-processed point

Now when a writer writes a stroke or character on writing pad or on an interface for recognition, the points will be pre-processed and then stored in SVM format. This, pre-processed file, was scaled first, to the same parameters as it was used for scaling the training file. Then this scaled file will be passed to the SVM machine with the model file and we get the recognized stroke id which is identified by the SVM as an output.

3.3 Post-Processing

Printing of respective character, which is recognized by SVM, on the user interface of the system is done in this phase. After getting stroke-id which is recognized by the Support Vector Machine as discussed earlier in section 3.2, it is then passed to post-processing phase. This phase contains a database of the rules for creating a character (kumar and sharma, 2013). A particular character can be created though different strokes, the database contains all those rules of strokes that can make a character individually or by combination of stroke. For example stroke id 142 individually forms Ura (ઉ) and combination of strokes 141+121 also forms Ura (ઉ) as shown in Table 3.4. A character formation rule can be one or more than one is depending on the classification of stroke-id made. As shown in Table 3.4 some character has 2 rules and some has 4 rules of its formation and it can be more than ten also.

Table 3.4 shows some example of the rules that can form a character.

Character	Rule 1	Rule 2	Rule 3	Rule 4
ੳ	142	141+121		
ਅ	144	144+121	145+162+121	
ੲ	150	182+147	150+121	146+147+121
ਸ	153	217+154	151+154	151+162+121
ਹ	156	155+121		
ਕ	158	157+121		
ਖ	160	159+163+129	161+162+163	159+163
ਗ	164+154	164+162+121	165+154	165+162+121
ਘ	166+121	167+162+121	166	167+162
ਙ	169	168+121	168	

With the help of these rules, a Unicode is return by the program of post-processing and respective character of Gurmukhi is printed using the Unicode standards.

CHAPTER 4

RESULTS AND DISCUSSIONS

Algorithms, discussed in Chapter-3, have been implemented on the raw data of strokes which was captured through an electronic device written by different writers. The co-ordinates of the strokes were captured and stored in a text file. On this data pre-processing algorithms were processed and saved in a text file in SVM format which is further passed to SVM for recognition of stroke-id. This data were divided in three datasets of 30, 50 and 70 for each zone, *i.e.*, lower zone, middle zone and upper zone. Dataset of 30 samples contains 30 samples of each stroke. Similar work was done for dataset of 50 and 70.

In this work, *k*-fold cross validation was used for testing. In *k*-fold cross validation we first divide the training set into k equal subsets, out of which 1 subset is tested by classifier SVM, and remaining subsets are used for training the SVM classifier. By cross validation, each subset of data is tested and the average percentage of correctly recognized stroke-id count is recorded.

In 3 fold testing the data set of 30 samples is divided into 3 equal datasets out of which 2 will be used for training the SVM machine and 1 will be used for testing and then accuracy for different parameters of gamma and epsilon (SVM parameters) value is recognized. Every divided dataset was tested and average accuracy was recorded. A similar process was followed for 4 and 5 fold testing.

Data was pre-processed using algorithms proposed in Agrawal, (2012) and proposed in this thesis. Testing of both result sets was performed using 3 schemes and their result is presented further using three dimensional graphs. The red color shows the resultant accuracy of algorithms proposed by Agrawal, (2012) while the green color shows the resultant accuracy of the proposed algorithms in this thesis work. In these graphs the X-axis represents epsilon value ranging from 0.1 to 0.5, the Y-axis represents the gamma value ranging from 0.001 to 0.02 and Z –axis represents the recognition accuracy in percentage.

4.1 SCHEME 1: 30 samples of lower, middle and upper zones each has been used for cross validation.

For the lower zone 30 samples of each stroke have been taken which are categorized in lower zone as described earlier. Maximum accuracy for 3, 4, 5 fold cross validation is given in Table 4.1.

Table 4.1: Cross validation accuracy of lower zone for 3, 4, and 5 cross validation for 30 samples

***k*-Fold Cross Validation (SVM)**	**Maximum accuracy of Algorithm proposed in Agrawal, (2012)**	**Maximum Accuracy of Algorithm proposed in this work**
3	97.1%	95.7%
4	97.1%	95.7%
5	96.6%	96.1%

Three Dimensional graphs are shown in Figure 4.1, Figure 4.2 and Figure 4.3 for all the values of SVM parameters (gamma ranging from 0.001-0.02 and epsilon ranging from 0.1-0.5) have been drawn. These graphs show the variation between results obtained in Agrawal, (2012) and results obtained in this thesis work. It can be concluded from the graphs that the accuracy is better in Agrawal, (2012) as compared to results in this work for 3, 4 and 5 fold cross validation on a data set of 30 samples for lower zone.

It can be reasoned that the pre-processing phase contains the missing point interpolation step which needs at least 10 points to be implemented. In Agrawal, (2012) the Bindu of Gurmukhi character is bypassed and not processed but in this work it has been taken in the consideration. As a result, the accuracy rate becomes lower in the lower zone.

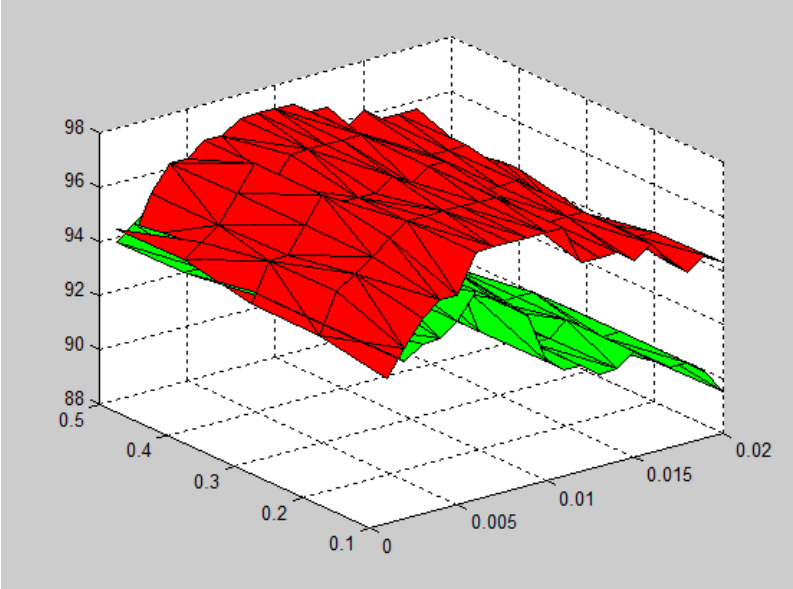


Figure 4.1: Surface graph showing variation in two methods for 3 fold cross validation on a data set of 30 samples for lower zone

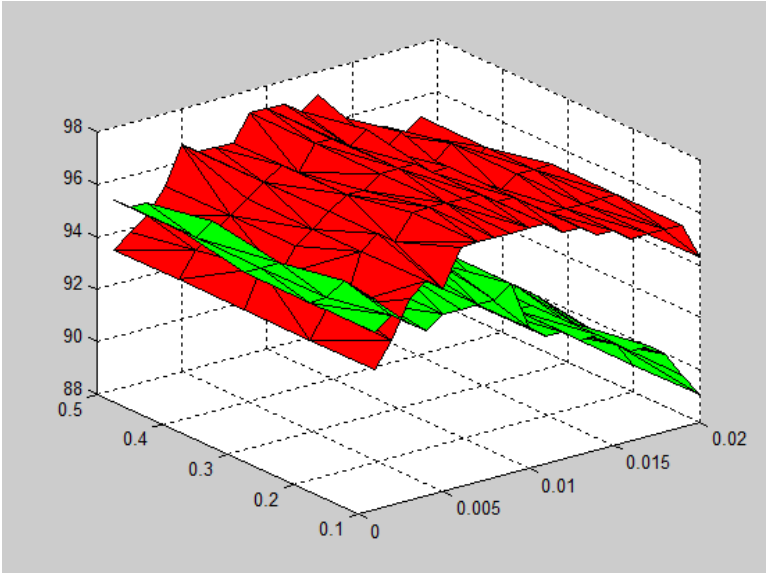


Figure 4.2: Surface graph showing variation in two methods for 4 fold cross validation on a data set of 30 samples for lower zone

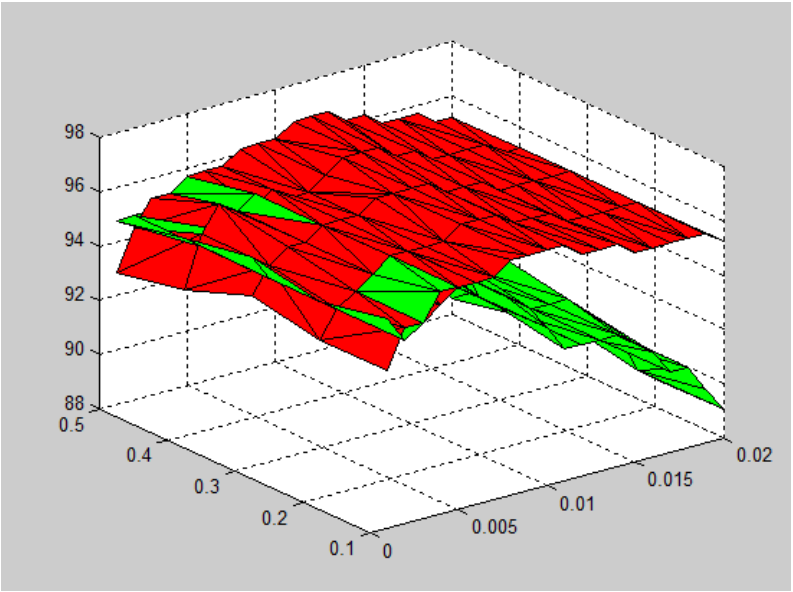


Figure 4.3: Surface graph showing variation in two methods for 5 fold cross validation on a data set of 30 samples for lower zone

For the middle zone 30 samples of each stroke have been taken. Maximum accuracy is given in the Table 5.1 for 3, 4 and 5 fold cross validation considering the parameters of the SVM.

Table 4.2: Cross validation accuracy of middle zone for 3, 4, and 5 cross validation for 30 samples.

***k*-Fold Cross Validation (SVM)**	**Maximum accuracy of Algorithm proposed in Agrawal, (2012)**	**Maximum Accuracy of Algorithm proposed in this work**
3	83.9%	89.1%
4	84.1%	90.2%
5	85.2%	90.8%

Three Dimensional graphs are shown in Figure 4.4, Figure 4.5 and Figure 4.6 for all the values of SVM parameters (gamma ranging from 0.001-0.02 and epsilon ranging from 0.1-0.5) have been drawn. These graphs show the variation between results obtained in Agrawal, (2012) and results obtained in this thesis work. It can be concluded from the graphs that the accuracy is better in this work as compared to results in Agrawal, (2012) for 3, 4 and 5 fold cross validation on a data set of 30 samples for middle zone.

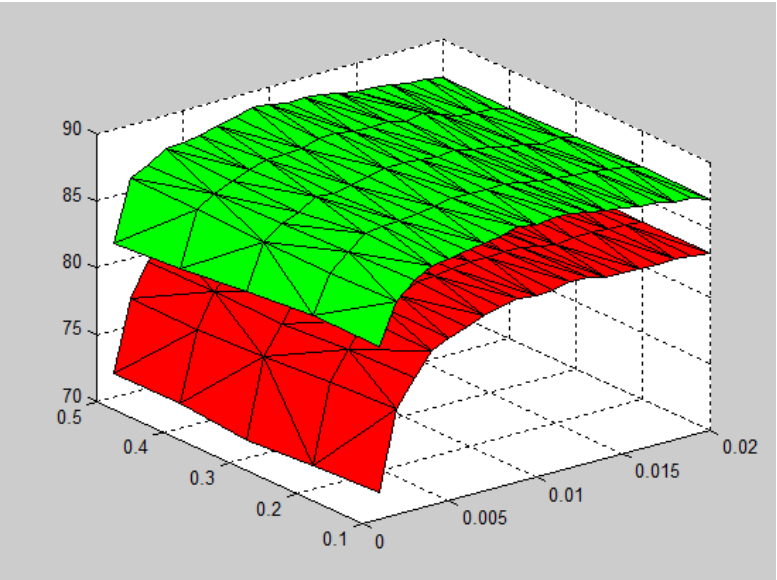


Figure 4.4: Surface graph showing variation in two methods for 3 fold cross validation on a data set of 30 samples for middle zone

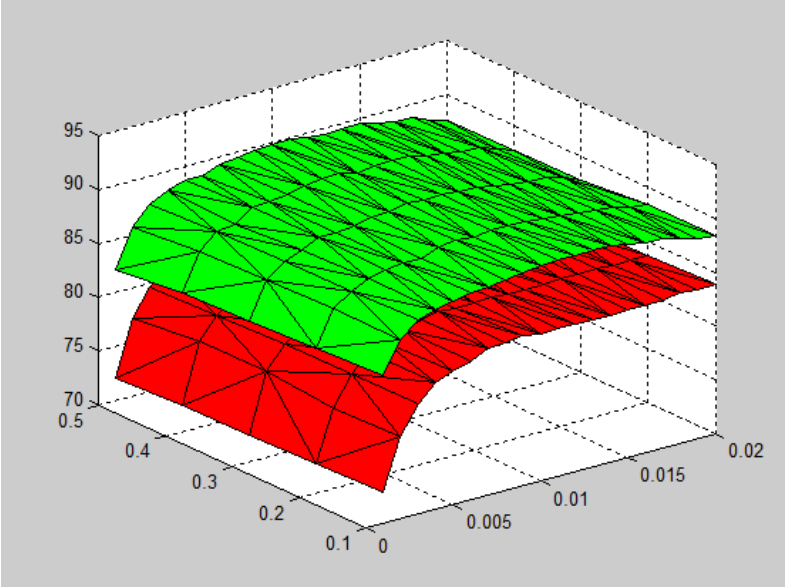


Figure 4.5: Surface graph showing variation in two methods for 4 fold cross validation on a data set of 30 samples for middle zone

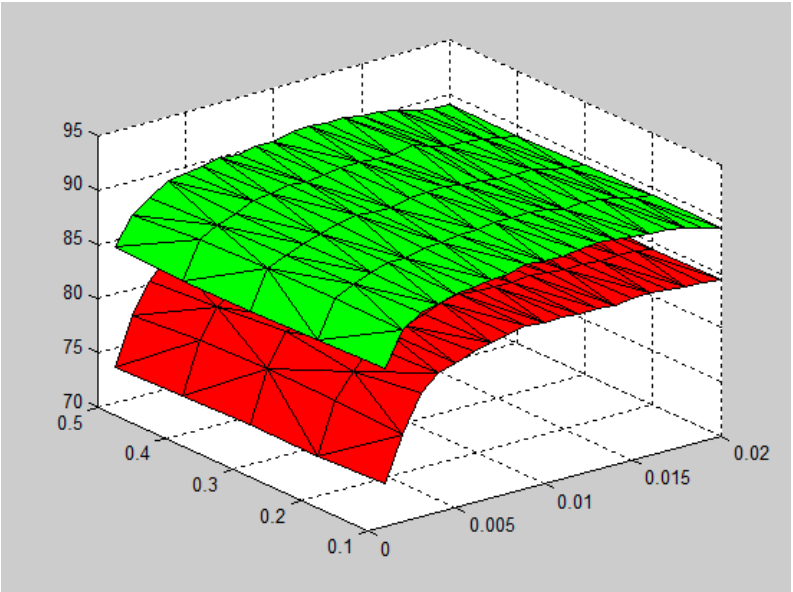


Figure 4.6: Surface graph showing variation in two methods for 5 fold cross validation on a data set of 30 samples for middle zone

In the upper zone 30 samples of each stroke have been taken. Maximum accuracy is given in the Table 5.1 for 3, 4, 5 fold cross validation.

Table 4.3: Cross validation accuracy of upper zone for 3, 4, and 5 cross validation for 30 samples

***k*-Fold Cross Validation (SVM)**	**Maximum accuracy of Algorithm proposed in Agrawal, (2012)**	**Maximum Accuracy of Algorithm proposed in this work**
3	87.3%	88.1%
4	86.4%	88.8%
5	86.2%	88.6%

Three Dimensional graphs are shown in Figure 4.7, Figure 4.8 and Figure 4.9 for all the values of SVM parameters (gamma ranging from 0.001-0.02 and epsilon ranging from 0.1-0.5) have been drawn. These graphs show the variation between results obtained in Agrawal, (2012) and results obtained in this thesis work. It can be concluded from the graphs that the accuracy is better in this work as compared to results in Agrawal, (2012) for 3, 4 and 5 fold cross validation on a data set of 30 samples for upper zone.

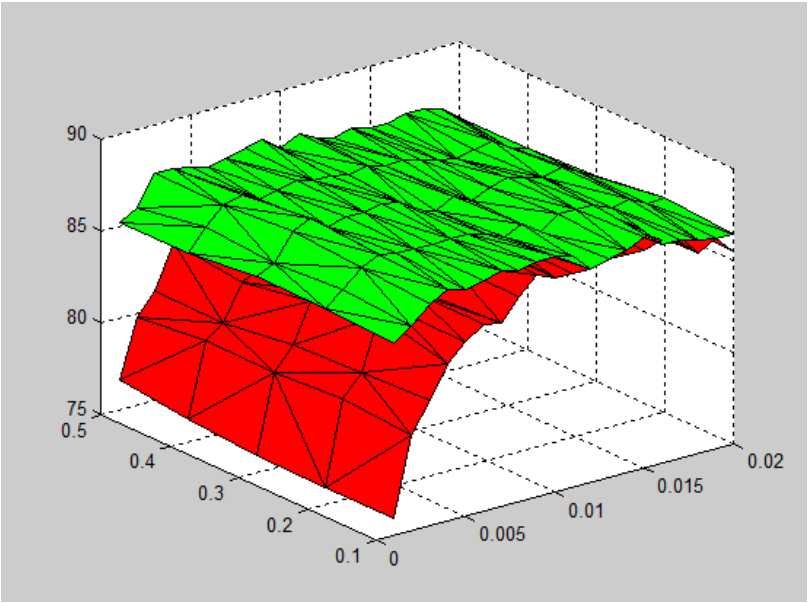


Figure 4.7: Surface graph showing variation in two methods for 3 fold cross validation on a dataset of 30 samples for upper zone

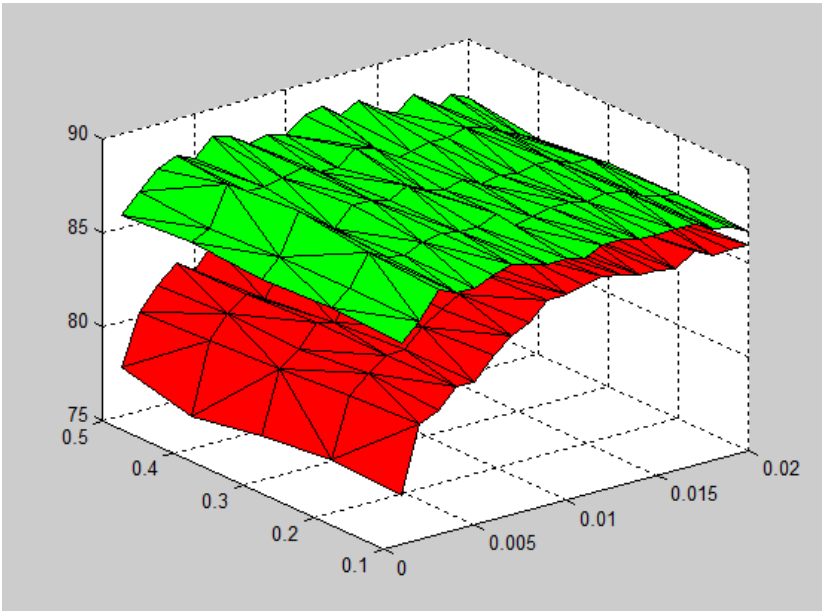


Figure 4.8: Surface graph showing variation in two methods for 4 fold cross validation on a dataset of 30 samples for upper zone

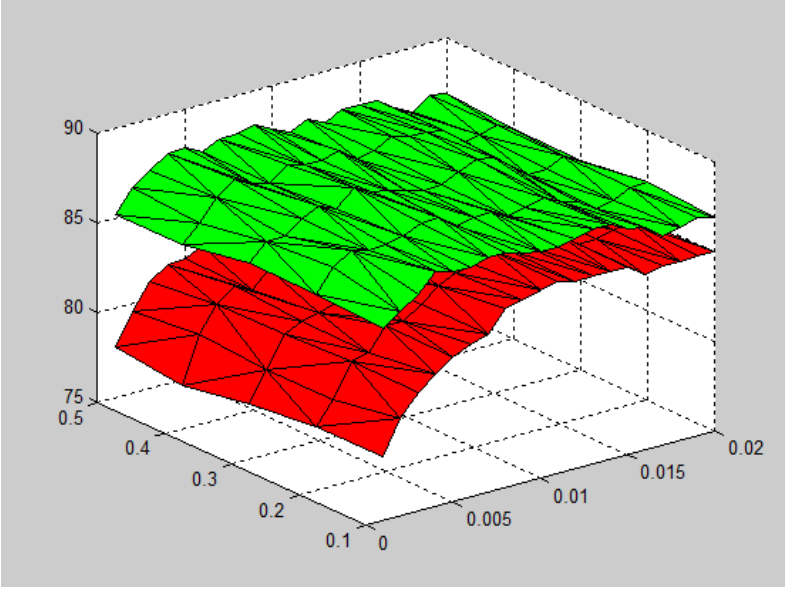


Figure 4.9: Surface graph showing variation in two methods for 5 fold cross validation on a dataset of 30 samples for upper zone

4.2 SCHEME 2: 50 samples in lower, middle and upper zones each has been taken for the cross validation. And the results are shown below.

For the lower zone 50 samples of each stroke have been taken. Maximum accuracy is given in the Table 5.1 for 3, 4, 5 fold cross validation.

Table 4.4: Cross validation accuracy of lower zone for 3, 4, and 5 cross validation for 50 samples

***k*-Fold Cross Validation (SVM)**	**Maximum accuracy of Algorithm proposed in Agrawal, (2012)**	**Maximum Accuracy of Algorithm proposed in this work**
3	92.3%	96.3%
4	93.7%	96.3%
5	93.4%	96.3%

Three Dimensional graphs are shown in Figure 4.10, Figure 4.11 and Figure 4.12 for all the values of SVM parameters (gamma ranging from 0.001-0.02 and epsilon ranging from 0.1-0.5) have been drawn. These graphs show the variation between results obtained in Agrawal, (2012) and results obtained in this thesis work. It can be concluded from the graphs that the accuracy is better in this work as compared to

results in Agrawal, (2012) for 3, 4 and 5 fold cross validation on a data set of 50 samples for lower zone.

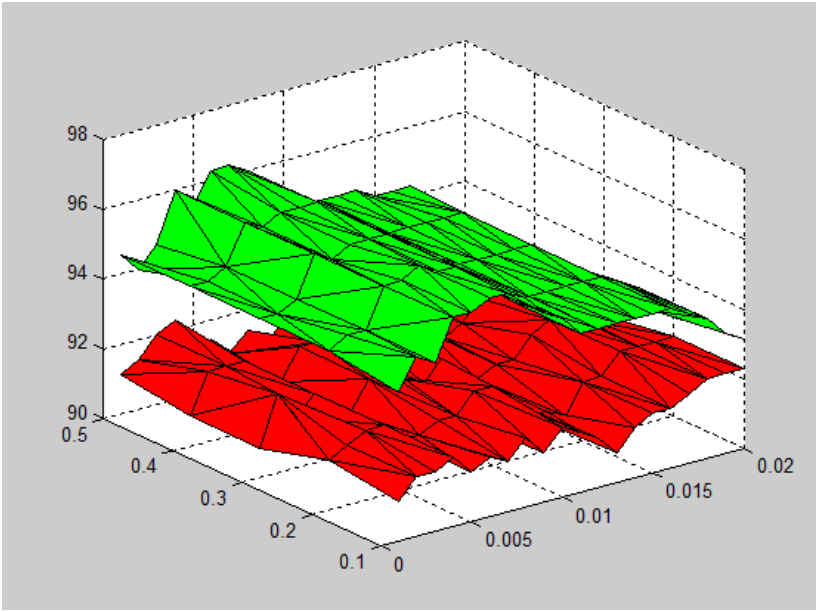


Figure 4.10: Surface graph showing variation in two methods for 3 fold cross validation on a dataset of 50 samples for lower zone

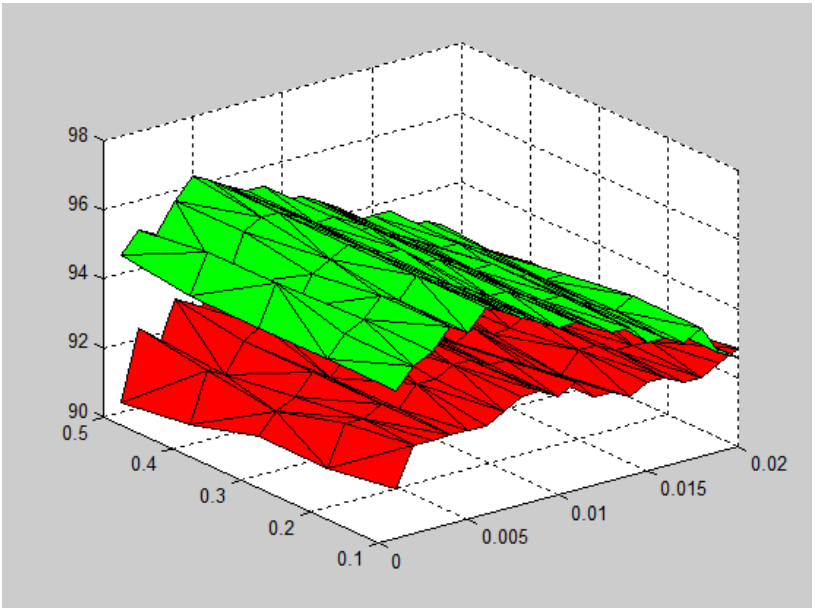


Figure 4.11: Surface graph showing variation in two methods for 4 fold cross validation on a dataset of 50 samples for lower zone

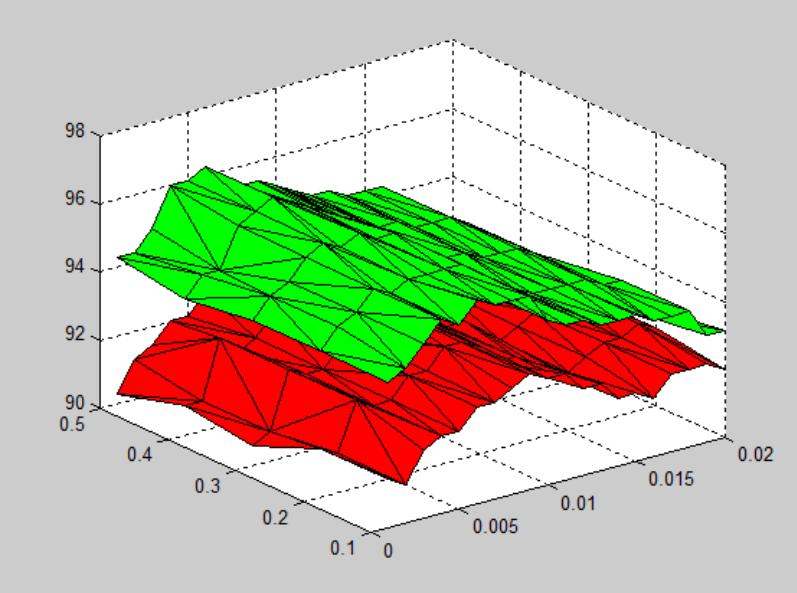


Figure 4.12: Surface graph showing variation in two methods for 5 fold cross validation on a dataset of 50 samples for lower zone

For the middle zone 50 samples of each stroke have been taken. Maximum accuracy is given in the Table 5.1 for 3, 4, 5 fold cross validation.

Table 4.5: Cross validation accuracy of middle zone for 3, 4, and 5 cross validation for 50 samples

***k*-Fold Cross Validation (SVM)**	**Maximum accuracy of Algorithm proposed in Agrawal, (2012)**	**Maximum Accuracy of Algorithm proposed in this work**
3	87.2%	91.8%
4	87.5%	91.9%
5	88.5%	92.6%

Three Dimensional graphs are shown in Figure 4.13, Figure 4.14 and Figure 4.15 for all the values of SVM parameters (gamma ranging from 0.001-0.02 and epsilon ranging from 0.1-0.5) have been drawn. These graphs show the variation between results obtained in Agrawal, (2012) and results obtained in this thesis work. It can be concluded from the graphs that the accuracy is better in this work as compared to results in Agrawal, (2012) for 3, 4 and 5 fold cross validation on a data set of 50 samples for middle zone.

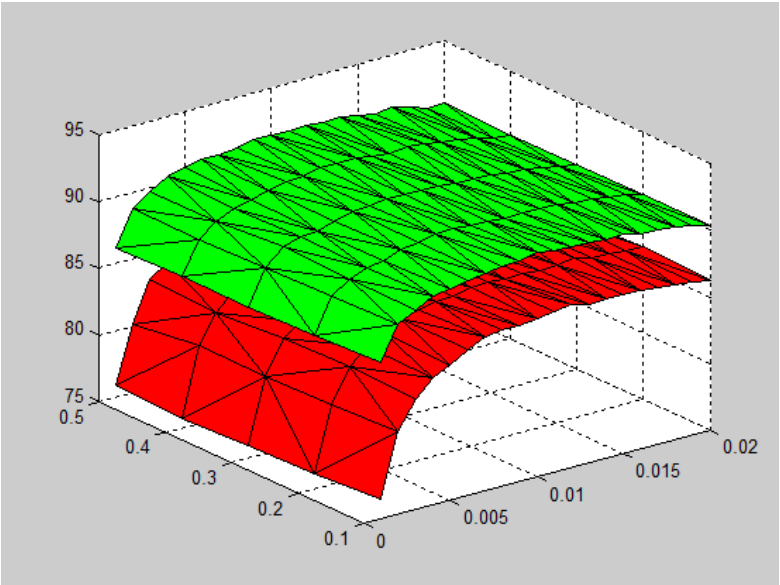


Figure 4.13: Surface graph showing variation in two methods for 3 fold cross validation on a dataset of 50 samples for middle zone

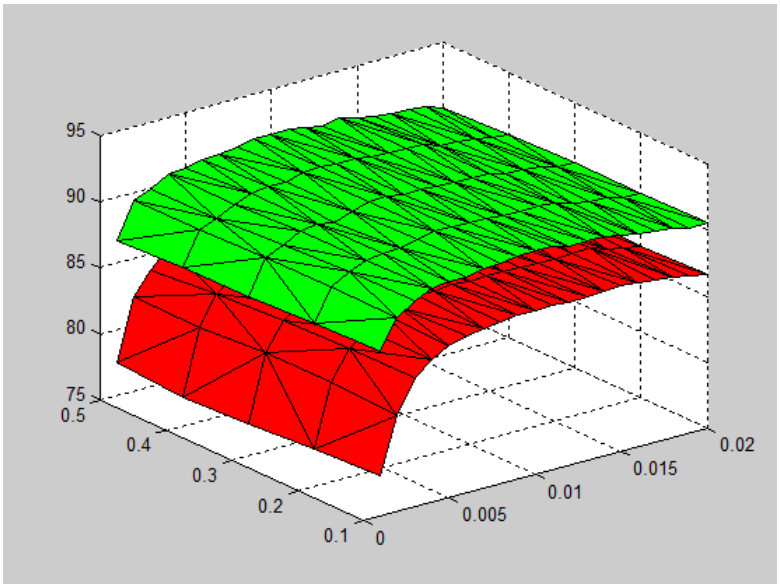


Figure 4.14: Surface graph showing variation in two methods for 4 fold cross validation on a dataset of 50 samples for middle zone

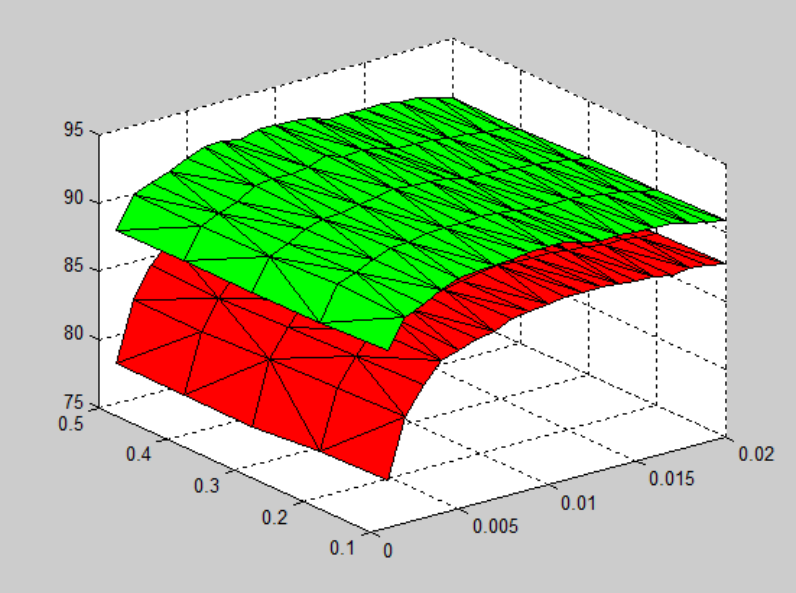


Figure 4.15: Surface graph showing variation in two methods for 5 fold cross validation on a dataset of 50 samples for middle zone

In the upper zone 50 samples of each stroke have been taken. Maximum accuracy is given in the Table 5.1 for 3, 4, 5 fold cross validation.

Table 4.6: Cross validation accuracy of upper zone for 3, 4, and 5 cross validation for 50 samples

***k*-Fold Cross Validation (SVM)**	**Maximum accuracy of Algorithm proposed in Agrawal, (2012)**	**Maximum Accuracy of Algorithm proposed in this work**
3	85.7%	86.2%
4	85.4%	87.7%
5	85.3%	87.6%

Three Dimensional graphs are shown in Figure 4.16, Figure 4.17 and Figure 4.18 for all the values of SVM parameters (gamma ranging from 0.001-0.02 and epsilon ranging from 0.1-0.5) have been drawn. These graphs show the variation between results obtained in Agrawal, (2012) and results obtained in this thesis work. It can be concluded from the graphs that the accuracy is better in this work as compared to results in Agrawal, (2012) for 3, 4 and 5 fold cross validation on a data set of 50 samples for upper zone.

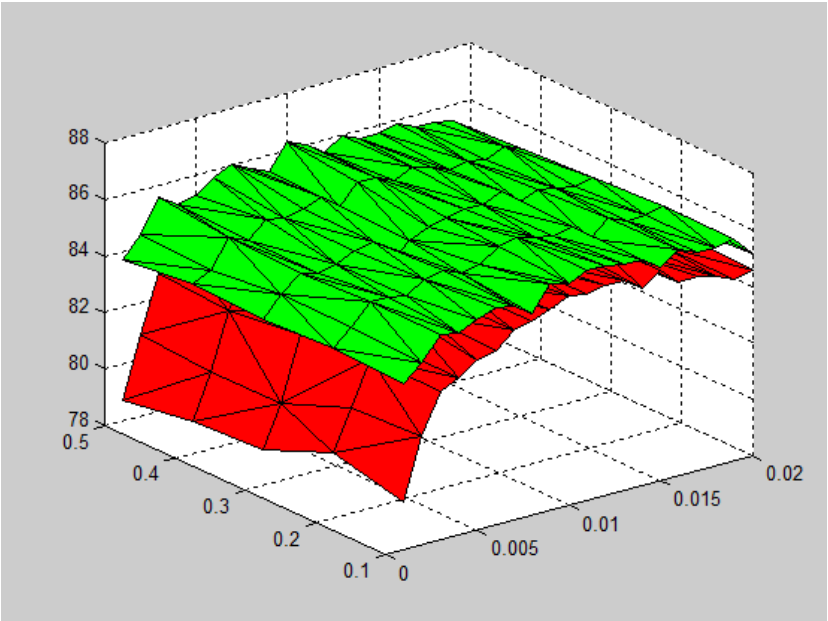


Figure 4.16: Surface graph showing variation in two methods for 3 fold cross validation on a dataset of 50 samples for upper zone

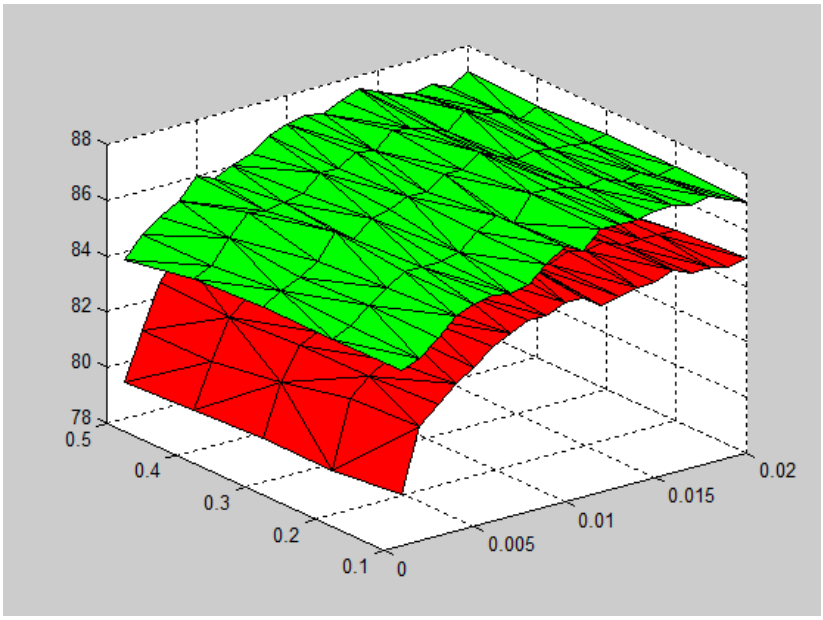


Figure 4.17: Surface graph showing variation in two methods for 4 fold cross validation on a dataset of 50 samples for upper zone

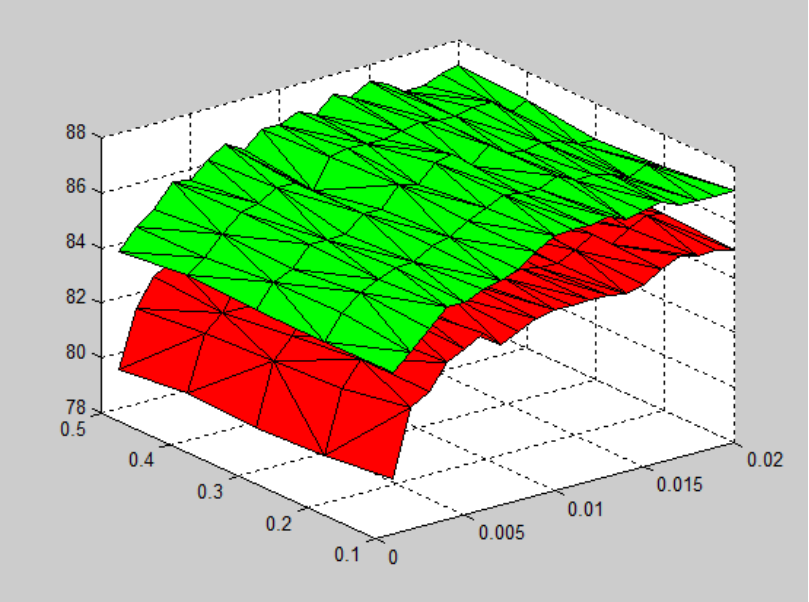


Figure 4.18: Surface graph showing variation in two methods for 5 fold cross validation on a dataset of 50 samples for upper zone

4.3 SCHEME 3: 70 samples in lower, middle and upper zones each has been taken for the cross validation. And the results are shown below.

For the lower zone 70 samples of each stroke have been taken. Maximum accuracy is given in the Table 5.1 for 3, 4, 5 fold cross validation.

Table 4.7: Cross validation accuracy of lower zone for 3, 4, and 5 cross validation for 70 samples

***k*-Fold Cross Validation (SVM)**	**Maximum accuracy of Algorithm proposed in Agrawal, (2012)**	**Maximum Accuracy of Algorithm proposed in this work**
3	94.6%	96.7%
4	95.5%	96.5%
5	96.5%	95.3%

Three Dimensional graphs are shown in Figure 4.19, Figure 4.20 and Figure 4.21 for all the values of SVM parameters (gamma ranging from 0.001-0.02 and epsilon ranging from 0.1-0.5) have been drawn. These graphs show the variation between results obtained in Agrawal, (2012) and results obtained in this thesis work. It can be concluded from the graphs that the accuracy is better in this work as compared to

results in Agrawal, (2012) for 3, 4 and 5 fold cross validation on a data set of 70 samples for lower zone.

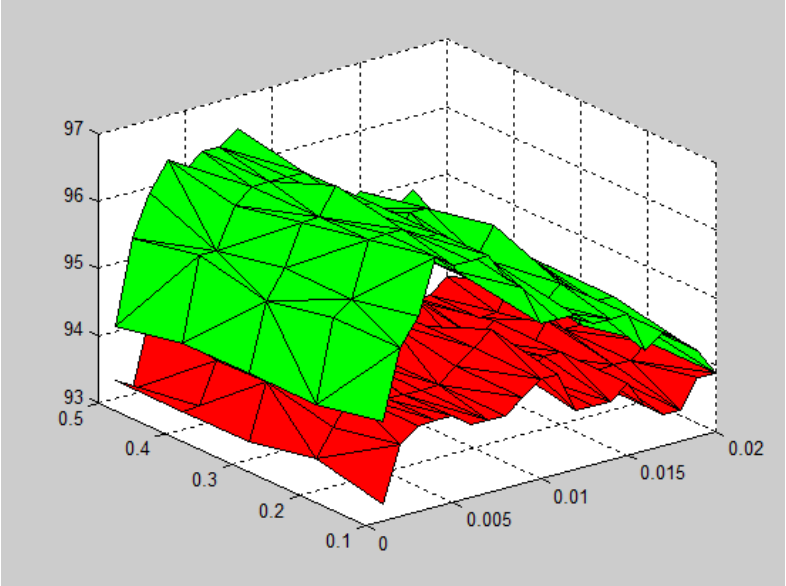


Figure 4.19: Surface graph showing variation in two methods for 3 fold cross validation on a dataset of 70 samples for lower zone

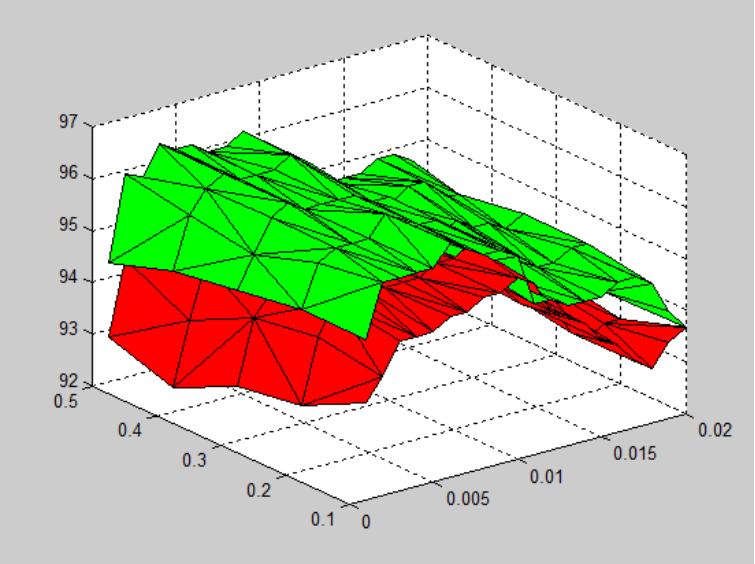


Figure 4.20: Surface graph showing variation in two methods for 4 fold cross validation on a dataset of 70 samples for lower zone

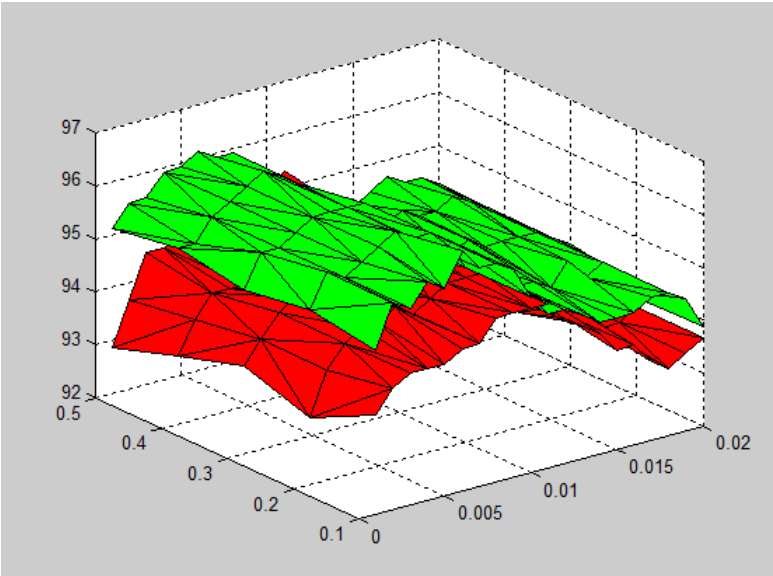


Figure 4.21: Surface graph showing variation in two methods for 5 fold cross validation on a dataset of 70 samples for lower zone

For the middle zone 70 samples of each stroke have been taken. Maximum accuracy is given in the Table 5.1 for 3, 4, 5 fold cross validation.

Table 4.8: Cross validation accuracy of middle zone for 3, 4, and 5 cross validation for 70 samples

***k*-Fold Cross Validation (SVM)**	**Maximum accuracy of Algorithm proposed in Agrawal, (2012)**	**Maximum Accuracy of Algorithm proposed in this work**
3	89.3%	92.8%
4	89.9%	93.4%
5	90.3%	93.6%

Three Dimensional graphs are shown in Figure 4.22, Figure 4.23 and Figure 4.24 for all the values of SVM parameters (gamma ranging from 0.001-0.02 and epsilon ranging from 0.1-0.5) have been drawn. These graphs show the variation between results obtained in Agrawal, (2012) and results obtained in this thesis work. It can be concluded from the graphs that the accuracy is better in this work as compared to results in Agrawal, (2012) for 3, 4 and 5 fold cross validation on a data set of 70 samples for middle zone.

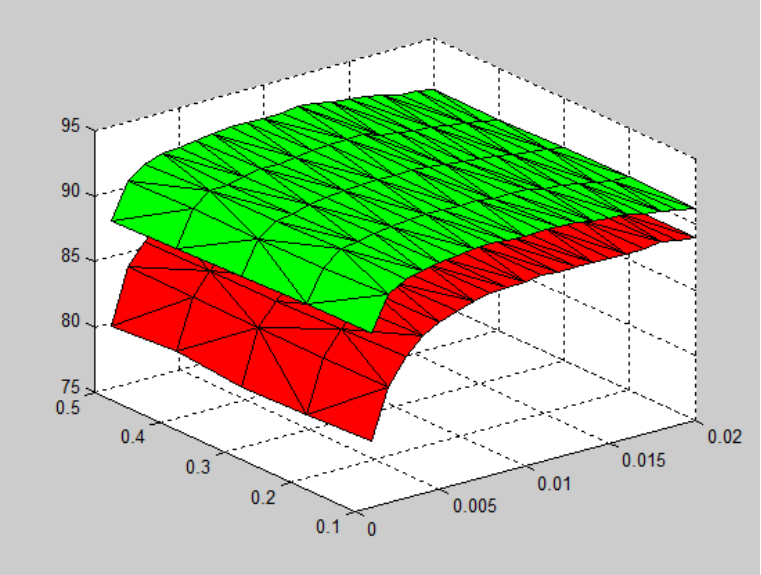


Figure 4.22: Surface graph showing variation in two methods for 3 fold cross validation on a dataset of 70 samples for middle zone

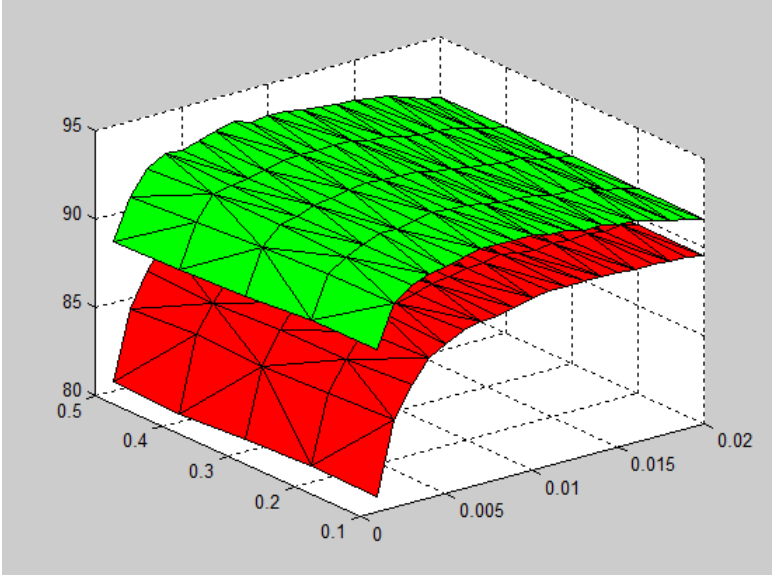


Figure 4.23: Surface graph showing variation in two methods for 4 fold cross validation on a dataset of 70 samples for middle zone

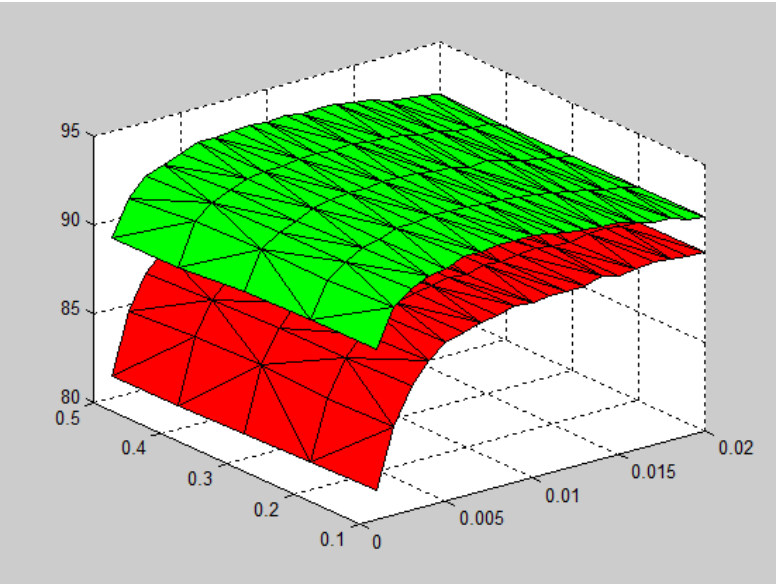


Figure 4.24: Surface graph showing variation in two methods for 5 fold cross validation on a dataset of 70 samples for middle zone

For the upper zone 70 samples of each stroke has been taken. Maximum accuracy is given in the Table 5.1 for 3, 4, 5 fold cross validation.

Table 4.9: Cross validation accuracy of upper zone for 3, 4, and 5 cross validation for 70 samples

***k*-Fold Cross Validation (SVM)**	**Maximum accuracy of Algorithm proposed in Agrawal, (2012)**	**Maximum Accuracy of Algorithm proposed in this work**
3	85.6%	88.0%
4	86.1%	88.5%
5	86.0%	88.7%

Three Dimensional graphs are shown in Figure 4.25, Figure 4.26 and Figure 4.27 for all the values of SVM parameters (gamma ranging from 0.001-0.02 and epsilon ranging from 0.1-0.5) have been drawn. These graphs show the variation between results obtained in Agrawal, (2012) and results obtained in this thesis work. It can be concluded from the graphs that the accuracy is better in this work as compared to results in Agrawal, (2012) for 3, 4 and 5 fold cross validation on a data set of 70 samples for upper zone.

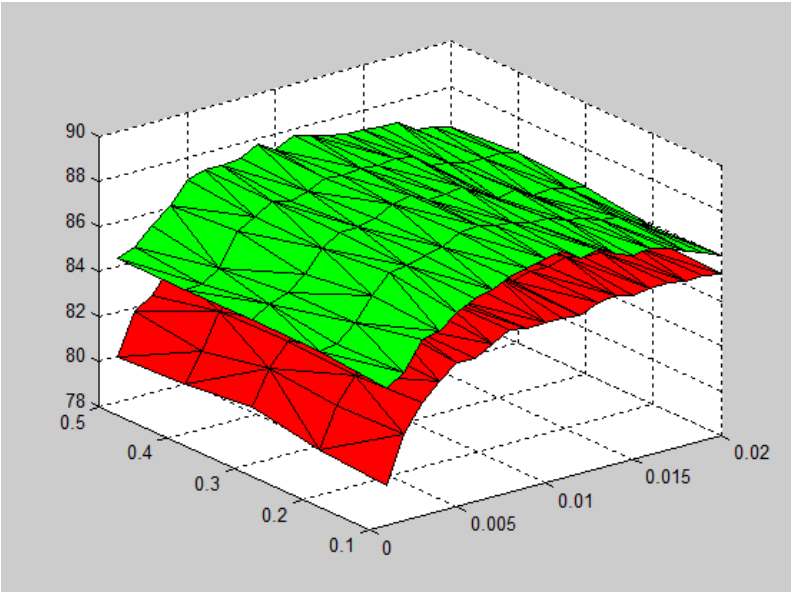


Figure 4.25: Surface graph showing variation in two methods for 3 fold cross validation on a dataset of 70 samples for upper zone

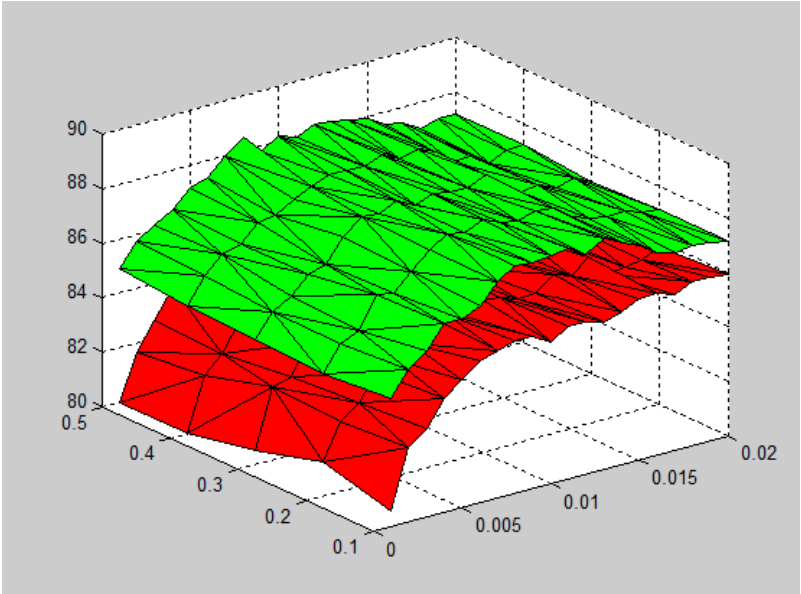


Figure 4.26: Surface graph showing variation in two methods for 4 fold cross validation on a dataset of 70 samples for upper zone

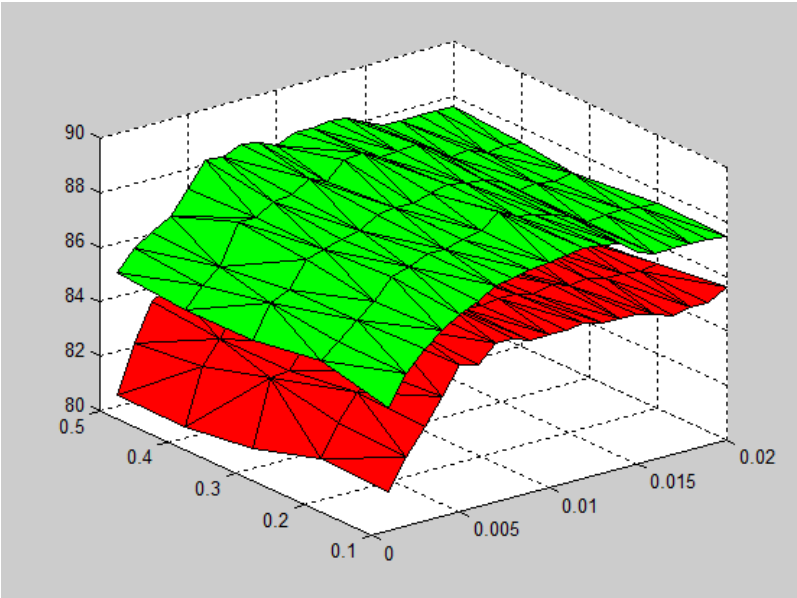


Figure 4.27: Surface graph showing variation in two methods for 5 fold cross validation on a dataset of 70 samples for upper zone

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

CONCLUSION

In this thesis, recognition accuracy of strokes has been improved by reconsidering the algorithms in pre-processing phase. The algorithm implemented in Agrawal, (2012) achieved a maximum accuracy of 90.4% but algorithms implemented in this thesis work have achieved an accuracy rate of 94.4% when a 75-25 ratio of training and testing data was considered. Algorithms implemented in this thesis are:

- Removal of duplicate points
- Normalization and centering
- Missing point interpolation
- Resampling of points

The main importance of the pre-processing phase is that after resampling, the 64 points should be at equal distance from each other and should retain its shape as it was written by the writer, which makes the difference of accuracy therefore in this thesis work we have got the higher accuracy rate. Thus, it can be concluded that pre-processing phase plays an important part in obtaining greater accuracy of the recognition system.

Future Scope

To increase the accuracy of recognition system we can add some new process in pre-processing phase as

- Smoothing of strokes can be done while the pen is moving from writing the stroke or after completion of stroke.
- Slant correction can also be added for increasing the accuracy as some writer has the habit of writing in cursive and in slant way.
- The complexity of the algorithms can be reduced to decrease the time of recognition.

- Strokes with less than 10 points captured can also be handled through extrapolation and through post-processing.
- We can add more features like curvature and direction for getting better accuracy.

REFERENCES

1) Aparna, K.H., Subramanian, V., Kasirajan, M., Prakash, G. V., Chakravarthy, V.S., and Madhvanath, S., 2004. Online handwriting recognition for Tamil. Ninth International Workshop on Frontiers in Handwriting Recognition, pp. 438-443.

2) Araki, N., Okuzaki M., Ishigaki, K. H., 2008. A Statistical Approach for Handwritten Character Recognition Using Bayesian Filter, 3rd International Conference on Innovative Computing Information and Control (ICICIC), pp. 194-198.

3) Arora, S., Bhattacharjee, D., Nasipuri, M., Basu, D. K. and Kundu, M., 2008. Combinig Multiple Feature Extraction Techniques for Handwriting Devnagari Character Recognition, Industrial and Information Systems, IEEE Region 10 Colloqium and the Third ICIIS, pp. 1-6.

4) Dungre, V. J. *et al*., 2010. A review of Research on Devnagari Character recognition, International Journal of Computer Applications, vol. 12, No, 2, pp. 8-15.

5) Guerfali, W. and Plamondon, R., 1993. Normalizing and restoring online handwriting", Pattern Recognition, Vol. 26, No. 3, pp. 419, 1993.

6) Homayoon, S.M., Beigi, K. N., Gregory J. Clary, and Subrahmonia. J., 1994. Size normalization in on-line unconstrained handwriting, IEEE Journal 0-8186-6950-0194.

7) Hosny, I., Abdou, S. and Fahmy, A., 2011. Using Advanced Hidden Markov Models for Online Arabic Handwriting Recognition, First Asian Conference on Pattern Recognition, pp.565-569.

8) Kumar, A. and Bhattacharya, S., 2010. Online Devnagari Isolated Character Recognition for the iPhone using Hidden Markov Model. Proceedings of the IEEE

Students' Technology Symposium, pp. 300-304.

9) Lehal, G. S., Singh, C., 2000. A Gurmukhi Script Recognition System, Proceeding of International Conference on Pattern Recognition, Vol. 2, pp. 557-560.

10) Lehal, G. S., Singh, C., 2002. A post Processor for Gurmukhi OCR, *Sadhana,* Vol. 27, Part 1, pp. 99-111.

11) Lehal, G. S., Singh, C., 2006. A Complete Machine printed Gurmukhi OCR, *Vivek.*

12) Mukherji, P., Rege, P., 2009. Shape Feature and Fuzzy Logic Based Offline Devnagari Handwritten Optical Character Recognition, Journal of Pattern Recognition Research 4, pp. 52-68.

13) Nair, A. and Leedham, C.G., 1991. Pre-processing of line codes for online recognition purposes, Electronics Letters, vol.2, no. 1, pp. 1-2.

14) Pal, U., Wakabayashi, Kimura. 2009. Comparative Study of Devnagari Handwritten Character Recognition using Different Feature and Classifier, 10th International Conference on Document Analysis and Recognition, pp. 1111-1115.

15) Pal, U. and Chaudhary, B.B., 2004. Indian Script Character Recognition, A survey, Pattern Recognition, Elsevier, pp. 1887-1899.

16) Pal, S., Mitra, J., Ghose, S., Banerjee P., 2007. A projection Based Statistical Approach for Handwritten Character Recognition, Proceedings of International Conference on Computational Intelligence and Multimedia Applications, vol. 2, pp. 404-408.

17) Sharma, A., Kumar, R. and Sharma, R. K., 2008. Online Handwritten Gurmukhi Character Recognition Using Elastic Matching, Congress on Signal and Image Processing, vol. 2, pp. 391-396.

18) Sharma, A., Kumar, R. and Sharma, R. K., 2009. Digit Extraction and Recognition from Machine Printed Gurumukhi Documents, Proceedings of the International Workshop on Multilingual OCR, MORC, Spain.

19) Kumar, R., Sharma, R. K., 2013. An Efficient Post-processing Algorithm for Online Handwritten Gurmukhi Character Recognition using Set Theory, International Journal for Pattern Recognition and Artificial Intelligence. Vol. 27, No. 4. Pp. 1353002 (17).

20) Singh, S., Aggarwal, A., and Dhir, R., 2012. Use of Gabor Filters for Recognition of Handwritten Gurmukhi Character, International Journal of Advance Research in Computer Science and Software Engineering. Vol. 2, Issue 5.

21) Tappert C.C., 1984. Adaptive on-line handwriting writing. Presented at The 7th International Conference on Pattern Recognition.

22) Tappert, C.C., Suen, C.Y. and Wakahara, T., 1988. Online handwriting recognition-A survey, Ninth International Conference on Pattern Recognition, vol.2, pp. 1123-1132.

23) Tappert, C.C. and Suen, C.Y., Wakahara, T., 1990. The state of the art in online handwriting recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 12, no. 8, pp. 787-808.

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

25) Zheng, J., and Zhu, G., 2006. On-Line handwriting signature recognition based on wavelet energy feature matching, The Sixth World Congress on Intelligent Control and Automation, vol. 2, pp. 9885-9888.

26) Agrawal, R., 2012. Recognition of Online Handwritten Gurmukhi Stroke using

Support Vector Machine, M.tech. thesis, School of Mathematics and Computer Applications, Thapar University, Patiala.

27) Sharma, A., 2009. Online handwritten Gurmukhi Character Recognition, Ph.D. thesis, School of Mathematics and Computer Applications, Thapar University, Patiala.

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

29) Support Vector Machine (SVM), Wikipedia website [Online]. Available: http://en.wikipedia.org/wiki/Support_vector_machine.

30) Unicode Inc., (1991-2013). Unicode 6.1 Character Code Charts [Online]. Available: http://www.unicode.org/charts/PDF/U0A00.pdf.

30) LIBSVM-Tools, A Library for Support Vector Machine [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/libsvm/