

**Recognition of Online Handwritten Gurmukhi  
Strokes using Support Vector Machine**

*A Thesis*

*Submitted in partial fulfillment of the  
Requirements for the award of the degree of*

**Master of Technology  
in  
Computer Science and Applications**

**Submitted by**

**Ashwani Kumar**

**(Roll No. 601303006)**

**Under the supervision of**

**Dr. R. K. Sharma**

*Professor, CSED*



**COMPUTER SCIENCE AND ENGINEERING DEPARTMENT**

**Thapar University**

**Patiala – 147004**

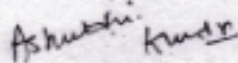
**July 2015**

## CERTIFICATE

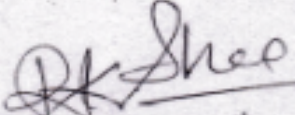
---

I hereby certify that the work which is being presented in the thesis entitled, "**Recognition of Online Handwritten Gurmukhi Strokes using Support Vector Machine**", in partial fulfillment of the requirements for the award of degree of Master of Technology in Computer Science and Application submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of **Dr. R. K. Sharma** and refers other researcher's work which are duly listed in the reference section.

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

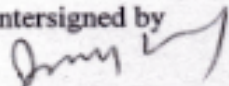
  
(Ashwani Kumar)

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

  
(R. K. Sharma)

Professor CSED

Countersigned by

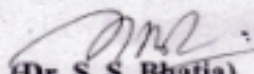
  
(Dr. Deepak Garg)

Head

Computer Science and Engineering Department

Thapar University

Patiala

  
(Dr. S. S. Bhatia)

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 guidance and encouragement of my supervisor **Dr. R. K. Sharma**, Professor, Computer Science and Engineering Department. I thank my supervisor for his time, patience, and valuable comments.

I am equally thankful to the entire faculty and staff members for their direct and indirect help, and cooperation which made my stay in Thapar University memorable.

Last but not least, I would like to thank my parents and friends for their wonderful love and encouragement, without their support none of this would have been possible.

Above all, I owe my reverence to almighty for the kindness who blessed me at finish of whole work.

(Ashwani Kumar)

## 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 Gurmukhi script using Support Vector Machine (SVM) by improving processes of pre-processing phase used for recognition. Gurmukhi is a script of Punjabi Language which is widely used across the world. This thesis is divided into five chapters. A brief outline of each chapter is given in the following paragraph.

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 the second chapter of literature review, a detailed literature survey on each phase of established procedure of online handwriting recognition has been presented.

Third 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. In the end feature extraction is explained.

Fourth chapter describes the recognition techniques that can be used for online handwritten recognition system. In this work SVM is used as a classifier for the recognition of handwritten strokes.

The results of the algorithm used in this thesis work are shown in the fifth Chapter. The

cross-validation testing has been used for calculating the accuracy. I have worked on 2-fold, 3-fold 4-fold, 5-fold and 6-fold cross-validation testing on a sample of 30, 50 and 100 of each class. Finally, in the last chapter result of my work is represented. Future scope of the online work is also mentioned.

# TABLE OF CONTENTS

---

<b>CERTIFICATE</b>	<b>i</b>
<b>ACKNOWLEDGEMENTS</b>	<b>ii</b>
<b>ABSTRACT</b>	<b>iii</b>
<b>TABLE OF CONTENTS</b>	<b>v</b>
<b>LIST OF FIGURES</b>	<b>vii</b>
<b>LIST OF TABLES</b>	<b>X</b>
<b>LIST OF ABBREVIATIONS</b>	<b>Xi</b>
<b>CHAPTER 1: INTRODUCTION</b>	<b>1-15</b>
1.1 About Gurmukhi Script	1
1.2 Classification of Handwriting Recognition	4
1.2.1 Online Handwriting Recognition	5
1.2.2 Offline Handwriting Recognition	5
1.3 Issues in Online Handwriting Recognition System	6
1.3.1 Handwriting Styles Variations	6
1.3.2 Constrained and Unconstrained Handwriting	8
1.3.3 Personal, Situational and Material Factors	9
1.3.4 Writer Dependent vs. Writer Independent Recognition Systems	9
1.4 Online Handwritten Character Recognition System	9
1.4.1 Data Collection	10
1.4.2 Pre-processing of Strokes	12

1.4.3 Feature Extraction or Computation of Features	12
1.4.4 Segmentation	13
1.4.5 Recognition Phase	14
<b>CHAPTER 2: LITERATURE REVIEW</b>	<b>14-18</b>
2.1 Work Related to Pre-processing	16
2.2 Work Related to Character Recognition	16
2.3 Work Related Gurmukhi Character Recognition	18
2.4 Problem Statement	18
<b>CHAPTER 3: DATA COLLECTION, PRE-PROCESSING AND FEATURE EXTRACTION</b>	<b>19-27</b>
3.1 Data Collection Phase	19
3.2 Pre-processing	21
3.2.1 Size Normalization and Centering of Handwritten Stroke	24
3.2.2 Smoothing of Stroke	24
3.2.3 Interpolating the Missing Points	25
3.2.4 Resampling of Points in a Stroke	26
3.3 Feature Selection	27
<b>CHAPTER 4: RECOGNITION OF GURMUKHI STROKES</b>	<b>28-40</b>
4.1 Introduction to SVM	28
4.1.1 SVM Kernels	29
4.1.2 SVM Applications	29
4.2 Recognition Gurmukhi Characters	29
4.2.1 <i>k</i> -Fold Cross-Validation for Training and Testing	39

<b>CHAPTER 5: RESULTS AND DISCUSSIONS</b>	<b>41-65</b>
5.1 Scheme 1: <i>k</i> -fold cross-validation on data set of 30 samples per class	42
5.2 Scheme 2: <i>k</i> -fold cross-validation on date set of 50 samples per class	45
5.3 Scheme 3: <i>k</i> -fold cross-validation on date set of 100 samples per class	49
5.4 Testing accuracy of each class tested for different writers	52
<b>CHAPTER 6: CONCLUSION AND FUTURE SCOPE</b>	<b>58-59</b>
<b>REFERENCES</b>	<b>60-65</b>

## LIST OF FIGURES

---

<b>Figure No.</b>	<b>Title of Figures</b>	<b>Page No.</b>
<b>Figure 1.1</b>	Handwriting recognition Classification.	4
<b>Figure 1.2</b>	Input sampling on tablet digitizer with the computer.	5
<b>Figure 1.3</b>	Gurmukhi characters written by the five different writers.	6
<b>Figure 1.4</b>	Variation in the Gurmukhi characters written by the same writer.	7
<b>Figure 1.5</b>	Handwriting in boxed discrete format.	8
<b>Figure 1.6</b>	Distinct styles of writing.	8
<b>Figure 1.7</b>	Phases of online handwriting recognition system.	10
<b>Figure 1.8</b>	Mainly used handwriting capturing devices.	11
<b>Figure 3.1</b>	Input Area.	20
<b>Figure 3.2</b>	Showing text file containing data of handwritten character shown in Figure 3.1.	20
<b>Figure 3.3</b>	Input handwritten Gurmukhi character in MS-Excel.	21
<b>Figure 3.4</b>	Steps in preprocessing phase.	22
<b>Figure 3.5</b>	Interface showing all preprocessing stages.	23
<b>Figure 3.6</b>	Figure. 3.6(a): Gurmukhi character of size smaller than 300×300 pixels. Figure 3.6(b): Transformation of Gurmukhi character (given in Figure 3.6(a)) into size normalization and centering.	29
<b>Figure 3.7</b>	Figure 3.7(a): Gurmukhi character with size normalization and centering in 300×300 pixels. Figure 3.7(b): Transformation of Gurmukhi character (given in Figure 3.6(a)) after applying smoothing.	25
<b>Figure 3.8</b>	Showing interpolating missing points.	26
<b>Figure 3.9</b>	Gurmukhi character of size smaller than 300×300 pixels. Figure 3.9(b): Transformation of Gurmukhi character (given in Figure 3.9(a)) into re-sampling of stroke into 64 points.	27
<b>Figure 4.1</b>	Idea behind support vectors.	28
<b>Figure 4.2</b>	Idea behind support vector machine.	28
<b>Figure 4.3</b>	Pre-processed patterns in SVM format.	35
<b>Figure 4.4</b>	Scaled file of pre-processed strokes.	36
<b>Figure 4.5</b>	Working of 3-fold cross-validation.	39
<b>Figure 4.6</b>	Model file of pre-processed strokes.	40
<b>Figure 5.1</b>	Figure showing 2-fold cross-validation on a data set of 30 samples per class.	43
<b>Figure 5.2</b>	Figure showing 3-fold cross-validation on a data set of 30	43

	samples per class.	
<b>Figure 5.3</b>	Figure showing 4-fold cross-validation on a data set of 30 samples per class.	44
<b>Figure 5.4</b>	Figure showing 5-fold cross-validation on a data set of 30 samples per class.	44
<b>Figure 5.5</b>	Figure showing 6-fold cross-validation on a data set of 30 samples per class.	45
<b>Figure 5.6</b>	Figure showing 2-fold cross-validation on a data set of 50 samples per class.	46
<b>Figure 5.7</b>	Figure showing 3-fold cross-validation on a data set of 50 samples per class.	47
<b>Figure 5.8</b>	Figure showing 4-fold cross-validation on a data set of 50 samples per class.	47
<b>Figure 5.9</b>	Figure showing 5-fold cross-validation on a data set of 50 samples per class.	48
<b>Figure 5.10</b>	Figure showing 6-fold cross-validation on a data set of 50 samples per class.	48
<b>Figure 5.11</b>	Figure showing 2-fold cross-validation on a data set of 100 samples per class.	50
<b>Figure 5.12</b>	Figure showing 3-fold cross-validation on a data set of 100 samples per class.	50
<b>Figure 5.13</b>	Figure showing 4-fold cross-validation on a data set of 100 samples per class.	51
<b>Figure 5.14</b>	Figure showing 5-fold cross-validation on a data set of 100 samples per class.	51
<b>Figure 5.15</b>	Figure showing 6-fold cross-validation on a data set of 100 samples per class.	52

## LIST OF TABLES

---

<b>Table No.</b>	<b>Title of Table</b>	<b>Page No.</b>
<b>Table 1.1</b>	Characters of Gurmukhi Script	2
<b>Table 1.2</b>	Unique Vowel Characters	4
<b>Table 1.3</b>	Six uncommon consonants in Gurmukhi	4
<b>Table 4.1</b>	Showing all Gurmukhi strokes with their IDs.	29
<b>Table 4.2</b>	Types of SVM and their option value	37
<b>Table 4.3</b>	Types of kernels and their option values	37
<b>Table 4.4</b>	Option available in LIBSVM and their default values	38
<b>Table 5.1</b>	Accuracy of Gurmukhi strokes for 2-fold, 3-fold, 4-fold, 5-fold and 6-fold cross-validation for 30 samples per class.	42
<b>Table 5.2</b>	Accuracy of Gurmukhi strokes for 2-fold, 3-fold, 4-fold, 5-fold and 6-fold cross-validation for 50 samples per class.	45
<b>Table 5.3</b>	Accuracy of Gurmukhi strokes for 2-fold, 3-fold, 4-fold, 5-fold and 6-fold cross-validation for 100 samples per class.	49
<b>Table 5.5</b>	Testing accuracy of each class tested by different writers.	53

## ABBREVIATIONS

---

HMM	Hidden Markov Model
OCR	Optical Character Recognition
OHCR	Online Handwritten Character Recognition
OHWR	Online Handwriting Recognition
PC	Personal Computer
PDA	Personal Digital Assistant
SVC	Support Vector Classifier
SVM	Support Vector Machine
SVR	Support Vector Regressor

# CHAPTER 1

---

---

## INTRODUCTION

---

---

Since the origination of computers we are witnessing a great deal of research activities in the field of computer. With the critical invention happening, the way of life of individuals is additionally changing. PCs have turn out to be more intense in processing and smaller in size at an extremely quick rate. The natural handwriting is a easy way of exchanging information between computers and human beings. It is a bit difficult to provide input data to computers for scripts like Devanagiri and Gurmukhi because of their complex typing nature. Two fast and characteristic methods for correspondence in between users and PCs are inputting the information through manually written archives and through speech .There are lots of limitations of speech recognition in noisy environment and mainly where privacy of an individual is much required. Here, In this work, we have concentrated on the issue of penmanship. Varieties in penmanship is one major issues and accomplishing high level of accuracy is another difficult task. Different writing styles cause these variations

For more than four decades, handwriting recognition is in exploration and has pulled in consideration of numerous scientists over the world. Incredible advances have been made via analysts here and quality of online penmanship based gadgets has been expanded. In gadgets like tablet PCs, the presence of online penmanship recognition is an extremely valuable. Today, these gadgets are in enormous interest and their usage has expanded in a great deal. The presence of online penmanship recognizer should give a characteristic method for correspondence in between clients and PCs for Gurmukhi, Devanagiri and other Asian scripts and it will build the use of individual computerized partner or tablet PCs in Indian dialects.

### **1.1 About Gurmukhi Script**

In the mid of 16th century the Gurmukhi script was formulated. Guru Nanak, the very first Sikh master promoted Gurmukhi Script. Gurmukhi is the most well-known script utilized for composing the Punjabi dialect in India. The name Gurmukhi is gotten from

the Old Punjabi term "Gurmukhi", signifying "from the soul of the Guru". There are 40 letters in Gurmukhi. Out of these three letters are different in Gurmukhi script as they build the basis of vowel and are not consonants. A vertical line is utilized to demonstrate the end of a sentence. Vertical bars demonstrate longer delay between sentences or sections. Table 1.1 shows these Gurmukhi characters.

**Table 1.1:** Characters of Gurmukhi Script

ਲ	LULA
ਉ	URHA
ਅ	ERHA
ੲ	EERHI
ਸ	SUSSA
ਹ	HAHA
ਕ	KUKA
ਖ	KHUKHA
ਗ	GUGGA
ਘ	GHUGGA
ਙ	UNGA
ਚ	CHUCHAA
ਛ	CHHUCHA
ਜ	JUJJA
ਝ	JHUJA
ਞ	YANZA

ਟ	TAINKA
ਠ	THUTHA
ਡ	DUDA
ਢ	DUDDA
ਣ	NANHA
ਤ	TUTTA
ਥ	TUTHA
ਧ	DHUDA
ਨ	NUNNA
ਪ	PHUPPA
ਘ	PUPHA
ਬ	BUBA
ਭ	BHUBA
ਮ	MUMA
ਯ	YIYYA
ਰ	RARAA
ਵ	VAVAA
ੜ	RAARA

From these character, first three are distinct vowels, they are hardly used their own. The remaining of the Gurmukhi script are the consonants. These matchless Vowel are shown in the Table 1.2.

**Table 1.2: Matchless Vowel Characters**

ੳ	URHA
ਅ	ERHA
ੲ	EERHI

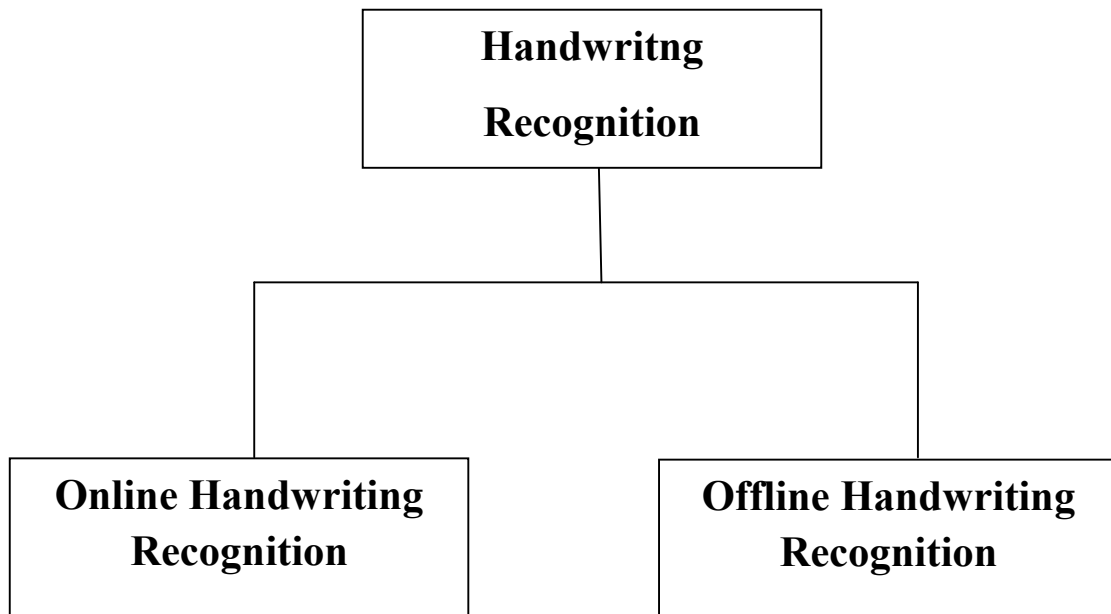
There are six more consonants build by putting a point at the bottom of the consonants. Table No.1.3. is showing these consonants.

**Table 1.3: Six uncommon consonants in Gurmukhi**

ਸ਼	SUSSA PAR BINDI
ਖ਼	KHUKHA PAR BINDI
ਗ਼	GUGGA PAR BINDI
ਜ਼	JAJJA PAR BINDI
ਫ਼	PHUPHA PAR BINDI
ਲ਼	LALLA PAR BINDI

## 1.2 Classification of Handwriting Recognition

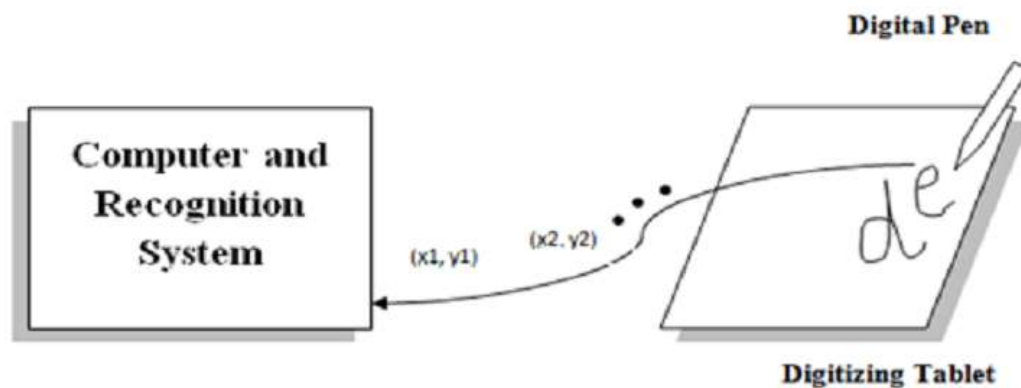
The recognition of handwriting can be classified into two categories shown in Figure 1.1.



**Figure 1.1:** Classification of Handwriting recognition

### **1.2.1 Online Handwriting Recognition System**

OHWR system refers to procedures and managing with the automatic processing of the stroke. Before any type of recognition process, the gained informatio is first preprocessed to diminish noise and to normalize stroke. In online handwriting recognition system, the current data, captured from input surface, is introduced to the system and recognition is completed in the meantime. Fundamentally, it gets the input in the form of x, y co-ordinates from an electronic pen touching a pressure touchy computerized tablet. There are various kind of applications of online handwriting recognition. It includes various kinds of interactive user-interfaces. Input sampling on tablet digitizer with the computer is shown in below Figure number 1.2.



**Figure 1.2:** Input sampling on the computer with the help of tablet digitizer.

### **1.2.2 Offline Handwriting Recognition System**

Offline character recognition takes a picture from a scanner or any other input data sources. The image is binarized based on the color pattern through threshold technique. When the picture is binarized whatever remains of the procedures for classification can be identical. The elements utilized as a part of the recognition are initially improved and afterward removed from bitmap pictures by method for computerized picture handling.

### 1.3 Issues in online handwriting character recognition

The online handwriting recognition can possibly enhance client and computer correspondence. Number of variability in penmanship shapes and lots of distortions brought by the digitizing procedure is unreliable, even the best recognizer is recognizing the character. The OHWR innovation is utilized for identification of Gurmukhi characters with the help of gadgets, for example, PDA, cross pads and tablet PCs. The input data is collected with the help of stylus, which is utilized as to write on the screen. After taking that input, the PC changes over the character content into digital content.

#### 1.3.1 Handwriting styles variations

Variations in handwriting style rely upon arrangements and diverse type of characters. These kind of varieties have geometrical in nature. There are various geometrical properties such as slant of strokes, position, size, aspect ratio of strokes.

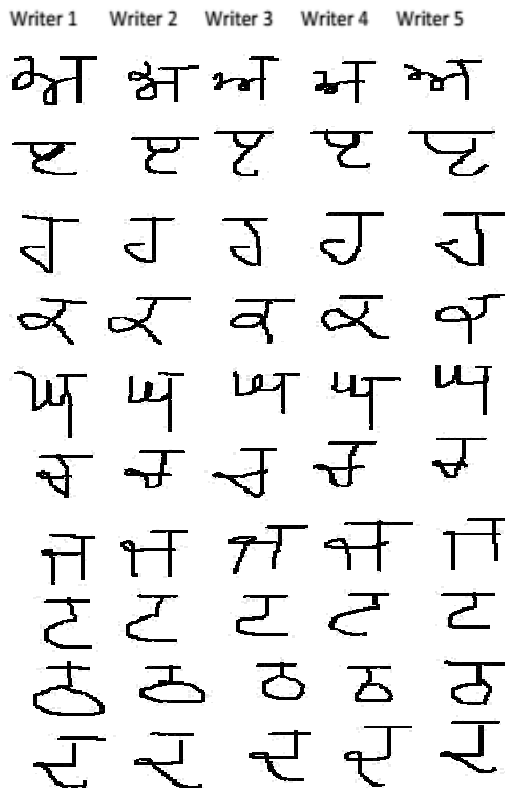
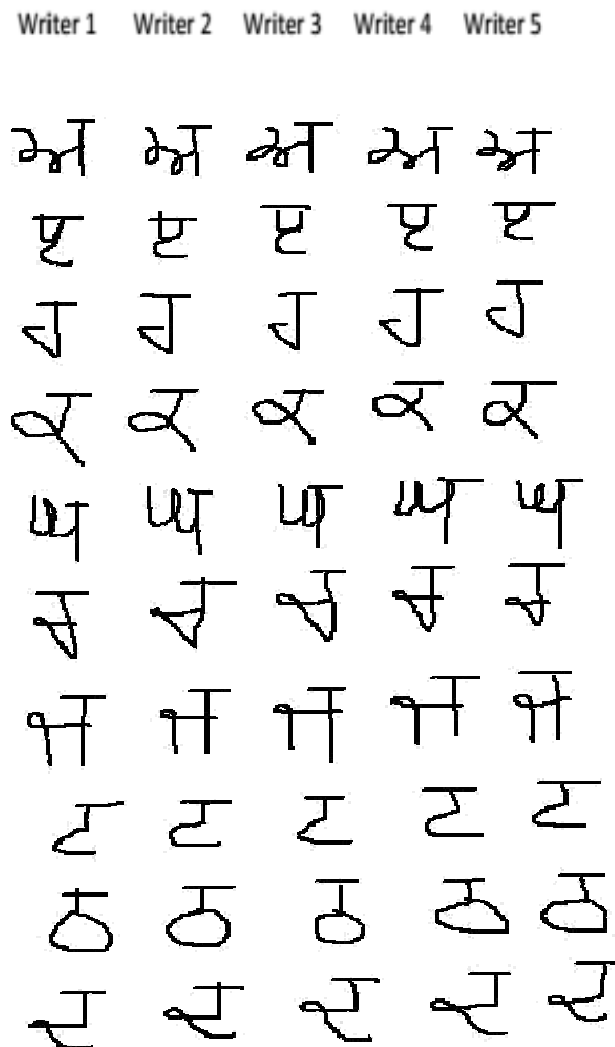


Figure 1.3: Gurmukhi characters written by the five different writers.

One can undoubtedly recognize that a varieties exists in every example of a character, such example share very high similarities. Figure 1.3, Sharma(2009) and Figure 1.4, Sharma(2009) depicts a portion of the distinctive styles that can be used to write characters of Gurmukhi script. Figure 1.3 shows the few examples of Gurmukhi characters from five unique scholars. One can take note of that variations exist in every example of a character. Figure 1.4 shows five examples of some characters of Gurmukhi script from individual author.



**Figure 1.4:** Variation in the Gurmukhi characters written by the same writer.

### 1.3.2 Variations in Constrained and Unconstrained handwriting

The styles of writing can be unconstrained and constrained. Unconstrained penmanship has bended nature. Constrained penmanship is of two types called spaced discrete and boxed discrete. Every character is composed inside a unique box called boxed discrete penmanship and it is shown in the Figure 1.5, Sharma(2009). If each and every character is composed independently and touches different characters called run-on discrete penmanship. It is watched that a large number of the individuals make blended strokes that incorporates a mixture of spaced, keep running on discrete and cursive style handwriting. Every writer has own particular rate of writing. Additionally, in cursive penmanship, no clear domain is determined between characters to recognize them. Different styles of writing is shown in the Figure 1.6, Sharma(2009).

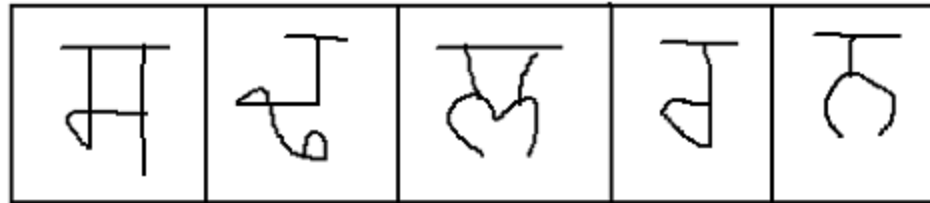


Figure 1.5: Handwriting in boxed discrete format.

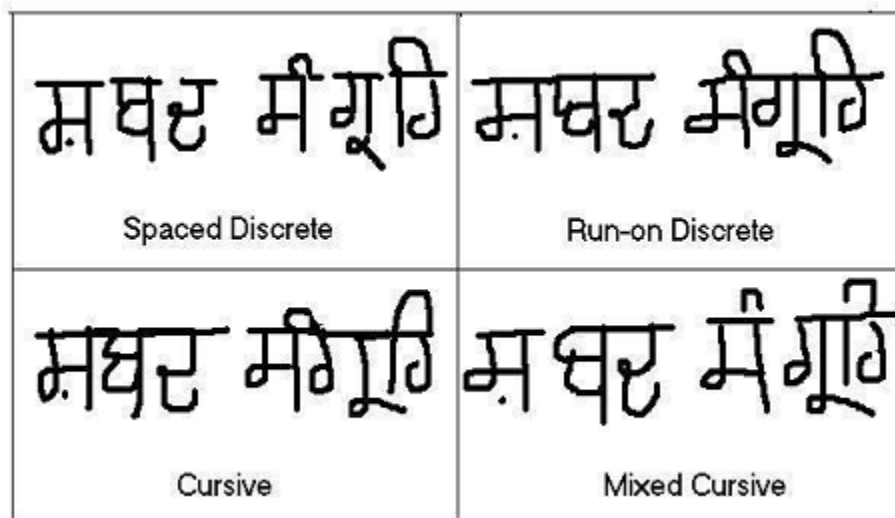


Figure 1.6: Distinct writing styles.

### **1.3.3 Personal, Material and Situational Factors**

Personal and situational factors make a great impact on the handwriting. A writer can write from both hands too but it is exceptional case. An essayist is either write with the left handed or right handed. Left handed and right handed individuals use distinctive positions and bearings. A decent recognition requires perfect handwriting in terms of shape. In lots of the cases, it has been noticed that perfect handwriting does not happen as handwriting of individuals additionally relies upon their profession. Situational elements rely upon the method for presentation of writing. The material elements rely on upon the equipment utilized as a part of writing.

### **1.3.4 Writer Dependent vs. Writer Independent Recognition Systems**

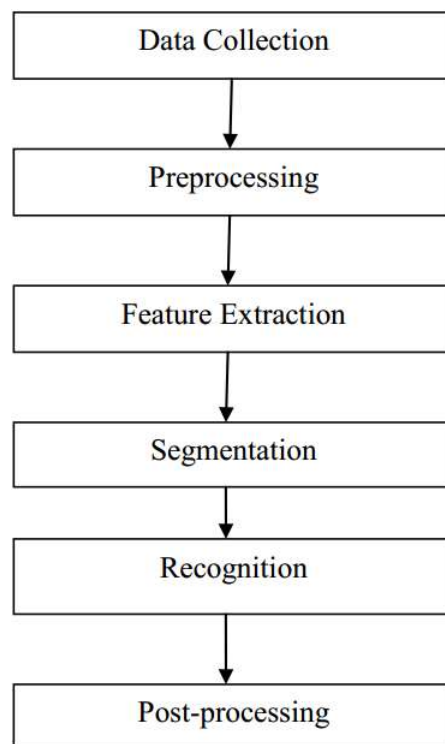
There are two categories of OHWR system called Writer dependent system and writer independent system. These classifications of writing dependent and writing independent system rely upon the information by which recognition system has been prepared. The writer dependent systems are trained in such a way that, their recognition limitations regarding stored styles. Consequently, this type of system is prepared with information gathered from the writers whom writing, to be recognized later on. On the other hand writer independent systems are developed for any writing styles. They also implied for the unknown penmanship styles. Likewise, system which is writer indepenedt requests all possible alternatives to store handwriting varieties in the datasets (Subrahmonia, 2000).

## **1.4 Online Handwritten Character Recognition System**

Digitizer, tablets were accessible in which resistive system and simple to the computerized change method was utilized, In the mid-seventies. It was conceivable to measure the pen tip utilizing these tablets. Various innovations were accessible for tablets or writing pads. These advances were in light of electronic or electromagnetic or electrostatic. The tablets with mix of input and output digitizer or show on the same surface were most basic in handwriting recognition. There are six major phases in online handwriting recognition system (Jaeger *et al.*, 2001; Suen *et al.*, 2003) shown in the Figure 1.7.

### 1.4.1 Data Collection

In this phase we need something such as cross pad, personal data assistant that can capture the writing as it is written. These devices are commonly electronic tablet or digitizer. A pen or styles is used to write on the screen of these devices. A styles has to perform two actions, PenDown and PenUp. The action performed after PenDown and before PenUp is called stroke. These input devices are shown in Figure 1.8.



**Figure 1.7:** Phases of online handwriting recognition system.

Commonly used personal digital assistants available today are Amstrad, Apple, Newton Message pad 2000, Casio, Z-7000, IBM, Motorola, Marco, Sharp, Zaurus ZR-5800FX, Sony, PIC-1000, Digital, Nokia and HTC etc.

Cross pads available today are Badger, Fujitsu, 325Point, Stylistic 500, Stylistic RF, ThinkPad 730T, K2000, K2100, Micros late, Telepad, SL, Toshiba, T200CS, Tusk, Zenith, IBM, IBM 2488 Pen-10 etc.



**Tablet PC**



**Cross Pad**



**Personal Digital Assistant**



**Digimemo**

**Figure 1.8:** Mainly used handwriting capturing devices.

### **1.4.2 Pre-processing of Strokes**

The preprocessing phase in handwriting recognition has come into existence because there are various problems in input such as noise or distortions. To remove these kind of distortion preprocessing is applied. These problems include irregular size of text, missing points, jitter and many more. There are five common steps in preprocessing (Jaeger *et al.*, 2001). These are:

- Size Normalization and Centering
- Interpolating missing points
- Smoothing
- Slant correction
- Resampling of points

The utilization of size normalization procedures in online handwriting recognition have been talked about by Beigi *et al.* (1994). Fast handwriting results into missing points. Missing points like these can be interpolated using different strategies, for example, Bezier insertion (Unser *et al.*, 1993). When we provide input, maximum time jitter is present. To remove this jitter present in handwriting smoothing is required for good shape of the stroke (Kavallieratou *et al.*, 2002). Smoothing generally averages a point with its neighbours. The slant rectification and normalizing is essential for handwriting acknowledgment (Madhanath *et al.*, 1999; Slavik and Govindaraju, 2001; Yimei *et al.*, 2000). Handwritten words are typically inclined because of the mechanism of handwriting and writer's personality. In resampling, made the points equidistant from neighbouring points as far as feasible.

### **1.4.3 Feature Extraction**

By selecting a suitable feature, one can get high recognition performance. The problem complexity can also be diminished if we find suitable features. There are two kinds of feature called low-level and high-level features. Features may change depends on the script. there is no standard strategy for processing features of a script (Jaeger *et al.*, 2001).

Rocha and Pavlidis (1994) outlined an feature extractor that diminished measurement of the problem and gave auxiliary description of a character shape that comprises of the

detail of its features and their extraordinary interrelations. In high-level features, they presented global features which demonstrate whether an information point has a place with a loop or near to intersections or cusp. Lehal and Singh(2000) examined features in machine printed Gurmukhi script and exhibited two arrangements of basic features, specifically, primary and secondary. Verma *et al.* (2004) exhibited an extractor procedure for online handwriting recognition that consolidates numerous characteristics of handwritten characters. This technique was free of character size and ready to extract features from extract features information without resizing.

#### **1.4.4 Segmentation**

In Segmentation data are represented on the level of stroke so that each of the stroke can be studied individually. It is one of the most important phase.

There are two types of segmentation:

- External Segmentation
- Internal Segmentation

External segmentation provides greater interactivity. It has been observed that segmentation study in offline handwriting recognition is better to understand segmentation in online handwriting recognition. Both recognition systems identify characters in word level segmentation. Next paraFigure reviews the literature of segmentation in offline handwriting recognition systems.

Lu and Shridar (1996) introduced an overview of strategies used as a part of segmenting characters for handwritten words. Their review comprises of three major parts, handprinted word segmentation, written by hand numeral segmentation and cursive word division. The present passage incorporates recent literature on division in online handwriting recognition.

Blachand and Artieres (2004) worked in poorly organized online transcribed documents segmentation utilizing probabilistic component syntaxes. Oliveira and Sabourin (2004) utilized segmentation based acknowledgment framework utilizing heuristic over segmentation.

### **1.4.5 Recognition Phase**

Syntactical and structural, Statistical, neural network, Support Vector Machine and elastic matching are mostly used handwriting recognition methods. In the Subsections, I have discussed online handwriting recognition methods.

#### **Neural Network Methods**

Neural systems is a parallel processing systems comprising of extremely large number of basic processors with numerous interconnections (Bellegarda *et al.*, 1993; Jain *et al.*, 2000). Neural system models endeavor to utilize some organizational standards, for example, learning, speculation, flexibility, fault tolerance, circulated representation.

#### **Structural and Syntactical Methods**

Where structures and grammar are considered, the method structural and syntactical comes into act. How the given pattern is constructed, provided by the structural methods. In syntactic method, an analogy is made between the formulation of patterns and the language syntaxes. The punctuation for each class must be derived from the accessible training examples (Jain *et al.*, 2000). The chain codes are broadly used structural representations of online penmanship.

#### **Statistical Methods**

By using this approach, every pattern is shown in the form of features and viewed as a point in vast dimensional space. This method is purely based on probabilities of classes and variations in handwriting. Statistical methods are of two types Parametric and Non-Parametric methods. By this method, handwritten samples are statistical variable from distribution and each class has its own distribution parameters. Parameter is selected through the training datasets. Hidden Markov Model (HMM) examples of parametric methods.

The HMMs got to be well known in online handwriting recognition system in mid 1990s. HMMs discovered to be suitable for cursive penmanship (Plamondon also, Srihari, 2000). The outcomes obtained utilizing HMMs are dependable as results are numerical values and there is always an extension to enhance recognition system using HMMs.

An instructional exercise by Rabiner (1989) presents prologue to HMMs. Bellegarda *et al.* (1994) added to a probabilistic system suitable for the derivation of a quick statistical mixture algorithm. This derivation shows about the same level of many-sided quality as other recognition techniques, for example, elastic matching, while being more adaptable and conceivably more robust. The methodology depends on a front-end processor that, dissimilar to ordinary character or stroke-based processing, expresses around a little elementary unit of penmanship called an frame. The frame is taking into account creating feature vectors speaking to every frame in one or more component spaces utilizing Gaussian - means bunching and mixture demonstrating in these spaces. They found that discrete density models prompted preferred results over continuous models where as it was switch in speech recognition.

### **Elastic Matching Methods**

This approach is a widely used in pattern recognition. Using elastic matching method, we can easily determine the similarity between two entities. In this approach, the pattern which we want to recognize is matched with the stored sample. This matching is demanding, but this approach requires faster processors.

In ahead of schedule works, Fujimoto utilized elastic matching of filtered images. In this framework a manually written FORTRAN program is digitized and changed over to ASCII code. After digitization, the patterns are thinned and afterward they are coded into headings to build sequence of points. A separation in light of a dynamic programming formula, like elastic matching, is computed (Fujimoto *et al.*, 1976).

#### **1.4.6 Post-processing**

In the phase of post-processing, strategy of correcting misclassified results is applied with the help of semantic knowledge. Post-processing is handling the output from shape acknowledgment. Post-processing plays an important role in OHWR system. In this phase with the help of akshras rules, we provide the ouput. It is also used to refine the results. We implement post-processing phase based on loop, headline, feature and dots exists in strokes.

## CHAPTER 2

---

---

### LITERATURE SERVEY

---

---

Literature servey provides the current status of the research going on in every field that is why it is so important. The fundamental thought to do this literature survey is to get the initiation of online handwriting recognition for scripts and to get a thought of diverse recognition methods received by distinctive researchers from everywhere throughout the world.

#### 2.1 Work Related to Pre-processing.

Tappert (1982) utilized algorithm based dynamic programming and contrasted information and letter models line by line. Every information is described by two parameter called vertical position and the digression of pen way at present point. Investigations included cautious composing and user-dependent databases of letter models.

Hosny *et al.* (2011) proposed pre-processing methods for Arabic characters using four steps:

- Removal of Duplicate Points: Any two points in are the same or not, is checked . If found to be the same, one of them is removed.
- Interpolating Points: for adding any missing point between two consecutive points.
- Smoothing: To eliminate the bend present in the stroke.
- Re-sampling: To make the points equidistant from its neighbours.

#### 2.2 Work Related to Character Recognition

Brown (1983) utilized component vectors and assessment of the length of the word to represent to the word attributes. The recognition took place using extrated component vectors utilizing the k-closest neighbor system. The recognizer was prepared on the

information of the clients and tried on information of the other two. 64.2% to 81.3% acknowledgment rates is achieved.

Kurtberg (1987) proposed a system to perceive constrained discrete handwritten images in view of elastic coordinating against an arrangement of models produced by individual authors. He likewise presented component investigation with versatile matching to wipe out far-fetched prototypes.

Noubound and Plamondon (1991) utilized basic way to deal with perceive online handwritten characters. They introduced an ongoing requirement free hand printed character recognition framework in light of a basic methodologies.

Powalka and Noubound (1994) added to a words-based recognizer. This recognizer is utilized an exceptionally constrained arrangement of components comprising of group. Handwriting of 19 clients was assessed, every written work 200 words.

Lu and Shridar (1996) introduced an overview of strategies used as a part of segmenting characters for handwritten words. Their review comprises of three major parts, handprinted word segmentation, written by hand numeral segmentation and cursive word division. The present passage incorporates recent literature on division in online handwriting recognition.

Hu *et al.* (1997) utilized HMMs for essayist independent online penmanship acknowledgment framework utilizing blend of point situated and stroke oriented components.

Li and Yeung (1997) introduced a way to deal with online manually written alphanumeric character acknowledgment in light of successive penmanship signals.

Cho (1997) displayed three advanced classifiers based on neural network to understand complex pattern recognition issues that incorporate different classifier, HMMs based cross breed classifier and versatile self-sorting out guide classifier. Identification of this work was with unconstrained written by hand numerals.

Blachand and Artieres (2004) worked in poorly organized online transcribed documents segmentation utilizing probabilistic component syntaxes. Oliveira and Sabourin (2004)

utilized segmentation based acknowledgment framework utilizing heuristic over segmentation.

Noubound and Plamondon (1997) utilized basic way to deal with perceive online handwritten characters. They introduced an ongoing requirement free hand printed character recognition framework in light of a basic methodologies.

Kimura *et al.* (1997) introduced a two-stage various leveled framework comprising of a statistical simulated neural network to perceive a substantial number of classifications and having comparable classification sets. The recognition results are 98.6% and 97.5% for the input data and the test informations, separately.

### **2.3 Work Related Gurmukhi Character Recognition**

Sukhpreet Singh (2012) In this paper, Gurmukhi character recognition for seprated characters is proposed. Gabor Filter based technique for feature extraction is used. The database consists of 200 patterns of every of 35 characters of the Punjabi script taken from distint writer. These samples are first pre-processed and normalized to 30\*30 sizes. The best accuracy gained is 95.9% as 5-fold cross-validation of entire database with SVM classifier having RBF kernel.

### **2.4 Problem Statement**

In view of the survey completed in this area, I have concluded that there is not so much work accomplished for Gurmukhi script for perceiving the online OHWR of Gurmukhi numerals utilizing SVM as a classifier. SVM is one of the mostly used and effective classifier. Before SVM classification pre-processing is applied because pre-processing helps to improve the acknowledgment process. The more is the preprocessing done, the better is the recognition will take place. Additionally, the more is the amount of pre-processing done, the simpler it gets to be to perceive the character for the recognition accuracy. The present study is completed concentrating on preprocessing and SVM classification. But, I found various problems in classification because it has various similar classes. The challenge in front of me is to correctly classify the input data using SVM.

## CHAPTER 3

---

---

### COLLECTION OF DATA AND PRE-PROCESSING

---

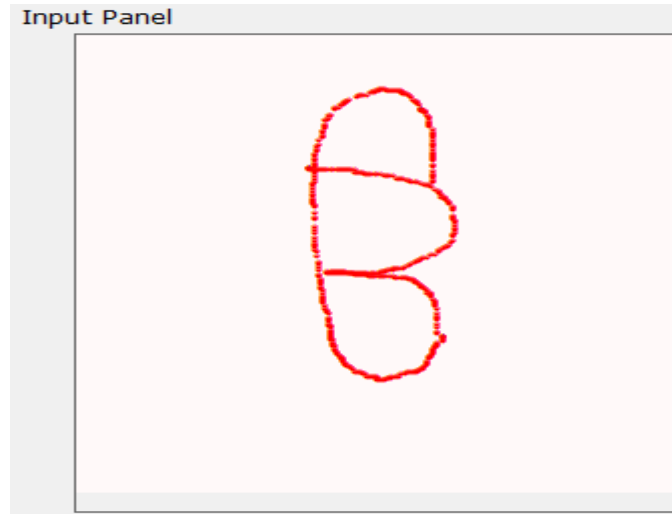
---

Collection of data, preprocessing and computation of features are the three stages needed before recognition stage in online handwriting recognition system. The following areas incorporate collection of data, pre-processing steps applied on input strokes and feature extraction. Section 3.1 focuses on information collection, Section 3.2 incorporates pre-processing and its different stages, and Section 3.3 talks about computation features. It might be noticed that these three handling phases of online handwriting recognition are not free with one another and should be arranged together. Since the execution of the strategies utilized as a part of these stages influence the general recognition rate.

#### 3.1 DATA COLLECTION PHASE

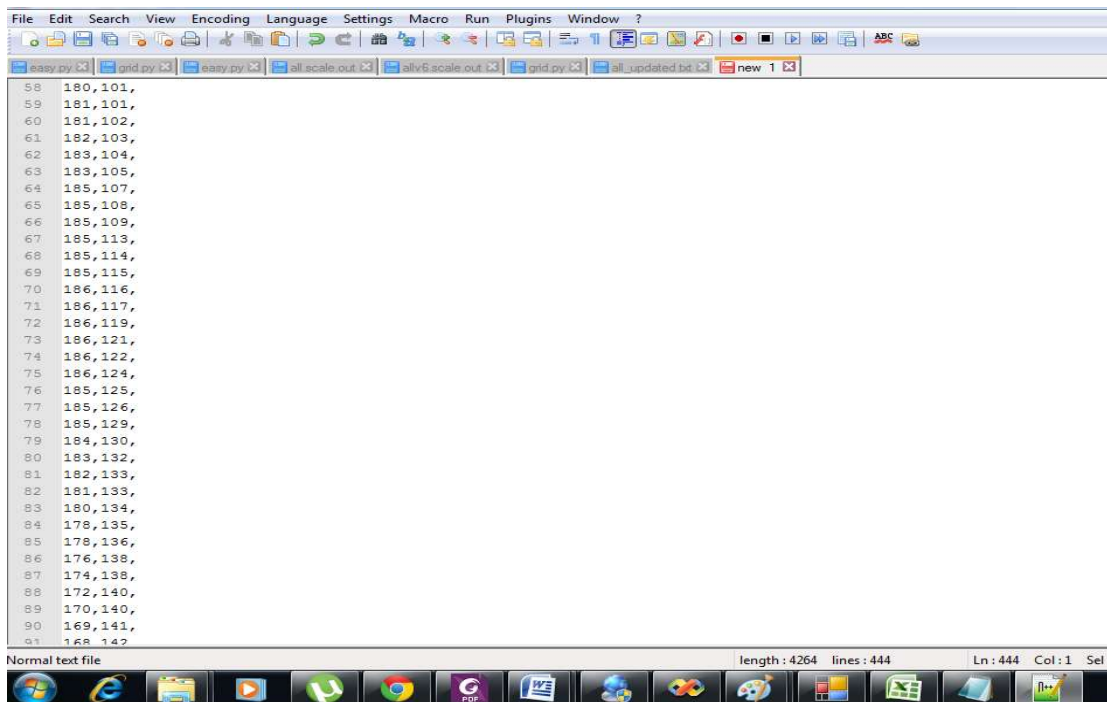
In data collection  $x, y$  coordinate points is taken in the motion. Associated parts of the pen follow, in which the pen point is touching the composing surface, are called strokes. The pen trace is normally examined with a steady rate also, along these lines information focuses are equitably circulated in time however not in space. One can take note of that the velocity of writing ordinarily backs off on sharp corners, in the first place of the stroke and toward the end of stroke. It additionally slows down if author is feeling delay in writing or taking an interruption.

In stroke catching stage, to catch the information points that speak to stroke patterns stylus or pen. An ordinary pen incorporates two activities, specifically, pen-down and pen-up. In our work the writing territory has a 300 x 300 pixels window size. The writer will write in the input panel shown in below Figure 3.1.



**Figure 3.1:** Input Area.

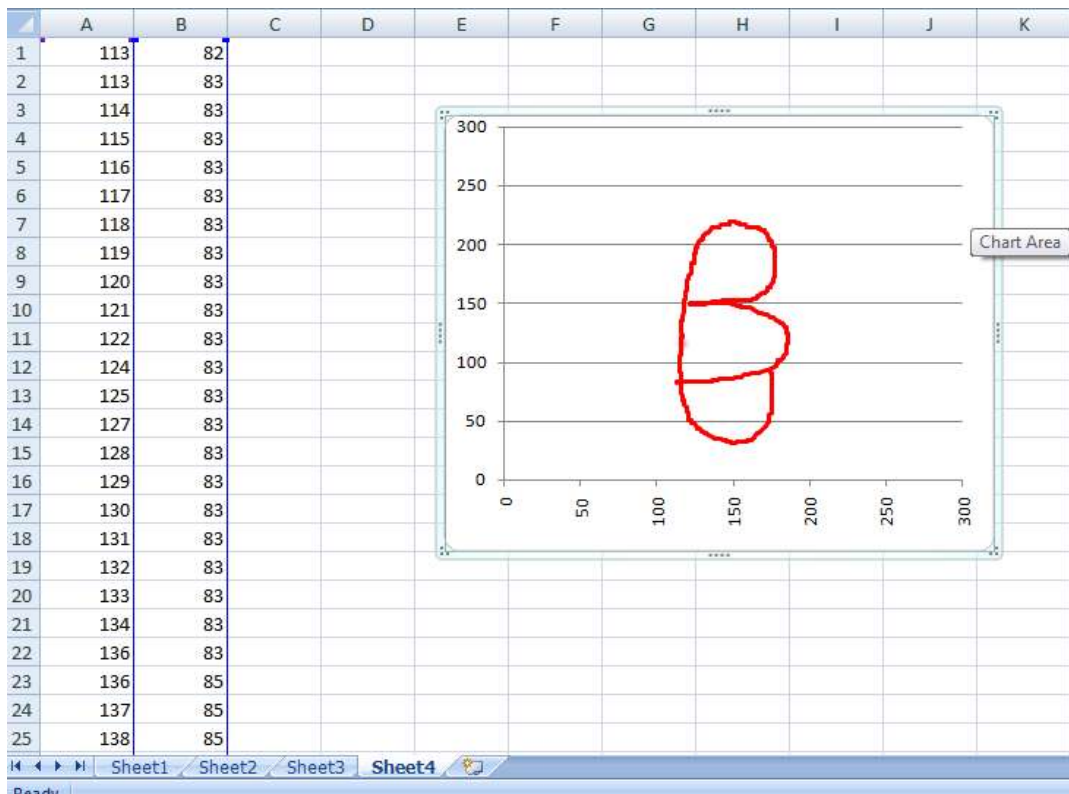
I have built up an online manually written Gurmukhi character recognizer which gives a Figureical user interface to show the gathered co-ordinate from pen movements.



**Figure 3.2:** Figure showing text file containing data of handwritten character shown in Figure 3.1.

This application gathers information pen movements and further stores these pen movements in a file in text format. Points record storage is obliged to hold original pen movements that are needed at later stages. These points are sent for preprocessing.

Figure 3.2 shows the screen shot of the data collected in text file. Figure 3.3 shows the shape of stored stroke in MS-Excel.

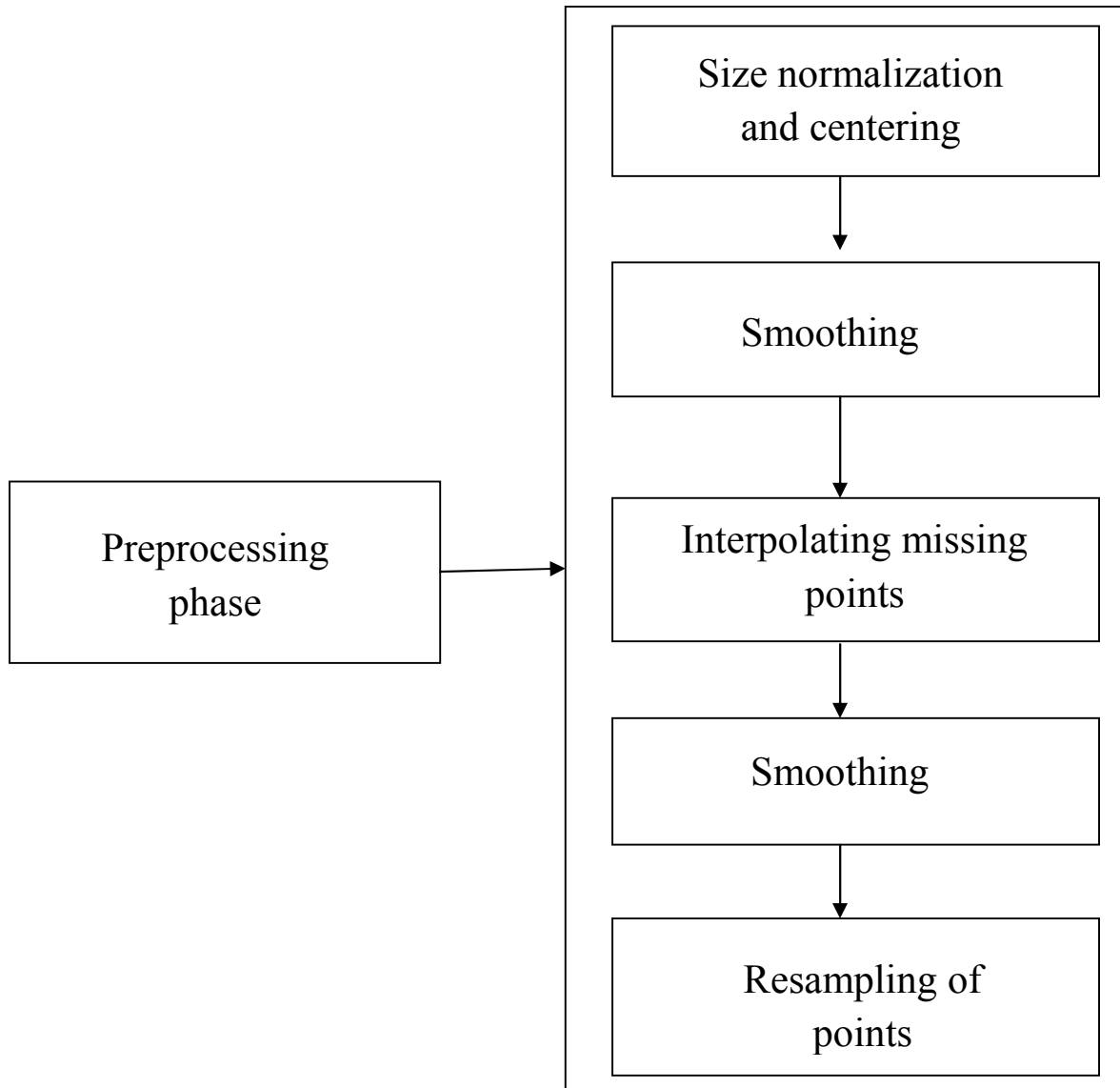


**Figure. 3.3:** Input handwritten Gurmukhi character in MS-Excel.

### 3.2 PREPROCESSING

Preprocessing is the very first and most essential stage in a OHWR system. The principle reason for preprocessing stage in OHWR system is to discard noise in data because of the limitations of hardware or software and change it into a smooth handwriting. The distortions of this type are incorporate diverse size of content, missing points in the stroke amid pen motion, jitter display in stroke and uneven separations of points from contiguous positions. In OHWR system, pre-processing incorporates five basic steps

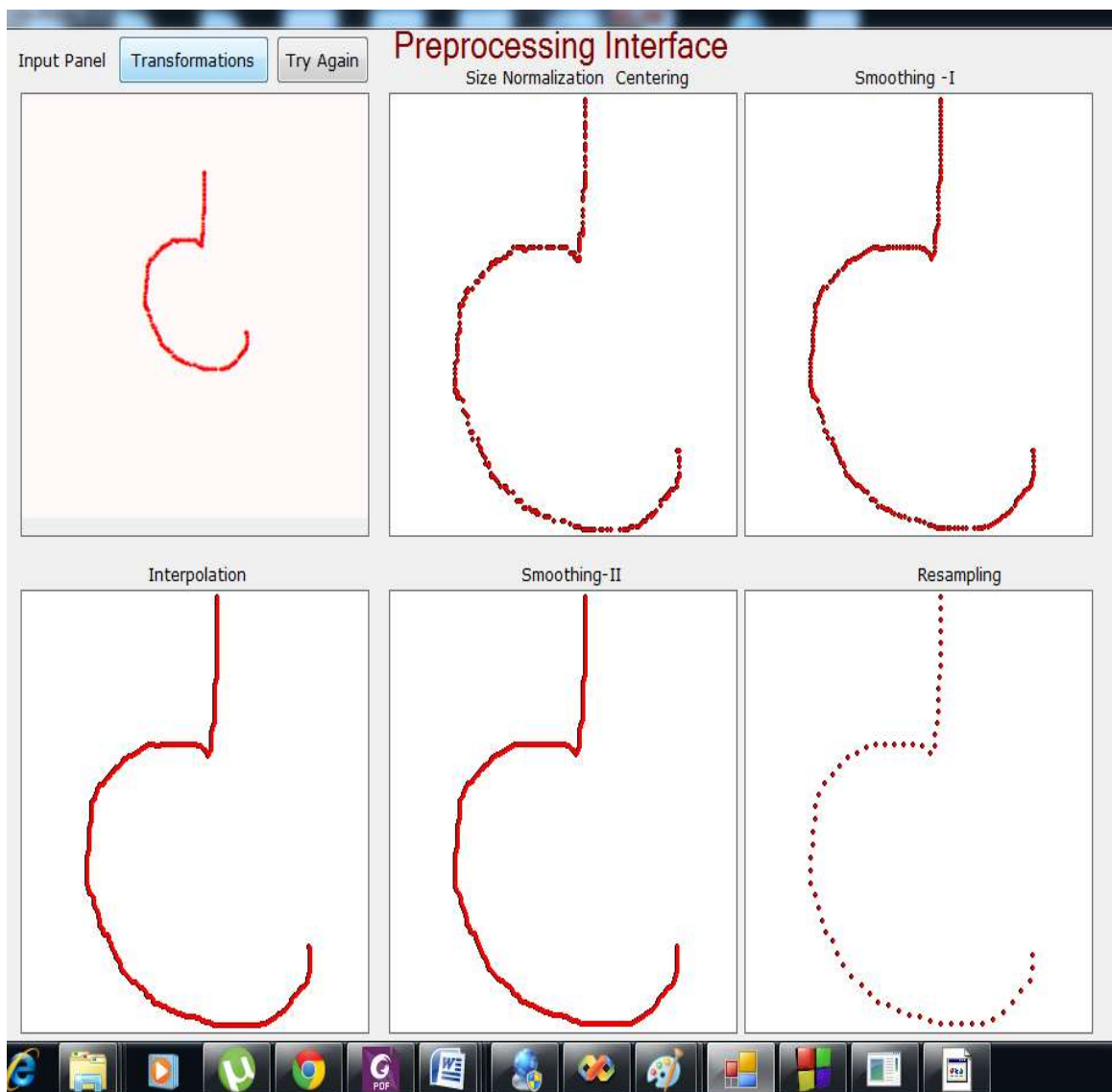
(Sharma, 2009). These steps of pre-processing that are followed in this report are demonstrated in Figure 3.4.



**Figure 3.4:** Steps in preprocessing phase

In the implementation of algorithms for pre-processing discussed, we have developed a pre-processing interface to take input from the writer shown in Figure 3.1. This interface is divided into six regions as shown in below Figure 3.5. In the very first box, named as input panel, where the user will write strokes. In the next region a size normalized and

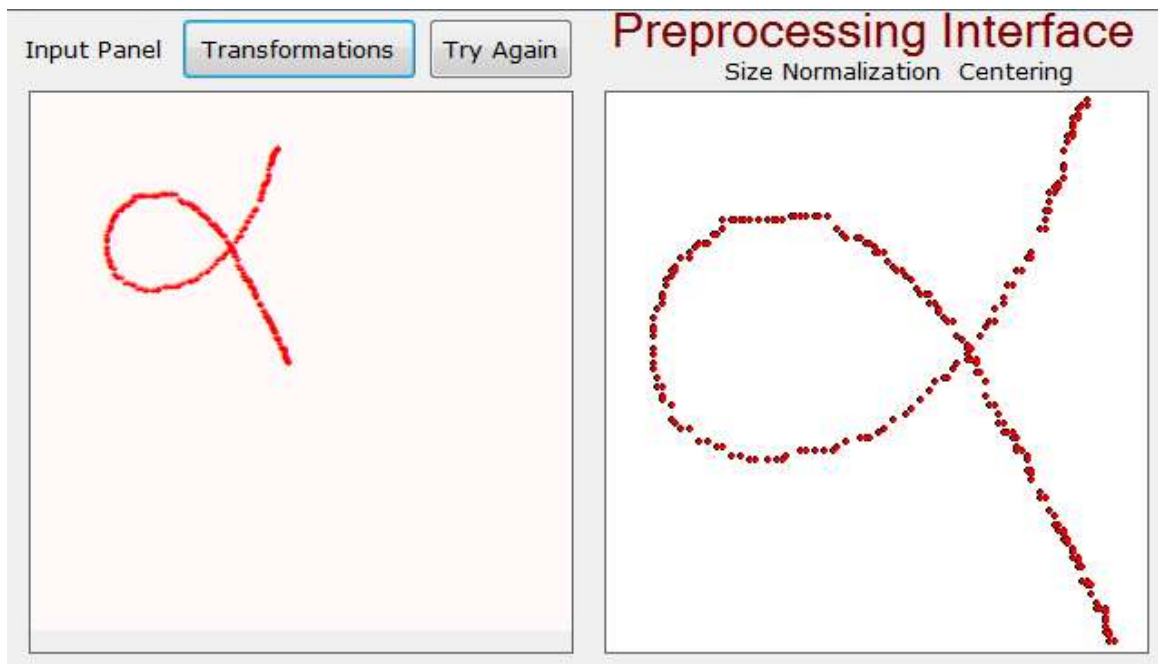
centered stroke is displayed and this region is given name as Normalization and centering. After this in the extreme right, region for displaying smoothing-I of stroke is present. I have used smoothing twice in the pre-processing stages because it is giving better stroke shape. Next window displays the stroke after applying interpolating missing point. In the right side of it, window displaying smoothing-II. The last window shows final output. i.e., resampling. The points after resampling window is given to the support vector machine for further recognition.



**Figure 3.5:** Interface showing all preprocessing stages.

### 3.2.1 Size Normalization And Centering of Stroke

It is basically relies on, how writer moves the pen on composing pad. When the user writes along the boundary of the composing surface then centering is required. When we move the pen along the boundry of the writing surface, normally the stroke is not centered. Sometimes this thing creates lots of problems. Size normalization and centering of stroke plays a vital role in request to recognize a stroke (Sharma, 2009).



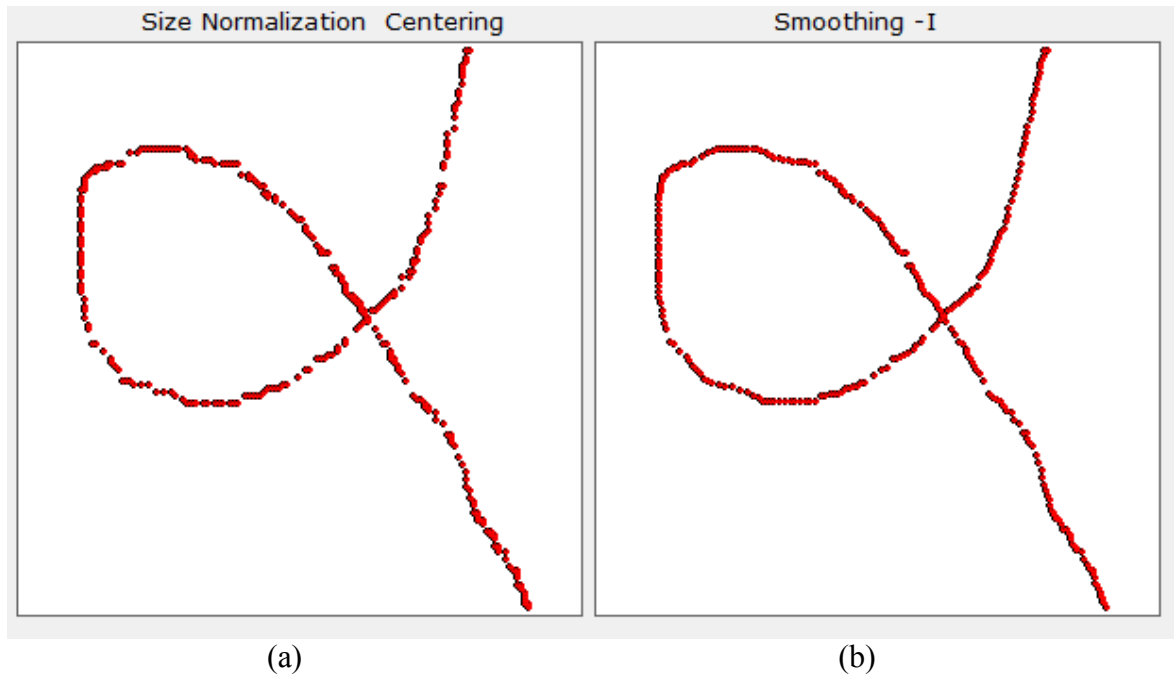
(a)

(b)

**Figure. 3.6(a):** Gurmukhi character of size smaller than 300×300 pixels. Figure 3.6(b): Transformation of Gurmukhi character (given in Figure 3.6(a)) into size normalization and centering.

### 3.2.2 Smoothing of Strokes

Because of individual penmanship styles and equipment constraints, there may be the chances of the jitters in the online handwriting. Smoothing is done keeping in mind to reduce these jitters from the manually written character.

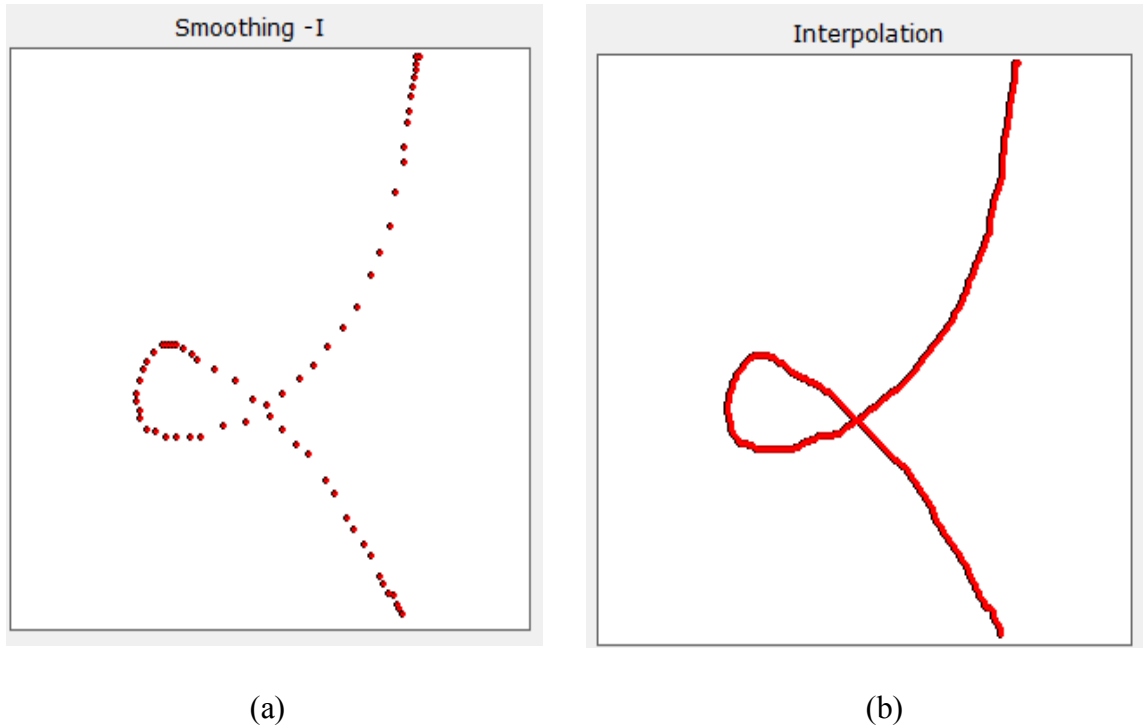


**Figure 3.7**

Figure 3.7(a): Gurmukhi character with size normalization and centering in 300×300 pixels. Figure 3.7(b): Transformation of Gurmukhi character (given in Figure 3.6(a)) after applying smoothing.

### 3.2.3 Interpolating Missing Points

The x, y coordinate collected in input panel, is purely relies on the writing velocity of writer. If we write fast, the number of points collected is less and if we write slowly, the number of points will be more. In this step of pre-processing, we put coordinates between two points if the coordinate is missing. To implement this approach, I have used Bezier curve. In Bezier Curve, an arrangement of four successive x, y coordinate is considered for having a Bezier. The following Figure 3.8 shows the pictorial representation of Bezier curve.

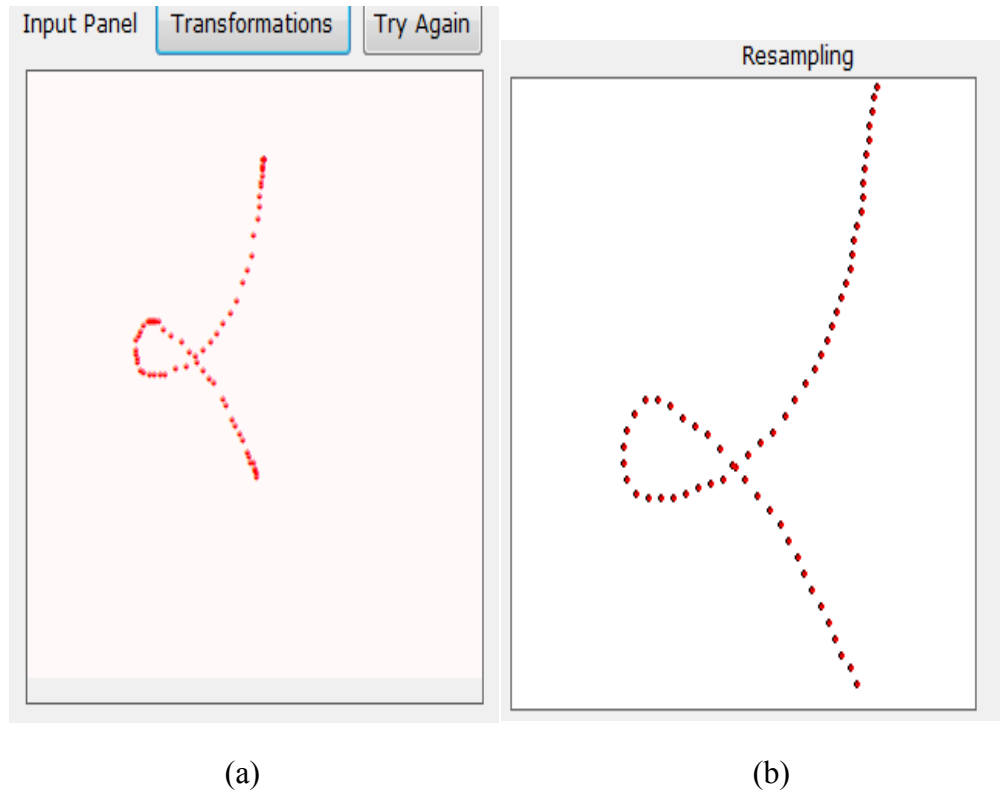


**Figure 3.8(b):** Showing interpolating missing points

When the separation is more than one between the points, Bezier interpolation is used. Insertion is finished between points whose separation is more than one. The Figure 3.8 comprises of two segments. The Figure 3.8(a) demonstrates the smoothing of input stroke.

### 3.2.4 Resampling of Stroke

Resampling of points is obliged to keep the points equivalent separations. For any pair of points having a separation more than one, we include a new point between such pairs. Any pair having separation under one is untouched. In my work, the quantity of x,y coordinates in resampling is settled to 64 points.



**Figure 3.9(a):** Gurmukhi character of size smaller than  $300 \times 300$  pixels. Figure 3.9(b): Transformation of Gurmukhi character (given in Figure 3.9(a)) into Resampling of stroke into 64 points.

The points are chosen in such a path that the shape of the character must be recovered. Meaning that we should apply resampling in such a way that the shape of the stroke must not be disturbed. The above Figure 3.9(a) represents the input stroke and Figure 3.9(b) shows resampling done on the input written character.

### 3.3 Feature Selection

Feature selection is one of the most important phase in OHWR system. Coordinates gathered after pre-processing are used as a feature for recognition. I have taken  $x, y$  coordinates as feature. The preprocessed coordinates are fixed for each and every stroke and are equally apart from each other as far as conceivable. In this work the pre-processed points in a stroke is fixed to 64.

## CHAPTER 4

---

---

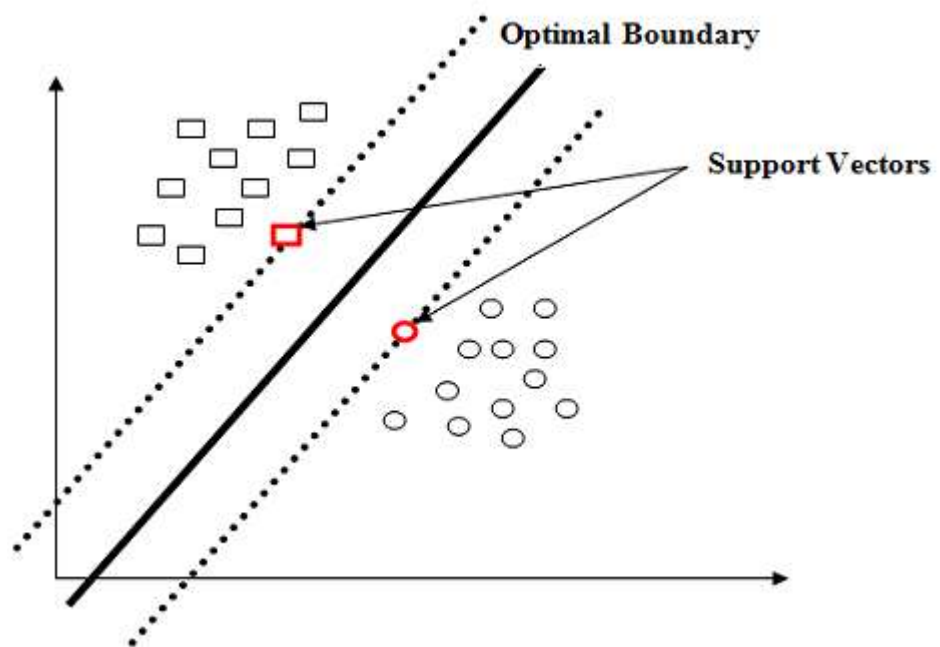
# RECOGNITION OF GURMUKHI CHARACTERS

---

---

### 4.1 Introduction to SVM

SVM as a classifier that differentiate directly between any of the two classes by discovering a hyperplane of most margin between them. This margin implies the minimal separation from the isolating hyperplane to the nearest data points in near by class. Support vector machine as a learning machines looks for an ideal isolating hyperplane where the margin is biggest. This idea is represented in Figure 4.1.



**Figure 4.1:** Idea behind support vectors.

#### 4.1.2 SVM Applications

Support vector machine classifier showed massive fame in recent years giving magnificent recognition accuracy. It brings about different applications. A portion of the applications of SVM at this moment are:





- Text arrangement
- Image classification
- Bioinformatics
- Topic float in page positioning algorithms
- OHWR system

## 4.2 Recognition of Gurmukhi Strokes

A Gurmukhi character can be written in one stroke or more than one stroke that depends on the user. Earlier, a character of Gurmukhi is divided into three zones, i.e., lower zone, center zone and upper zone, for recognition, stroke id is given to each conceivable stroke that can make a Gurmukhi character but we have not considering any zone concept. we are working only on single engine which consist of all the zone in single engine. IDs has been given to each stroke is demonstrated in Table 4.1. Symbol associated with the class Ids is also demonstrated.

With the assistance of electronic gadgets samples are captured, given in the beneath table. 100 patterns of each class is taken and the coordinates of each pattern stroke, i.e.,  $x$ ,  $y$  coordinates are saved in a file, in the txt format, called raw data, in a predetermined format which will be talked in the following Table 4.1.

**Table 4.1:** Showing all Gurmukhi strokes with IDs.

Serial.NO	CLASS ID	SHAPE	DESCRIPTON
1	101		Omkar
2	106		Dulinkar
3	121		Upper bar
4	122		Laawan

5	123	25-1	Dalwan
6	124	25-1	Hoorā
7	125	25-1	Ghanura bar
8	126	25-1	Aadhak
9	128	25-1	Tppi
10	133	25-1	Gnoura
11	134	25-1	Gnoura
12	141	25-1	Uraah
13	142	25-1	Uraah
14	143	25-1	Uraah
15	144	25-1	Aarh
16	145	25-1	Aarh
17	146	25-1	Eeddi
18	147	25-1	Eeddi
19	148	25-1	Eeddi
20	149	25-1	Eeddi
21	150	25-1	Eeddi
22	151	25-1	Sasa
23	152	25-1	Sasa
24	153	25-1	Sasa

25	154	ረ	Sasa
26	155	ረ	Haha
27	156	ረ	Haha
28	157	ጸ	Kakha
29	158	ጸ	Kakha
30	159	ረ	Kakha, Pappa
31	160	ረ	Kakha
32	161	ረ	Kakha, Pappa
33	162	—	Bar
34	163	—	Kakha, Dhdha
35	164	ረ	Gaga
36	165	ረ	Gaga
37	166	ጸ	Ghaga
38	167	ጸ	Ghaga
39	168	ጸ	Aeya
40	169	ጸ	Aeya
41	170	ረ	Chacha
42	171	ረ	Chacha
43	172	ረ	Chhacha

44	173	ᑭᑭᑭ	Chhacha
45	174	ᑭᑭᑭ	Jaja
46	175	ᑭᑭᑭ	Jaja
47	176	ᑭᑭᑭ	Jhaja
48	177	ᑭᑭᑭ	Jhaja
49	179	ᑭᑭᑭ	Neya, Vawa
50	180	ᑭᑭᑭ	Neya, Vawa
51	181	ᑭᑭᑭ	Vaawa
52	182	ᑭᑭᑭ	Tenka
53	183	ᑭᑭᑭ	Thatha
54	184	ᑭᑭᑭ	Thatha
55	185	ᑭᑭᑭ	Dadda
56	186	ᑭᑭᑭ	Dada
57	187	ᑭᑭᑭ	Dada
58	188	ᑭᑭᑭ	Dada
59	189	ᑭᑭᑭ	Nahnah
60	190	ᑭᑭᑭ	Nahnah
61	191	ᑭᑭᑭ	Tatta
62	192	ᑭᑭᑭ	Tatta

63	193	ଦାଦା	Dadha
64	194	ଦାଦା	Dadha
65	195	ଦାଦା	Dadha
66	196	ନାନ୍ହା	Nanha
67	197	ନାନ୍ହା	Nanha
68	198	ନାନ୍ହା	Nanha
69	200	ଫାଫା	Fafa
70	201	ଫାଫା	Fafa
71	202	ବାବା	Baba
72	203	ବାବ୍ବା	Babba
73	204	ଭାଭା	Bhabha
74	205	ଭାଭା	Bhabha
75	206	ମାମା	Mama
76	207	ଯାଯା	Yaya
77	208	ଯାଯା	Yaya
78	209	ଯାଯା	Yaya
79	210	ଲାଲା	Lala
80	211	ଲାଲା	Lala
81	212	ରାଆ	Rara

82	214		Rara
83	215		Cehri
84	216		Behri
85	217		Mama
86	218		Jhaja
87	220		Cehari
88	222		Jaja
89	223		Jaja
90	224		Hooda
91	225		Tanka
92	226		Chocha
93	351		One
94	399		Rupee Symbol

For recognition of a Gurmukhi character, initially preprocessed points having 64 resample points are taken, i.e., 64 x coordinates and 64 y coordinates makes total 128. These points are saved in a predetermined format of the SVM classifier. Here we have taken the resampled co-ordinates in SVM form. The format in which SVM takes the input is given 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.**

Thus, all the 100 samples of every class are pre-processed and the outcome is saved in the SVM predefined format shown in Figure 4.3. As depicts in the previous section SVM should be trained with an training data which gives a model file. Before training of machine, the preprocessed data must be scaled for higher accuracy. The pre-processed data is scaled in a lower and upper bound. After pre-processing of data, co-ordinate ranges from (0, 0) to (300, 300), these co-ordinates was scaled with the lower limit of 1 and highest of 9, using LIBSVM. This range of scaling is fixed with 1 to 9 due to experimental reason as we have getting higher accuracy in this region of scaling.



Figure 4.3: Pre-processed patterns in SVM format

For training the SVM machine, we need a scaled data because scaling can provide higher accuracy. The scaling should be possible with any lower and maximum limits as it relies purely upon the input data. In this report, to scale the raw data, lower limit is set to be 1 and upper is 9 taken. Although, we are getting almost same result on the limit of -5 to 5. This purely depends on the practical aspects. The scaled text file is demonstrated in Figure 4.4. The training data for preparing the SVM ought to be exact to get right recognition results, along these lines particular stroke-id of stroke is given, caught at the time of taking the input.



Figure 4.4: Scaled file of processed strokes.

## Parameter Selection

LibSVM is a straightforward, simple to-utilize, and productive tool for SVM classification and regression. It additionally gives an automatic model selection tool for

C-SVM arrangement. There are five types of SVM in Libsvm which is shown in below Table 4.2. These option values will be provided during the recognition phase and C-SVC is the default SVM type in LibSVM.

SVM TYPE	OPTION VALUES
C-SVC	0
NU-SVC	1
One-Class SVM	2
Epsilon-SVR	3
NU-SVR	4

**Table 4.2:** Types of SVM and their option value

KERNAL TYPE	OPTION VALUES
Linear	0
Polynomial	1
Radial Basis Function	2
Sigmoid	3
Pre-Computed Kernel	4

**Table 4.3:** Types of kernals and their option values

There are five types of kernels used in LibSVM and we have to provide their option values during the recognition phase. There is one default kernel in LibSVM named RBF (Radial Basis Function). The option values of these kernel and default values of the kernel is shown in the Table 4.3 . Apart from these parameters there are some others option values, which have to be provided at the time of recognition process. Those options and their description is shown in below Table 4.4. The effectiveness of SVM machine relies upon the Parameters utilized, i.e., the choice of SVM's kernel, kernel

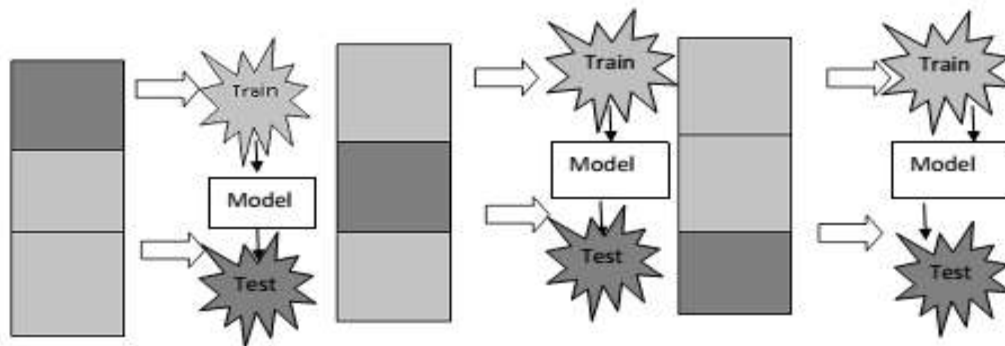
parameter and the penalty  $C$ . In libSVM there are five types of kernel but I have used RBF kernel because, normally, the RBF kernel is a sensible first decision.

<b>OPTION</b>	<b>NAME OF OPTION</b>	<b>DEFAULT VALUES</b>
-s	SVM type	0
-t	Kernel type	2
-d	Degree	3
-g	Gamma	1/Num Features
-r	Coef0	0
-c	Cost	1
-n	NU	0.5
-P	Epsilon	0.1
-m	Cache Size	100
-e	Epsilon	0.001
-h	Shrinking	1
-b	Probability Estimates	0
-wi	Weight	1
-v n	n-Fold Cross	NA
-q	Quiet Mode	NA

**Table 4.4:** Option available in LIBSVM and their default values

### 4.2.1 Training and Testing Using $K$ -Fold Cross-Validation

In  $k$ -fold cross-validation, we first separate the training set into  $k$  subsets of equivalent size. Successively one subset is tested utilizing the classifier trained on the remaining  $k-1$  subsets. Along these lines, every case of the entire training set is anticipated once so the cross-validation accuracy is the rate of data which are effectively classified. The main advantage of cross-validation scheme is that it prevents the over-fitting problem. Working of 3-fold cross-validation is shown in the Figure 4.5. There are two parameters, which I have used, for a RBF kernel penalty factor ( $C$ ) and gamma ( $\gamma$ ). It is not known previously, which  $C$  and  $\gamma$  are best for a given issue, therefore some sort of model determination (parameter look) must be finished.



**Figure 4.5:** Working of 3-fold cross-validation

The objective is to find  $C$  and  $\gamma$ , that the classifier can precisely anticipate unknown data (i.e. testing data). I have done a "grid-search" on  $C$  and  $\gamma$  utilizing cross-acceptance. Different sets of ( $C$ ,  $\gamma$ ) values are attempted and the one with the best cross-validation accuracy is picked. We found that attempting exponentially growing successions of  $C$  and  $\gamma$  is a practical system to recognize best parameters.

```

File Edit Format View Help
svm_type c_svc
kernel_type rbf
gamma 0.00195313
nr_class 95
total_sv 4621
rho 0.0143446 0.205645 4.9916 -0.579007 0.195244 -0.451772 -0.159565 -0.493791 -0.170736 -0.485659 -0.0795236 -0.339344 -0.0785813 -0.508374 -0.351902 0.82255 0.0229055
0.403808 -0.277202 -0.448016 -0.180007 -0.616729 -0.388661 0.211528 -0.331438 -0.482413 -0.39669 -0.0963396 -0.378213 -0.343145 0.0125304 -0.326936 -0.265167 -0.244171
2713 -0.180433 -0.0377704 -0.274081 -0.124227 0.615027 -0.107324 -0.538033 -0.678499 -0.493082 -0.260815 -0.112649 0.467826 -0.460176 -0.275476 -0.0241862 0.00710524 0.
578218 -0.37521 0.199948 -0.120002 0.192372 -0.0574652 0.249526 -0.338706 -0.218971 -0.229943 -0.817933 -0.869531 -0.473879 0.200154 0.06797 -0.473032 0.168508 0.35383
228375 -0.0583877 0.233602 -1.86853 -0.0600907 0.186147 -0.0293925 0.0333757 0.279239 0.450664 0.346843 0.160494 0.542096 0.0377233 0.414944 0.17418 0.561596 0.0126688
0.42727 -0.267308 -0.413757 -0.0561908 0.0496021 -0.219124 -0.443815 -0.299327 -0.343293 -0.500732 -0.340287 -0.612098 -0.366869 -0.284954 -0.262829 -0.420539 -0.612966
0.0615331 0.403562 -0.121445 -0.0713977 -0.397473 -0.308725 -0.24215 0.0354503 -0.299004 -0.125046 -0.27051 -0.115199 -0.383309 0.0611358 0.237053 -0.284692 0.392701 -0.
873 0.399707 0.240756 0.230852 0.426797 0.198126 0.312687 0.34522 0.248725 0.237978 -0.0436735 0.29534 0.178015 0.270593 0.223441 0.124211 0.231412 0.305072 0.392319 0.
369062 -0.205643 -0.325013 -0.0644431 -0.069646 0.078162 0.0467036 -0.0252743 -0.216736 -0.28785 -0.0807052 -0.384239 -0.373026 -0.126594 -0.273602 -0.180017 -0.169122
0.435878 0.0627539 0.198041 0.169921 0.763678 0.529341 0.329481 0.065929 0.35768 0.139551 -0.0273208 0.336791 0.369282 0.12906 0.240507 0.462001 0.0791366 -0.147842 -0.
89793 -0.0288228 -0.488598 -0.562732 0.0846374 0.0635874 0.100076 -0.131313 0.31953 0.527127 -0.0536194 -0.336711 -0.107287 0.281148 0.00118662 -0.00541151 0.304046 -0.
1209 -0.157449 -0.0823504 -0.670333 -0.871436 -0.251024 -0.0652871 0.141731 0.286183 -0.0260045 -0.0303012 -0.133295 -0.080894 -0.118278 -0.316313 -0.135267 -0.0627343 -
45 -0.193619 0.0849884 -0.148524 -0.0881187 0.148163 0.42403 0.236688 0.0629251 0.404449 0.0105491 0.239172 0.0921048 0.61309 -0.117435 0.22024 0.247933 0.383398 0.6167
0.26853 -0.103607 -0.585619 -0.386716 -4.386624 -0.419889 0.0388952 0.507531 -0.336255 -0.348843 0.652495 1.10421 0.213598 -0.0970226 0.0518547 0.57929 0.420539 -0.194716
0.0318011 -0.201195 -0.220959 -0.321932 -0.231677 -0.248022 -0.401893 -0.339001 -0.384305 -0.146022 -0.0967634 -0.166036 -0.124314 -0.193224 -0.377512 -0.442036 -0.18188
1 -0.625101 -0.320221 -0.117434 -0.177668 -0.499483 -0.384059 -0.424218 -0.346025 -0.0255116 -0.295581 -0.536858 -0.422667 -0.438272 -0.630451 -0.778931 -0.306971 -0.50
898 0.345518 0.509196 -0.157186 0.185955 0.386424 0.194992 0.188321 -0.181849 0.342052 0.976046 0.312732 0.172395 0.0564715 0.155016 0.26367 0.178892 0.668995 0.50717
0.049322 0.121531 0.497556 0.240891 0.153795 0.61758 0.12687 0.231948 0.224567 0.633478 0.0398983 0.193987 0.869442 0.447652 0.58825 0.375987 0.0373378 0.210246 0.141645
91 -0.0156909 0.347697 -0.122815 0.0210392 1.0576 0.136392 -0.415963 -0.436669 0.0661808 -0.0678403 -0.00582366 0.105843 -0.517086 -0.262153 0.436837 0.205631 0.28092 0.
38064 -0.132948 0.0909622 0.413144 0.137336 0.0349249 0.373135 -0.0805272 0.183143 0.103392 0.359761 -0.123098 0.133839 0.0598519 0.464563 0.462277 0.216965 -0.0627559
332 0.271517 0.0860239 0.266159 -0.12745 0.195219 0.153339 0.160038 0.107672 -0.015725 -0.106706 0.317587 0.452067 0.77179 0.317539 0.286236 0.432316 0.227888 0.348953
0.223057 -0.0587516 0.677077 -0.216162 0.113791 -0.207455 0.255063 0.612008 0.0631282 0.359175 -0.00564169 -0.485633 -0.567438 0.244184 0.187535 -0.584841 -0.0190861 0.
2325 -0.0201209 -0.061215 -0.191725 -0.0413313 0.243943 -0.0702821 -0.192787 -0.0333564 0.086504 0.255958 0.205102 -0.0786243 -0.0975061 -0.167359 -0.051838 -0.268826
166181 0.451963 0.124303 -0.289965 -0.382606 -0.351042 -0.165147 -0.0252732 0.89632 -0.436828 0.138664 0.53878 0.241913 0.437098 0.43818 0.30604 0.934147 1.38799 0.6259
0.311184 -0.201508 -0.0628465 0.0307042 0.534806 0.581728 0.410785 -0.203221 -0.025024 -0.180928 -0.301005 0.263779 0.226916 -0.374907 -0.00709164 0.360085 -0.231978 -0.
0.0279589 -0.182096 -0.264118 0.186351 -0.102697 -0.115151 0.031846 0.39387 0.00759172 0.0679882 0.320298 -0.0813838 0.187287 0.112779 0.313618 -0.146771 0.157174 -0.092
5.20617 -0.0526592 0.209429 0.030271 0.548252 0.723261 0.371868 0.986551 0.394402 0.0243128 -0.0305973 0.479151 -1.37302 -0.0838342 0.343887 -0.0118729 0.200652 0.49086
166181 0.451963 0.124303 -0.289965 -0.382606 -0.351042 -0.165147 -0.0252732 0.89632 -0.436828 0.138664 0.53878 0.241913 0.437098 0.43818 0.30604 0.934147 1.38799 0.6259
0.311184 -0.201508 -0.0628465 0.0307042 0.534806 0.581728 0.410785 -0.203221 -0.025024 -0.180928 -0.301005 0.263779 0.226916 -0.374907 -0.00709164 0.360085 -0.231978 -0.
0.0279589 -0.182096 -0.264118 0.186351 -0.102697 -0.115151 0.031846 0.39387 0.00759172 0.0679882 0.320298 -0.0813838 0.187287 0.112779 0.313618 -0.146771 0.157174 -0.092
5.20617 -0.0526592 0.209429 0.030271 0.548252 0.723261 0.371868 0.986551 0.394402 0.0243128 -0.0305973 0.479151 -1.37302 -0.0838342 0.343887 -0.0118729 0.200652 0.49086
166181 0.451963 0.124303 -0.289965 -0.382606 -0.351042 -0.165147 -0.0252732 0.89632 -0.436828 0.138664 0.53878 0.241913 0.437098 0.43818 0.30604 0.934147 1.38799 0.6259
0.311184 -0.201508 -0.0628465 0.0307042 0.534806 0.581728 0.410785 -0.203221 -0.025024 -0.180928 -0.301005 0.263779 0.226916 -0.374907 -0.00709164 0.360085 -0.231978 -0.
0.0279589 -0.182096 -0.264118 0.186351 -0.102697 -0.115151 0.031846 0.39387 0.00759172 0.0679882 0.320298 -0.0813838 0.187287 0.112779 0.313618 -0.146771 0.157174 -0.092
5.20617 -0.0526592 0.209429 0.030271 0.548252 0.723261 0.371868 0.986551 0.394402 0.0243128 -0.0305973 0.479151 -1.37302 -0.0838342 0.343887 -0.0118729 0.200652 0.49086
166181 0.451963 0.124303 -0.289965 -0.382606 -0.351042 -0.165147 -0.0252732 0.89632 -0.436828 0.138664 0.53878 0.241913 0.437098 0.43818 0.30604 0.934147 1.38799 0.6259
0.311184 -0.201508 -0.0628465 0.0307042 0.534806 0.581728 0.410785 -0.203221 -0.025024 -0.180928 -0.301005 0.263779 0.226916 -0.374907 -0.00709164 0.360085 -0.231978 -0.
0.0279589 -0.182096 -0.264118 0.186351 -0.102697 -0.115151 0.031846 0.39387 0.00759172 0.0679882 0.320298 -0.0813838 0.187287 0.112779 0.313618 -0.146771 0.157174 -0.092
5.20617 -0.0526592 0.209429 0.030271 0.548252 0.723261 0.371868 0.986551 0.394402 0.0243128 -0.0305973 0.479151 -1.37302 -0.0838342 0.343887 -0.0118729 0.200652 0.49086
166181 0.451963 0.124303 -0.289965 -0.382606 -0.351042 -0.165147 -0.0252732 0.89632 -0.436828 0.138664 0.53878 0.241913 0.437098 0.43818 0.30604 0.934147 1.38799 0.6259
0.311184 -0.201508 -0.0628465 0.0307042 0.534806 0.581728 0.410785 -0.203221 -0.025024 -0.180928 -0.301005 0.263779 0.226916 -0.374907 -0.00709164 0.360085 -0.231978 -0.
0.0279589 -0.182096 -0.264118 0.186351 -0.102697 -0.115151 0.031846 0.39387 0.00759172 0.0679882 0.320298 -0.0813838 0.187287 0.112779 0.313618 -0.146771 0.157174 -0.092
5.20617 -0.0526592 0.209429 0.030271 0.548252 0.723261 0.371868 0.986551 0.394402 0.0243128 -0.0305973 0.479151 -1.37302 -0.0838342 0.343887 -0.0118729 0.200652 0.49086
166181 0.451963 0.124303 -0.289965 -0.382606 -0.351042 -0.165147 -0.0252732 0.89632 -0.436828 0.138664 0.53878 0.241913 0.437098 0.43818 0.30604 0.934147 1.38799 0.6259
0.311184 -0.201508 -0.0628465 0.0307042 0.534806 0.581728 0.410785 -0.203221 -0.025024 -0.180928 -0.301005 0.263779 0.226916 -0.374907 -0.00709164 0.360085 -0.231978 -0.
0.0279589 -0.182096 -0.264118 0.186351 -0.102697 -0.115151 0.031846 0.39387 0.00759172 0.0679882 0.320298 -0.0813838 0.187287 0.112779 0.313618 -0.146771 0.157174 -0.092
5.20617 -0.0526592 0.209429 0.030271 0.548252 0.723261 0.371868 0.986551 0.394402 0.0243128 -0.0305973 0.479151 -1.37302 -0.0838342 0.343887 -0.0118729 0.200652 0.49086
166181 0.451963 0.124303 -0.289965 -0.382606 -0.351042 -0.165147 -0.0252732 0.89632 -0.436828 0.138664 0.53878 0.241913 0.437098 0.43818 0.30604 0.934147 1.38799 0.6259
0.311184 -0.201508 -0.0628465 0.0307042 0.534806 0.581728 0.410785 -0.203221 -0.025024 -0.180928 -0.301005 0.263779 0.226916 -0.374907 -0.00709164 0.360085 -0.231978 -0.
0.0279589 -0.182096 -0.264118 0.186351 -0.102697 -0.115151 0.031846 0.39387 0.00759172 0.0679882 0.320298 -0.0813838 0.187287 0.112779 0.313618 -0.146771 0.157174 -0.092
5.20617 -0.0526592 0.209429 0.030271 0.548252 0.723261 0.371868 0.986551 0.394402 0.0243128 -0.0305973 0.479151 -1.37302 -0.0838342 0.343887 -0.0118729 0.200652 0.49086
166181 0.451963 0.124303 -0.289965 -0.382606 -0.351042 -0.165147 -0.0252732 0.89632 -0.436828 0.138664 0.53878 0.241913 0.437098 0.43818 0.30604 0.934147 1.38799 0.6259
0.311184 -0.201508 -0.0628465 0.0307042 0.534806 0.581728 0.410785 -0.203221 -0.025024 -0.180928 -0.301005 0.263779 0.226916 -0.374907 -0.00709164 0.360085 -0.231978 -0.
0.0279589 -0.182096 -0.264118 0.186351 -0.102697 -0.115151 0.031846 0.39387 0.00759172 0.0679882 0.320298 -0.0813838 0.187287 0.112779 0.313618 -0.146771 0.157174 -0.092
5.20617 -0.0526592 0.209429 0.030271 0.548252 0.723261 0.371868 0.986551 0.394402 0.0243128 -0.0305973 0.479151 -1.37302 -0.0838342 0.343887 -0.0118729 0.200652 0.49086
166181 0.451963 0.124303 -0.289965 -0.382606 -0.351042 -0.165147 -0.0252732 0.89632 -0.436828 0.138664 0.53878 0.241913 0.437098 0.43818 0.30604 0.934147 1.38799 0.6259
0.311184 -0.201508 -0.0628465 0.0307042 0.534806 0.581728 0.410785 -0.203221 -0.025024 -0.180928 -0.301005 0.263779 0.226916 -0.374907 -0.00709164 0.360085 -0.231978 -0.
0.0279589 -0.182096 -0.264118 0.186351 -0.102697 -0.115151 0.031846 0.39387 0.00759172 0.0679882 0.320298 -0.0813838 0.187287 0.112779 0.313618 -0.146771 0.157174 -0.092
5.20617 -0.0526592 0.209429 0.030271 0.548252 0.723261 0.371868 0.986551 0.394402 0.0243128 -0.0305973 0.479151 -1.37302 -0.0838342 0.343887 -0.0118729 0.200652 0.49086
166181 0.451963 0.124303 -0.289965 -0.382606 -0.351042 -0.165147 -0.0252732 0.89632 -0.436828 0.138664 0.53878 0.241913 0.437098 0.43818 0.30604 0.934147 1.38799 0.6259
0.311184 -0.201508 -0.0628465 0.0307042 0.534806 0.581728 0.410785 -0.203221 -0.025024 -0.180928 -0.301005 0.263779 0.226916 -0.374907 -0.00709164 0.360085 -0.231978 -0.
0.0279589 -0.182096 -0.264118 0.186351 -0.102697 -0.115151 0.031846 0.39387 0.00759172 0.0679882 0.320298 -0.0813838 0.187287 0.112779 0.313618 -0.146771 0.157174 -0.092
5.20617 -0.0526592 0.209429 0.030271 0.548252 0.723261 0.371868 0.986551 0.394402 0.0243128 -0.0305973 0.479151 -1.37302 -0.0838342 0.343887 -0.0118729 0.200652 0.49086
166181 0.451963 0.124303 -0.289965 -0.382606 -0.351042 -0.165147 -0.0252732 0.89632 -0.436828 0.138664 0.53878 0.241913 0.437098 0.43818 0.30604 0.934147 1.38799 0.6259
0.311184 -0.201508 -0.0628465 0.0307042 0.534806 0.581728 0.410785 -0.203221 -0.025024 -0.180928 -0.301005 0.263779 0.226916 -0.374907 -0.00709164 0.360085 -0.231978 -0.
0.0279589 -0.182096 -0.264118 0.186351 -0.102697 -0.115151 0.031846 0.39387 0.00759172 0.0679882 0.320298 -0.0813838 0.187287 0.112779 0.313618 -0.146771 0.157174 -0.092
5.20617 -0.0526592 0.209429 0.030271 0.548252 0.723261 0.371868 0.986551 0.394402 0.0243128 -0.0305973 0.479151 -1.37302 -0.0838342 0.343887 -0.0118729 0.200652 0.49086
166181 0.451963 0.124303 -0.289965 -0.382606 -0.351042 -0.165147 -0.0252732 0.89632 -0.436828 0.138664 0.53878 0.241913 0.437098 0.43818 0.30604 0.934147 1.38799 0.6259
0.311184 -0.201508 -0.0628465 0.0307042 0.534806 0.581728 0.410785 -0.203221 -0.025024 -0.180928 -0.301005 0.263779 0.226916 -0.374907 -0.00709164 0.360085 -0.231978 -0.
0.0279589 -0.182096 -0.264118 0.186351 -0.102697 -0.115151 0.031846 0.39387 0.00759172 0.0679882 0.320298 -0.0813838 0.187287 0.112779 0.313618 -0.146771 0.157174 -0.092
5.20617 -0.0526592 0.209429 0.030271 0.548252 0.723261 0.371868 0.986551 0.394402 0.0243128 -0.0305973 0.479151 -1.37302 -0.0838342 0.343887 -0.0118729 0.200652 0.49086
166181 0.451963 0.124303 -0.289965 -0.382606 -0.351042 -0.165147 -0.0252732 0.89632 -0.436828 0.138664 0.53878 0.241913 0.437098 0.43818 0.30604 0.934147 1.38799 0.6259
0.311184 -0.201508 -0.0628465 0.0307042 0.534806 0.581728 0.410785 -0.203221 -0.025024 -0.180928 -0.301005 0.263779 0.226916 -0.374907 -0.00709164 0.360085 -0.231978 -0.
0.0279589 -0.182096 -0.264118 0.186351 -0.102697 -0.115151 0.031846 0.39387 0.00759172 0.0679882 0.320298 -0.0813838 0.187287 0.112779 0.313618 -0.146771 0.157174 -0.092
5.20617 -0.0526592 0.209429 0.030271 0.548252 0.723261 0.371868 0.986551 0.394402 0.0243128 -0.0305973 0.479151 -1.37302 -0.0838342 0.343887 -0.0118729 0.200652 0.49086
166181 0.451963 0.124303 -0.289965 -0.382606 -0.351042 -0.165147 -0.0252732 0.89632 -0.436828 0.138664 0.53878 0.241913 0.437098 0.43818 0.30604 0.934147 1.38799 0.6259
0.311184 -0.2015
```

## CHAPTER 5

---

### RESULTS AND DISCUSSIONS

---

---

As, discussed in the last Chapter 3, pre-processing have been applied on the raw data which was caught through an electronic gadget composed by distinctive writers. The coordinates of the patterns were caught and saved in a text file. On these received patterns pre-processing algorithms were applied and stored in a file with SVM format which is further forwarded to SVM for recognition. I have divided this datasets into three datasets of 30, 50 and 100 sample for every class. Dataset of 30 samples contains 30 samples of every class. Comparable work was accomplished for dataset of 50 and 100.

In k-fold cross-validation, we first separate the training set into  $k$  subsets of equivalent size. Successively one subset is tested utilizing the classifier trained on the remaining  $k-1$  subsets. Along these lines, every case of the entire training set is anticipated once so the cross-validation accuracy is the rate of data which are effectively classified. The main advantage of cross-validation scheme is that it prevents the over-fitting problem.

In 4-fold testing the data set of 30 examples is partitioned into 4 equivalent datasets out of which 3 will be utilized for training the SVM and 1 will be utilized for testing and at that point accuracy for distinctive parameters of SVM, gamma ( $\gamma$ ) and penalty factor ( $c$ ) value is perceived. Each isolated dataset was tried and normal accuracy was gained. A comparative procedure was taken after for remaining fold testing.

Earlier, the parameter of SVM is fixed manually and we have to try lots of combinations of gamma ( $\gamma$ ) and penalty factor ( $c$ ). I have used this approach many times but this is not fruitful and very time consuming. So, to find the best parameters, I have used a program based on python script language. Python scripting language find the best parameters for RBF kernel where the accuracy is maximum. The range of gamma ( $\gamma$ ) is set from  $2^3$  to  $2^{-15}$  and for penalty factor ( $c$ ) is set from  $2^{-5}$  to  $2^{15}$ . We have tried 110 combinations of penalty factor ( $c$ ) and gamma ( $\gamma$ ). The parameter epsilon ( $e$ ) is set to as its default value .001.

Data was pre-processed utilizing algorithms proposed in Sharma *et al.* (2009) and proposed in this report. Testing of 3 schemes of data was performed using svm and their result is shown further using Figure. In these Figures the  $x$ -axis speaks to penalty factor ( $c$ ) from  $2^{-5}$  to  $2^{15}$ , the  $y$ -axis speaks to the gamma ( $\gamma$ ) ranging from  $2^3$  to  $2^{-15}$ .

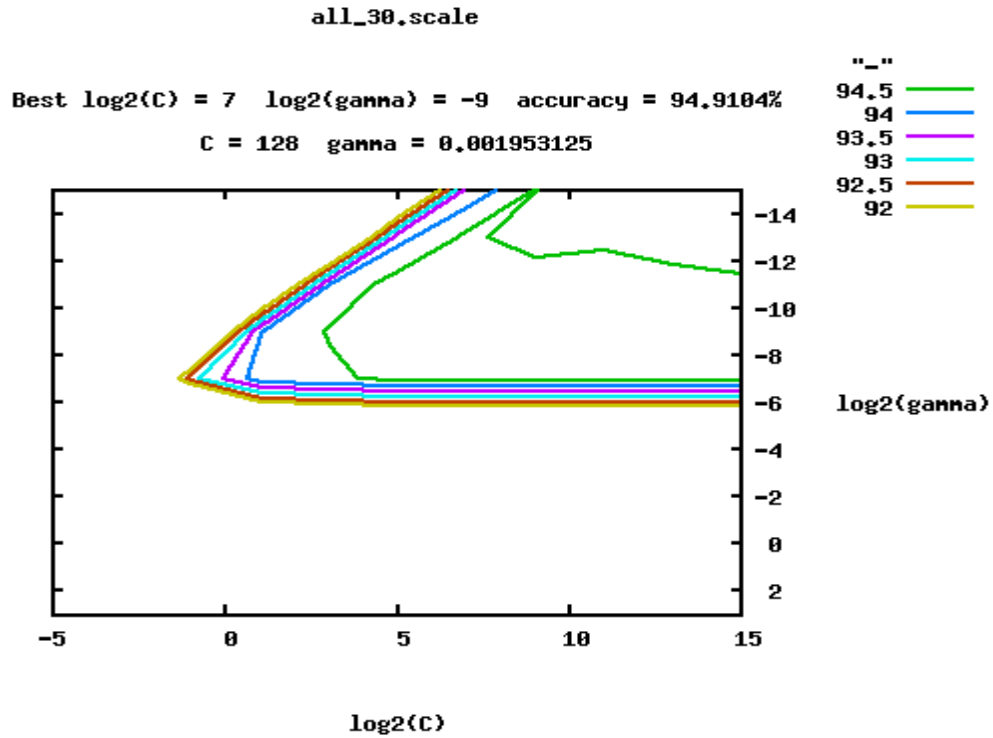
### 5.1 SCHEME 1: $K$ -Fold Cross-Validation of Datasets of 30 Samples Per Class

There are 95 classes in our datasets. 30 samples of each class have been used for cross-validation. Maximum accuracy of 2-fold, 3-fold, 4-fold, 5-fold, 6-fold cross-validation is given in the Table 5.1.

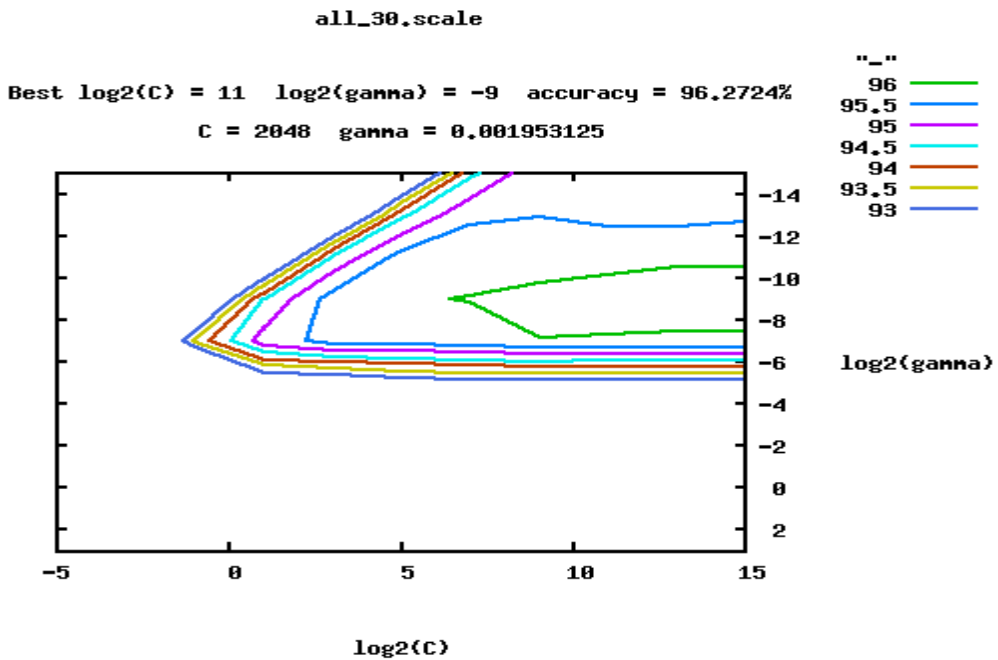
**Table 5.1:** Accuracy of Gurmukhi strokes for 2-fold, 3-fold, 4-fold, 5-fold and 6-fold cross-validation for 30 samples per class.

	<b>Accuracy (in %)</b>	<b>Gamma(<math>\gamma</math>)</b>	<b>Penalty factor(<math>C</math>)</b>
<b>2-fold</b>	94.91	.001953125	128
<b>3-fold</b>	96.27	.001953125	2048
<b>4-fold</b>	96.34	.001953125	8192
<b>5-fold</b>	96.59	.001953125	8192
<b>6-fold</b>	96.30	.001953125	8192

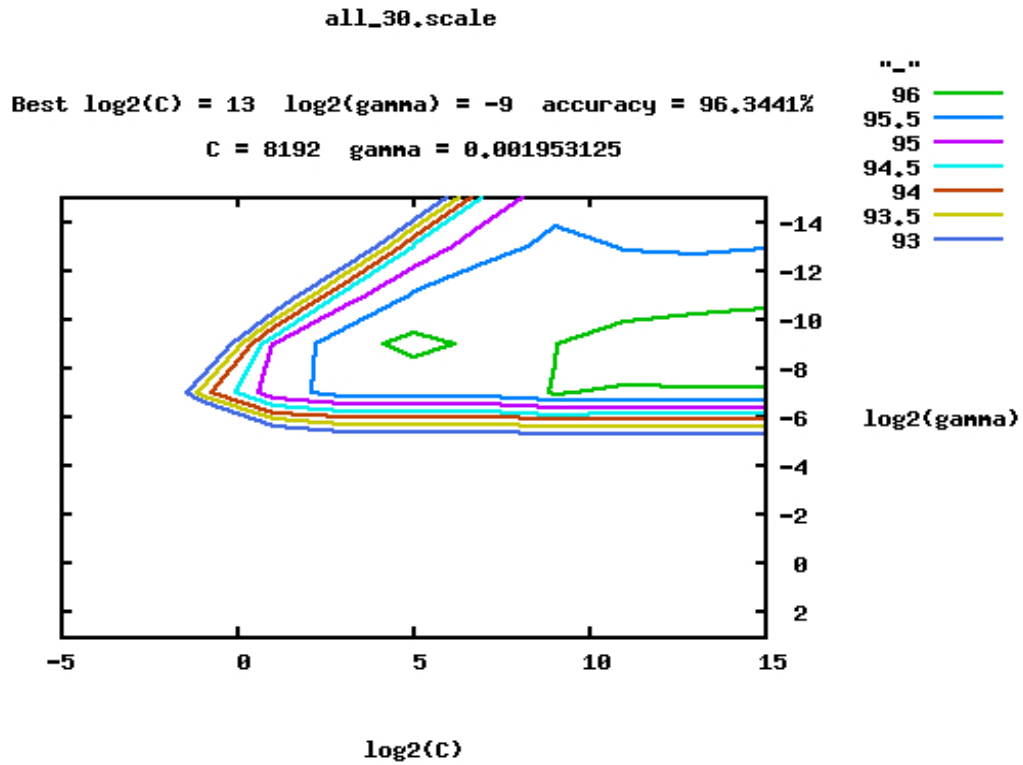
Figure 5.1 to Figure 5.5 shows all the accuracy achieved on 2-fold, 3-fold, 4-fold, 5-fold and 6-fold cross-validation for 30 samples per class.



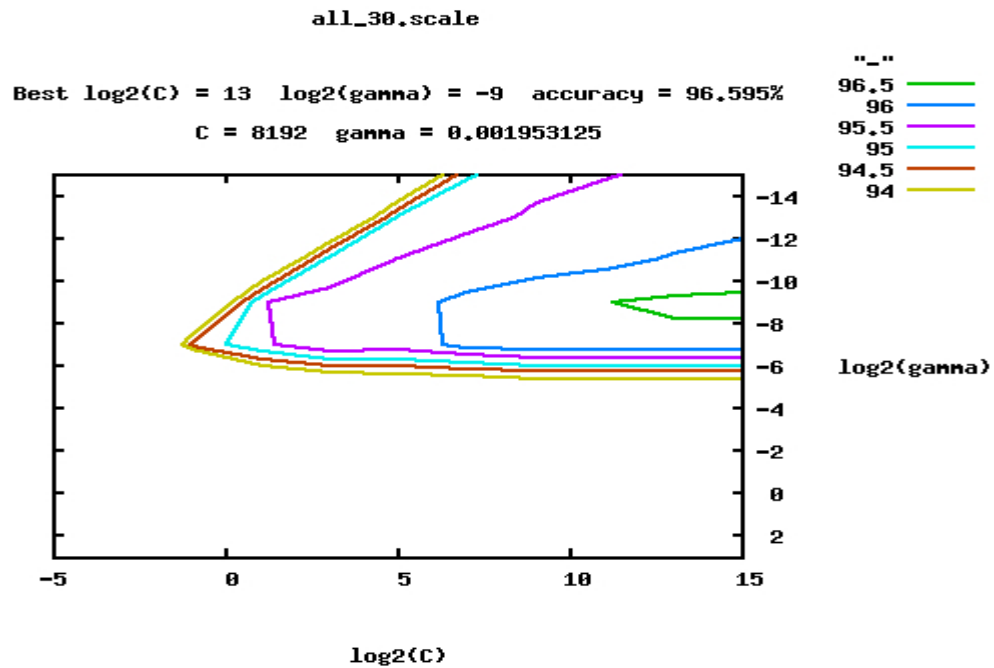
**Figure 5.1:** Figure showing 2-fold cross-validation on a data set of 30 samples per class.



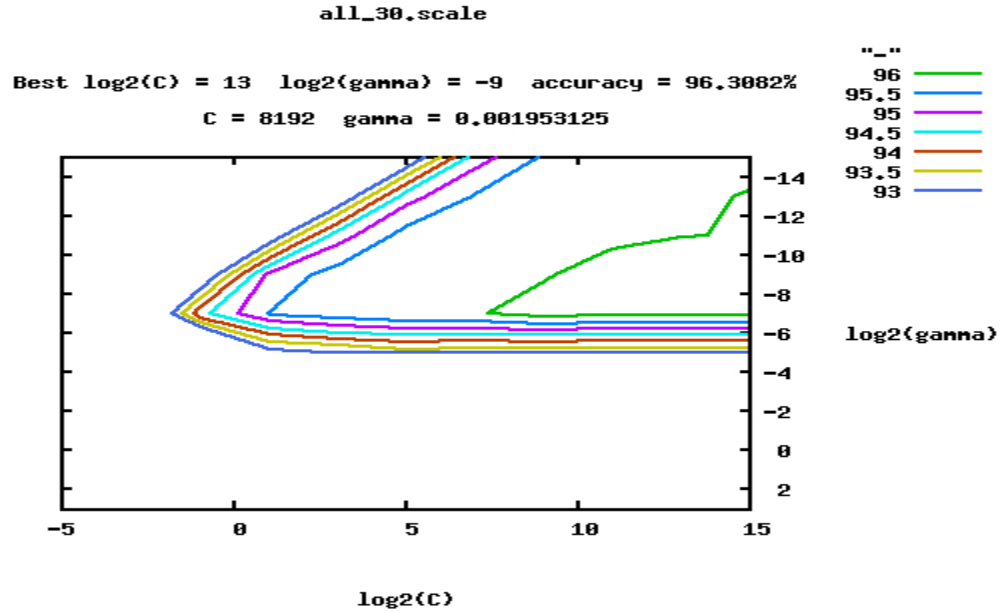
**Figure 5.2:** Figure showing 3-fold cross-validation on a data set of 30 samples per class



**Figure 5.3:** Figure showing 4-fold cross-validation on a data set of 30 samples per class.



**Figure 5.4:** Figure showing 5-fold cross-validation on a data set of 30 samples per class.



**Figure 5.5:** Figure showing 6-fold cross-validation on a data set of 30 samples per class.

## 5.2 SCHEME 2:

### K-Fold Cross-Validation of Datasets of 50 Samples Per Class

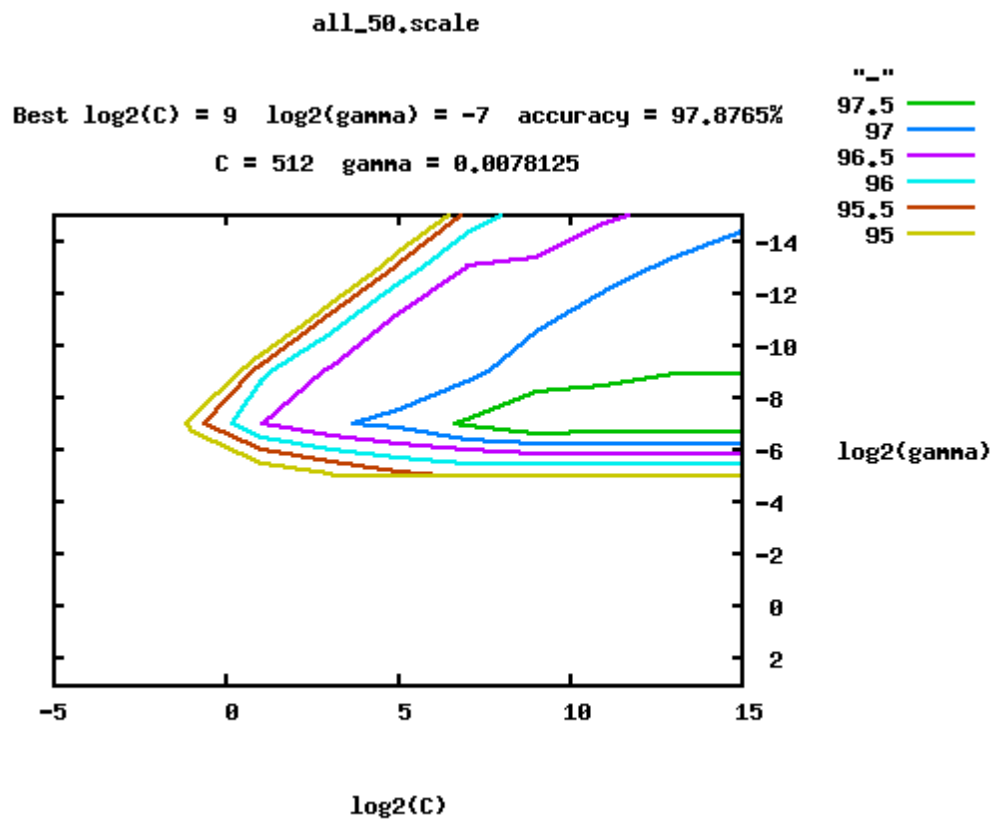
There are 95 classes in our datasets. 50 samples of each class have been used for cross-validation. Maximum accuracy of 2-fold, 3-fold, 4-fold, 5-fold, 6-fold cross-validation is given in the Table 4.2.

**Table 5.2:** accuracy of Gurmukhi strokes for 2-fold, 3-fold, 4-fold, 5-fold, 6-fold cross-validation for 50 samples per class.

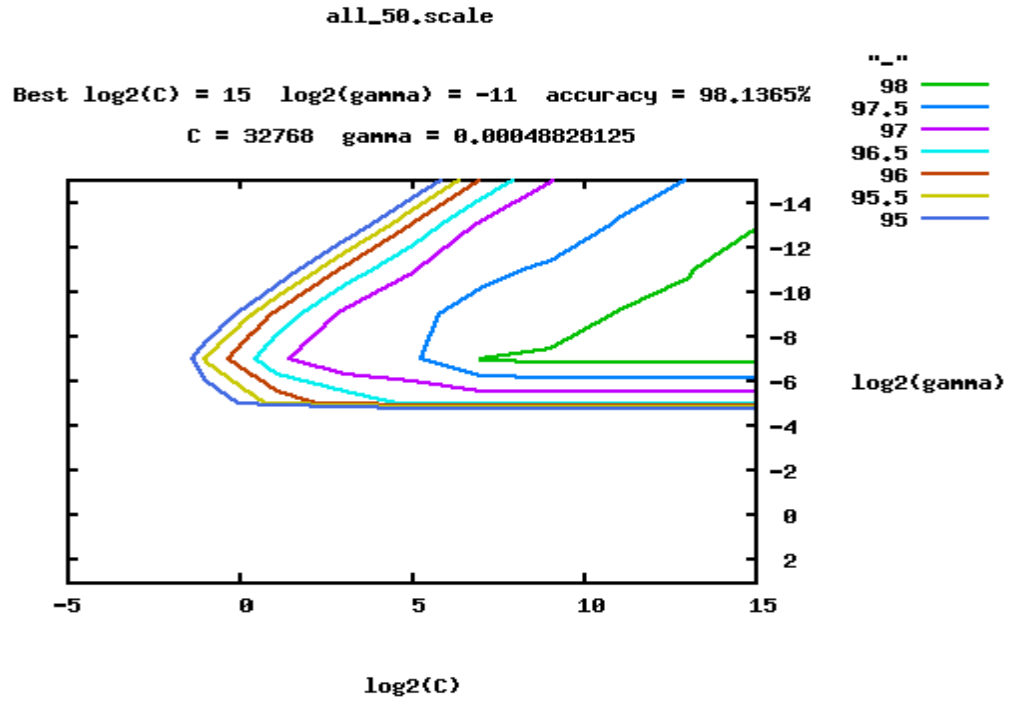
	Accuracy (in %)	Gamma( $\gamma$ )	Penalty factor(C)
<b>2-fold</b>	97.87	.0078125	512
<b>3-fold</b>	98.13	.00048828125	32768
<b>4-fold</b>	98.33	.0078125	2048

<b>5-fold</b>	98.46	.0078125	512
<b>6-fold</b>	98.43	.0078125	128

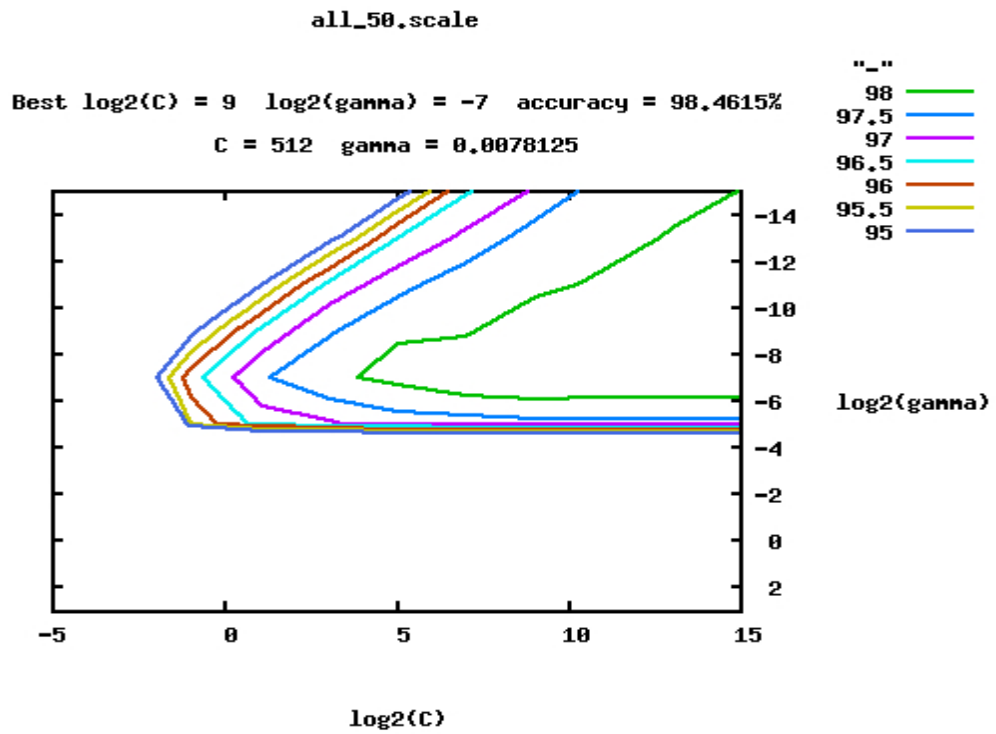
Figure 5.6 to Figure 5.10 shows all the accuracy achieved on 2-fold, 3-fold, 4-fold, 5-fold and 6-fold cross-validation for 50 samples per class.



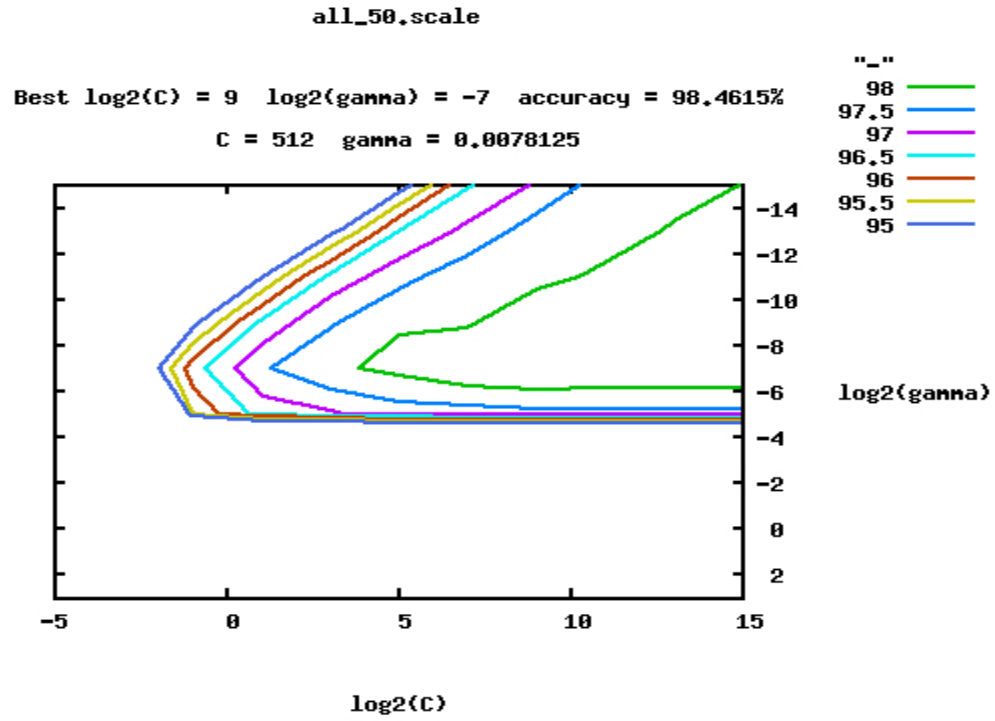
**Figure 5.6:** Figure showing 2-fold cross-validation on a data set of 50 samples per class.



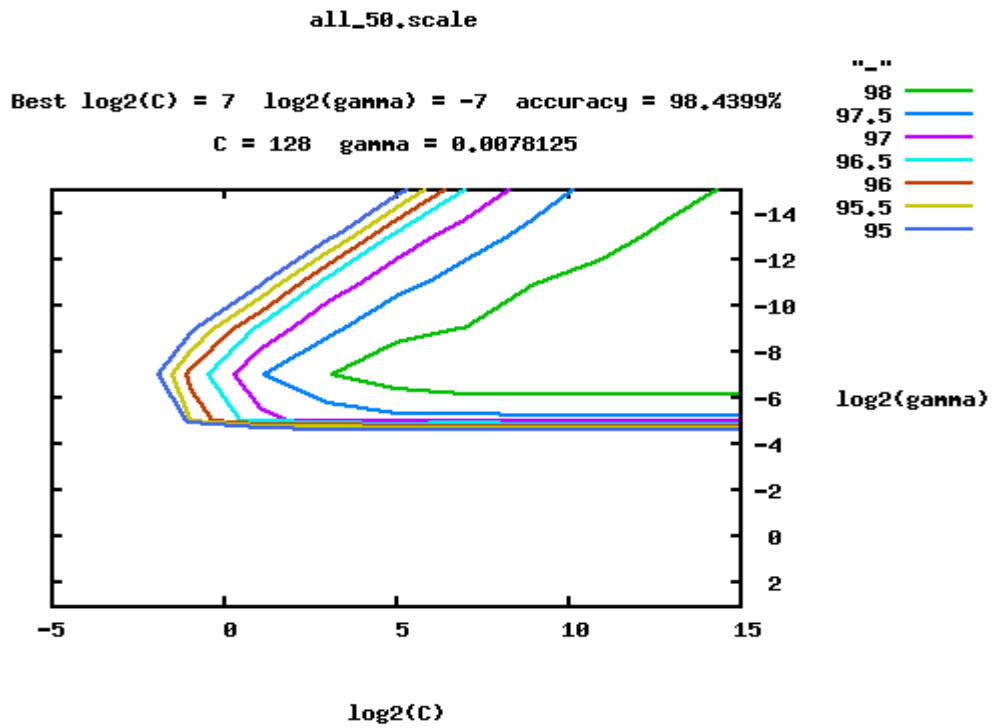
**Figure 5.7:** Figure showing 3-fold cross-validation on a data set of 50 samples per class.



**Figure 5.8:** Figure showing 4-fold cross-validation on a data set of 50 samples per class.



**Figure 5.9:** Figure showing 5-fold cross-validation on a data set of 50 samples per class.



**Figure 5.10:** Figure showing 6-fold cross-validation on a data set of 50 samples per class

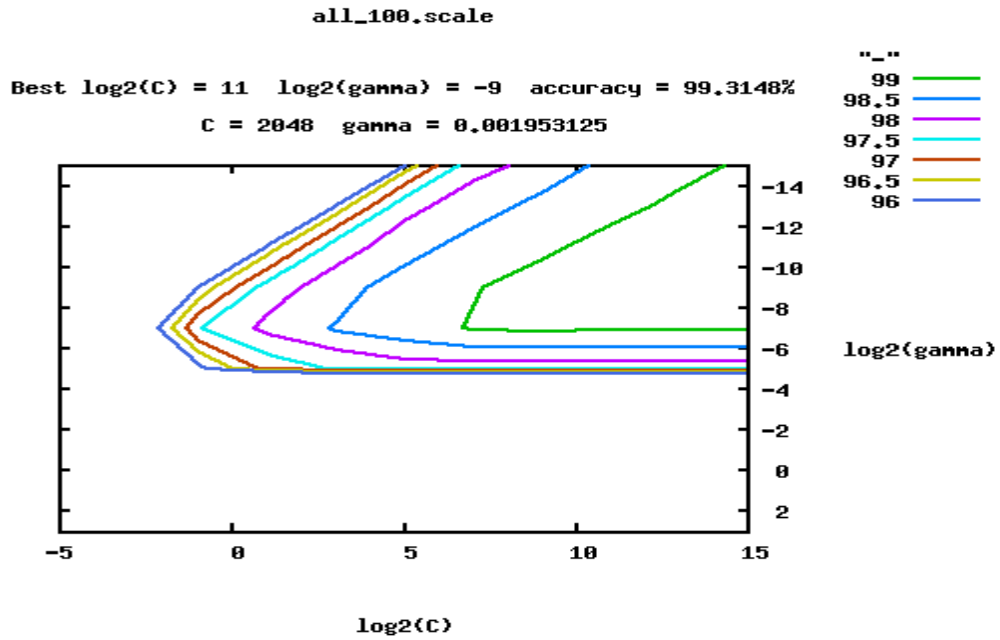
### 5.3 SCHEME 3: *K*-Fold Cross-Validation of Datasets of 100 Samples Per Class

There are 95 classes in our datasets. 100 samples of each class have been used for cross-validation. Maximum accuracy of 2-fold, 3-fold, 4-fold, 5-fold, 6-fold cross-validation is given in the Table 4.3.

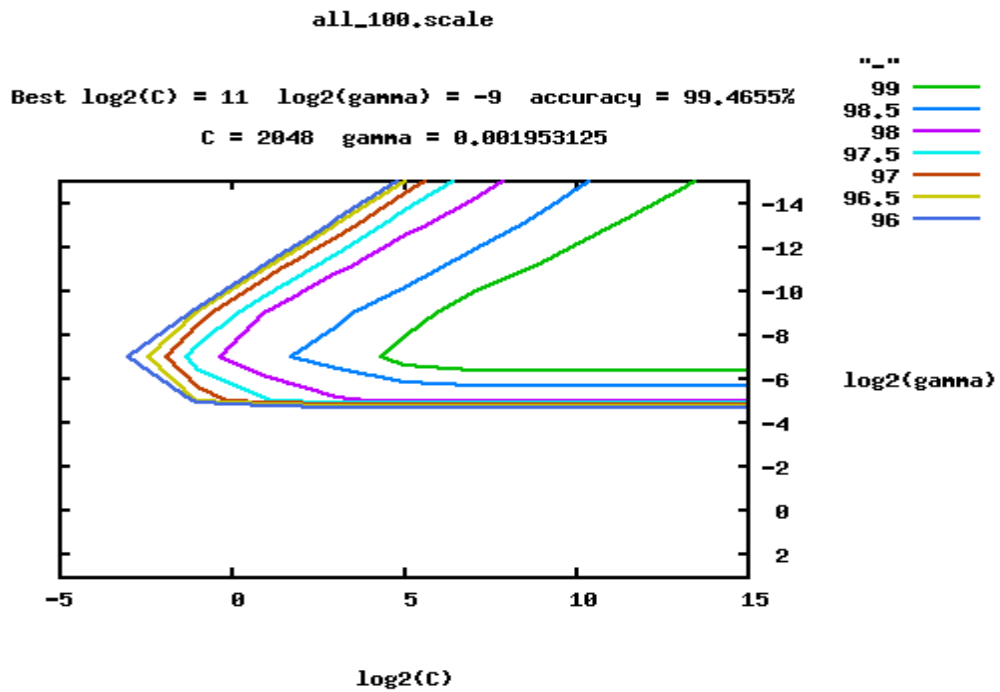
**Table 5.3:** Accuracy of Gurmukhi strokes for 2-fold, 3-fold, 4-fold, 5-fold, 6-fold cross-validation for 100 samples per class.

	<b>Accuracy (in %)</b>	<b>Gamma(<math>\gamma</math>)</b>	<b>Penalty factor(C)</b>
<b>2-fold</b>	99.31	.001953125	2048
<b>3-fold</b>	99.46	.001953125	2048
<b>4-fold</b>	99.45	.001953125	512
<b>5-fold</b>	99.46	.001953125	2048
<b>6-fold</b>	99.47	.001953125	2048

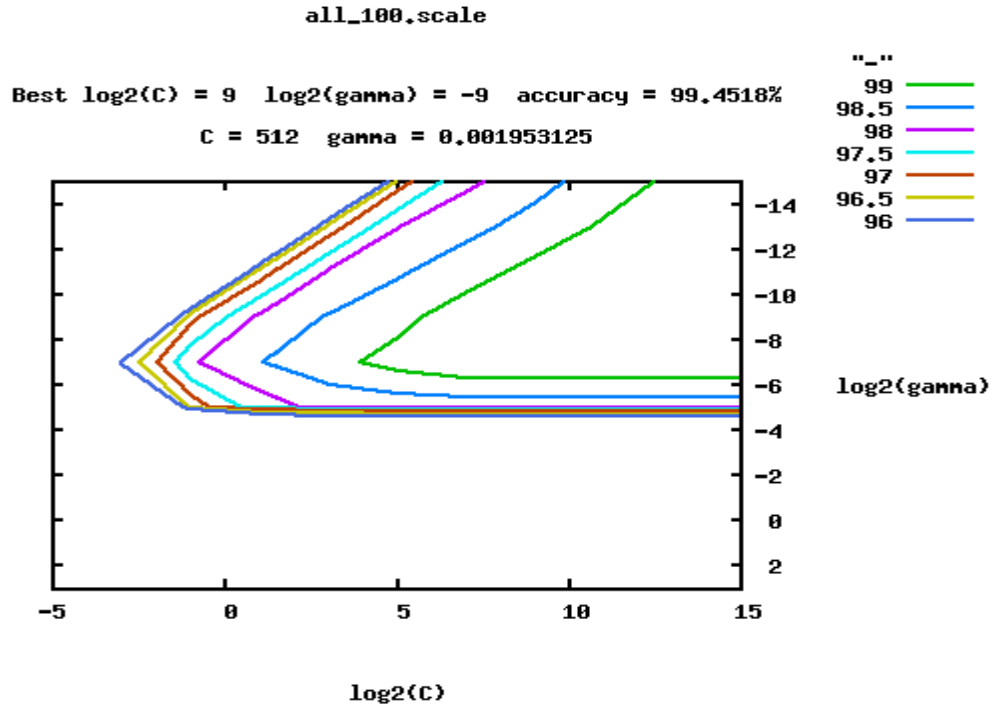
Figure 5.11 to Figure 5.15 shows all the accuracy achieved on 2-fold, 3-fold, 4-fold, 5-fold and 6-fold cross-validation for 100 samples per class.



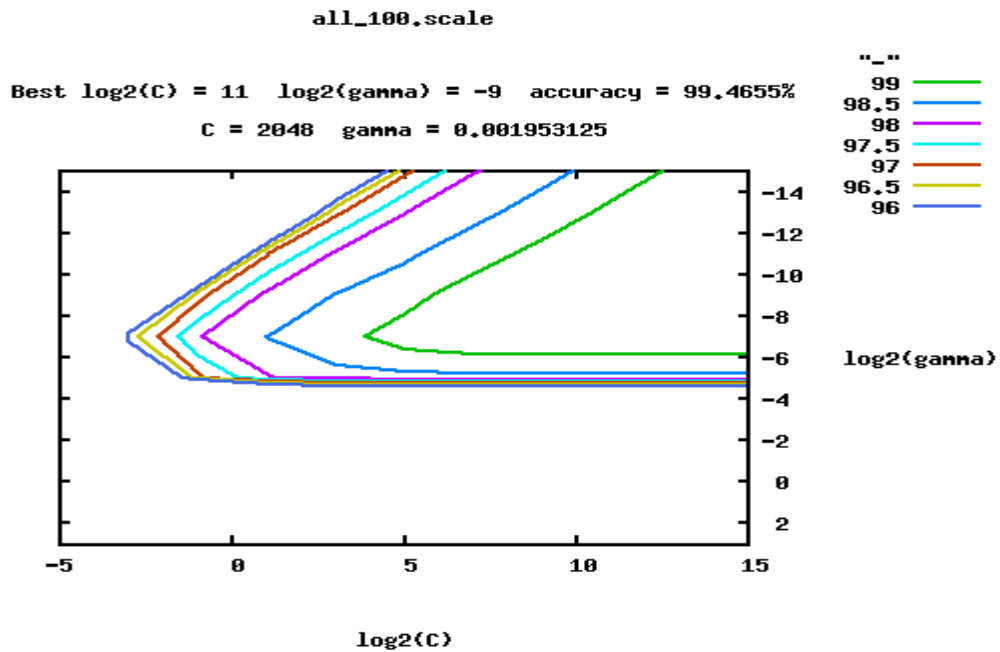
**Figure 5.11:** Figure showing 2-fold cross-validation on a data set of 100 samples per class.



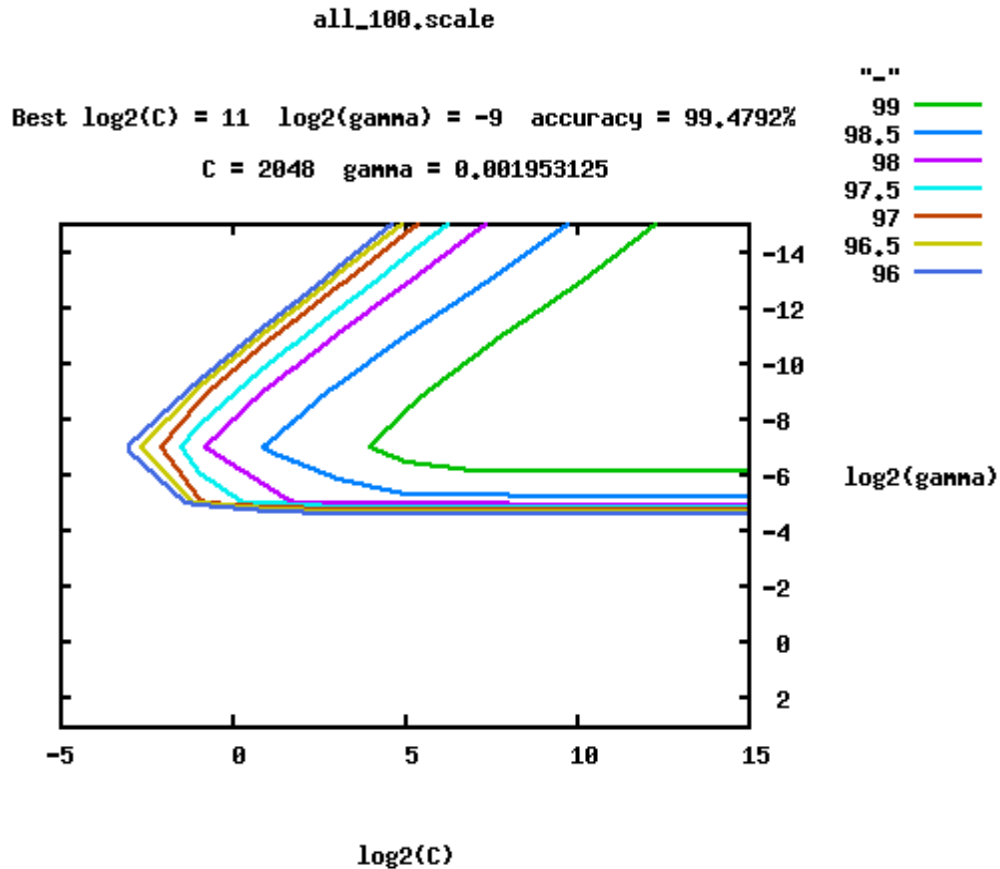
**Figure 5.12:** Figure showing 3-fold cross-validation on a data set of 100 samples per class.



**Figure 5.13:** Figure showing 4-fold cross-validation on a dataset of 100 samples per class.



**Figure 5.14:** Figure showing 5-fold cross-validation on a dataset of 100 samples per class.


















**Figure 5.15:** Figure showing 6-fold cross-validation on a dataset of 100 samples per class.

### 5.4 Testing accuracy of each class tested by different writers.

In the below Table no 5.4, we have shown the testing accuracy of our single engine. We have taken 94 classes. Many writers have been used to get the testing accuracy. On every class 50 samples is tested. In the below table, we have also given the confusion between the classes. Few of the classes is confused between each other and some of the is confused to some others. Some of the classes id not performing well on testing but majority of the classes is going very well.





















**Table No. 5.5:** Testing accuracy of each class tested by different writers.

Class ID	Shape	Number of Sample tested	Recognized	Confusion with Class ID	Accuracy (in %)
122		50	46	154(4)	92
123		50	50		100
124		50	35	224(2),122(8),154(5)	70
125		50	5	149(40),147(2),218(3)	10
126		50	5	211(25),123(20)	10
128		50	50		100
133		50	50		100
134		50	50		100
141		50	50		100
142		50	50		100
143		50	50		100
144		50	50		100
145		50	50		100
146		50	50		100
147		50	50		100

148	८	50	40	211(10)	80
149	८	50	50		100
150	८	50	50		100
151	८	50	50		100
152	८	50	50		100
153	८	50	50		100
154	८	50	50		100
155	८	50	50		100
156	८	50	50		100
157	८	50	50		100
158	८	50	50		100
159	८	50	50		100
160	८	50	50		100
161	८	50	50		100
163	—	50	5	122(25),153(20)	10
164	८	50	20	151(15),183(15)	40
165	८	50	50		100
166	८	50	50		100
167	८	50	50		100

168	ጆ	50	50		100
169	ጆ ጆ	50	50		100
170	ጆ	50	48	171(2)	96
171	ጆ	50	50		100
172	ጆ	50	50		100
173	ጆ ጆ	50	50		100
174	ጆ ጆ	50	50		100
175	ጆ ጆ	50	50		100
176	ጆ ጆ	50	50		100
177	ጆ	50	50		100
179	ጆ	50	50		100
180	ጆ ጆ	50	50		100
181	ጆ ጆ	50	50		100
182	ጆ ጆ	50	50		100
183	ጆ	50	50		100
184	ጆ ጆ	50	50		100
185	ጆ ጆ	50	45	399(5)	90
186	ጆ ጆ	50	41	168(4),214(5)	82
187	ጆ ጆ	50	50		100

188	ገጽ	50	50		100
189	ገ	50	20	147(10),207(5),122(15)	40
190	ኣ	50	50		100
191	ጠ	50	50		100
192	ጠ	50	50		100
193	ቲ	50	50		100
194	ቲ	50	50		100
195	ቲ	50	50		100
196	ቲ	50	50		100
197	ሀ	50	44	125(2),216(4)	100
198	ሀ	50	4	128(45)	10
200	ሀ	50	50		100
201	ሀ	50	50		100
202	ሀ	50	50		100
203	ሀ	50	50		100
204	ሀ	50	50		100
205	ሀ	50	50		100
206	ሀ	50	50		100
207	ሀ	50	50		100

208		50	50		100
209		50	50		100
210		50	50		100
211		50	50		100
212		50	50		100
214		50	45	168(5)	90
215		50	50		100
216		50	35	122(15)	70
217		50	50		100
218		50	50		100
220		50	50		100
222		50	50		100
223		50	50		100
224		50	50		100
225		50	50		100
226		50	50		100
351		50	35	215(15)	70
399		50	50		100
101		50	0	123(40),153(10)	0
106		50	3	153(12),399(10),158(15)	6

## CHAPTER 6

---

### CONCLUSION AND FUTURE SCOPE

---

#### CONCLUSION

This report depicts the pre-processing and the recognition stage. Pre-processing enhance the recognition accuracy of OHWR system. We have done recognition process using four kernels of SVM. Result acquired are sensibly good when the RBF kernel is used. Outstanding accuracy rate of 99.4% is demonstrated by RBF kernel. The work done in Agrawal, (2012) obtained a maximum accuracy rate of 91.4% and Khurana, (2013) obtained a maximum accuracy of 94.4%. But in this work, We have been able to achieve an accuracy rate of 99.4% when a six fold cross-validation strategy is applied on the datasets of 100 samples from each class. We have used the five algorithms of pre-processing in this thesis such as:

- Removal of duplicate points
- Normalization of Size and centering
- Interpolating missing points
- Smoothing of stroke
- Resampling

#### Future Scope

- To enhance the accuracy of handwritten recognition system one can include some new pre-processing processes.
- Lots of classes have confusion between them, so there is a possibility to remove that confusion.
- We can get more testing accuracy if the problem of confusing classes is eliminated.
- Slant correction can be included for enhancing the accuracy of the system because most of the user writes in cursive way .
- The algorithms, which has higher complexity and take large space, can be designed to decrease the time complexity and space of the algorithms. Space and

complexity are the two most important parameter which can be in mind while developing an algorithm.

- One can include more features like diagonal and direction for getting higher accuracy.
- The work can be enhanced by increasing the size of the datasets and to increase the number of users.

## REFERENCES

---

- 1 T. Artieres and P. Gallinari, "Stroke level HMMs for online handwriting recognition". Proceedings of *IWFHR*, pp. 227-232, 2000.
- 2 B. Bontempiand, A. Marcelli. "A genetic learning system for on-line character recognition", Proceedings of 12<sup>th</sup> *IAPR International Conference on Pattern Recognition*, vol. 2, pp. 83-87, 1994.
- 3 A. Brakensiek, A. Kosmalaand, G. Rigoll. "Evaluation of Confidence Measures for On-Line Handwriting Recognition", *Springer-Berlin Heidelberg*, pp. 507–514, 2002.
- 4 S. Cho, "Neural-network classifiers for recognizing totally unconstrained handwritten numerals", *IEEE Transactions on Neural Networks*, vol. 8, no. 1, pp. 43- 53, 1997.
- 5 L. Duneau and B. Dorizzi. "Online cursive script recognition: a system that adapts to an unknown user", Proceedings of *International Conference on Pattern Recognition*, vol. 2, pp. 24, 1994.
- 6 R. F. Farag. "Word-level recognition of cursive script", *IEEE Transactions on Computers*, vol. 28, no. 2, pp. 172-175,1979.
- 7 J. Brown, M. K. and W. Turin. "HMM based on-line handwriting recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 10, pp. 1039-1045, 1996.
- 8 Hu. J. Lim, S. G. and, M. K. Brown. "Writer independent on-line handwriting recognition using an HMM approach", *Pattern Recognition*, vol. 33, no. 1, pp. 133-147, 2000.
- 9 S. Impedovo." Plane curve classification through Fourier descriptors: An application to Arabichand-written numeral recognition", In Proceedings of 7<sup>th</sup> *International Conference on Pattern Recognition*, pp. 1069-1072, 1984.
- 10 S. Jaeger, S. Manke, J. Reichert and A. Waibel. "Online handwriting recognition: the Npen++ Recognizer". *International Journal of Document Analysis and Recognition*, vol. 3, no. 3, pp. 169-180, 2001.

- 11 Y. Kimura, T. Wakahara and K. Odaka. "Combining statistical pattern recognition approach with neural networks for recognition of large-set categories", *Proceedings of International Conference on Neural Networks*, vol. 3, pp. 1429-1432, 1997.
- 12 J. M. Kurtzberg and C. C. Tappert." Segmentation procedure for handwritten symbols and words". *IBM Tech, Disclosure Bull*, vol. 25, pp. 3848-3852, 1982.
- 13 J. M. Kurtzberg. "Feature analysis for symbol recognition by elastic matching", *IBM Journal of Research and Development*, vol. 31, no. 1, pp. 91-95, 1987.
- 14 M. Nakai, N. Akira, H. Shimodiara and S. Sagayama. "Substroke approach to hmm-based online Kanji handwriting recognition", *Proceedings of International Conference on Document Analysis and Recognition*, pp. 491-495, 2001.
- 15 R. K. Powalka, , N. Sherkat and R. J. Whitrow. "The Use of Word Shape Information for Cursive Script Recognition", *Fourth International Workshop on Frontiers of Handwriting Recognition*, Taiwan, pp. 67-76, 1994.
- 16 P. Scattolin, "Recognition of handwritten numerals using elastic matching", Master's thesis, Concordia University, Canada, 1995.
- 17 H. Beigi, K. Nathan, G. J. Clary and J. Subhramonia. "Size normalization in unconstrained online handwritng recognition", *Proceedings ICIP*, pp. 169-173, 1994.
- 18 S. V. Rajashekararadhya and P. V. Ranjan. "Support Vector Machine Based Handwritten Numeral Recognition of Kannada Script". *IEEE International Advance Computing Conference*, pp. 381-386, 2009.
- 19 A. Sharma. "Online handwritten Gurmukhi character recognition", *Ph.D. thesis*, School of Mathematics and Computer Applications, Thapar University, Patiala, 2009.
- 20 A. Sharma, R. Sharma and R. K. Sharma. "Online Handwritten Gurmukhi Character Recognition Using Elastic Matching", *Congress on Image and Signal Processing*, pp. 391-396, 2008.
- 21 A. Sharma, R. Sharma and R. K. Sharma. "Rearrangement of Recognized Strokes in Online Handwritten Gurmukhi Words Recognition", *10th International Conference on Document Analysis and Recognition*, pp. 1241-1245, 2009.
- 22 H. Beigi, "Pre-processing and dynamics of online handwriting data, feature

- extraction and recognition”. Proceedings of fifth *IWFHR*, pp. 255-258, 1996.
- 23 J. Subrahmonia, T. Zimmerman. “Pen computing: challenges and applications”. *Proceedings of 15th International Conference on Pattern Recognition*, vol. 2, pp. 60-66, 2000.
  - 24 A. L. Spitz. “Shape-based word recognition”, *International Journal of Document Analysis and Recognition*, vol. 1, pp. 178-190, 1999.
  - 25 C. Y. Suen, M. Berthod and S. Mori. “Automatic Recognition of Hand printed Character-the State of the Art”, Proceedings of *IEEE*, vol. 68, no. 4, pp. 469-487, 1980.
  - 26 C. C. Tappert, C.Y. Suen and T. Wakahara. “The state of art in online handwriting recognition”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12 no. 8, pp. 787-808, 1990.
  - 27 S. R. Veltman and R. Prasad. “Hidden markov models applied to online handwritten isolated character recognition”, *IEEE Transactions on Image Processing*, vol. 3, no. 3, pp. 314-318, 1994.
  - 28 T. Wakahara and K. Odaka. “Online cursive kanji character recognition using stroke-based affine transformation”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 12, pp. 1381-1385, 1997.
  - 29 J. Subrahmonia. “Similarity measures for writer clustering. In L. R. B. Schomaker and L. G. Vuurpijl (Eds.)” Proceedings of the Seventh *International Workshop on Frontiers in Handwriting Recognition*, pp. 541-546, 2000.
  - 30 J. Zhou, Q. Gan, K. Krzyzak, C. Y. Suen. “Recognition of handwritten numerals by quantum neural network with fuzzy features”, *International Journal of Document Analysis and Recognition*, vol. 2, pp. 30-36, 1999.
  - 31 S. S. Keerthi and C. J. Lin. “Asymptotic behaviors of support vector machines with Gaussian kernel”. *Neural Computation*, 15(7):1667–1689, 2003.
  - 32 H. T. Lin and C. J. Lin. “A study on sigmoid kernels for SVM and the training of nonPSD kernels by SMO-type methods”. Technical report, Department of Computer Science, National Taiwan University, 2003. URL <http://www.csie.ntu.edu.tw/~cjlin/papers/tanh.pdf>.

- 33 M. Unser, A. Aldroubi, M. Eden, "B-Spline signal processing: part II - efficient design and applications". *IEEE Transactions on Signal Processing*, vol. 41, no. 2, pp. 834-848, 1993.
- 34 A. K. Jain, R. P. W. Duin and J. Mao. "Statistical pattern recognition: a review". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1. pp. 4-37, 2000.
- 35 I. Hosny, S. Abdou and A. Fahmy. "Using Advanced Hidden Markov Models for Online Arabic Handwriting Recognition", *First Asian Conference on Pattern Recognition*, pp.565-569, 2011.
- 36 P. Bilane, E. Youssef, B. Eter, C. Sarraf and J. Constantin. "Simplified Point to Point Correspondence of the Euclidean Distance for Online Handwriting Recognition". *IEEE International Conference on Signal Processing and Communications*, pp. 1011- 1014, 2007.
- 37 B. Verma, M. Blumenstein, M. Ghosh. "A novel approach for structural feature extraction: Contour vs. direction". *Pattern Recognition Letters*, vol. 25, no. 9, pp. 975-988, 2004.
- 38 E. Kavallieratou, N. Fakatakis, G. Kolkkinakis. "An unconstrained handwriting recognition system". *International Journal of Document Analysis and Recognition*, vol. 4, no. 4, pp. 226-242, 2002.
- 39 S. Singh, A. Aggarwal and R. Dhir. "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, 2012.

## **VIDEO REFERENCE**

<https://www.youtube.com/watch?v=hOjYOuqhn38>