DFT Based Feature Extraction Technique for Recognition of Online Handwritten Gurmukhi Strokes using SVM

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

Master of Technology

in

Computer Science and Applications

Submitted by

Keerti Aggarwal

(Roll No. 601403013)

Under the supervision of:

Dr. R.K. Sharma

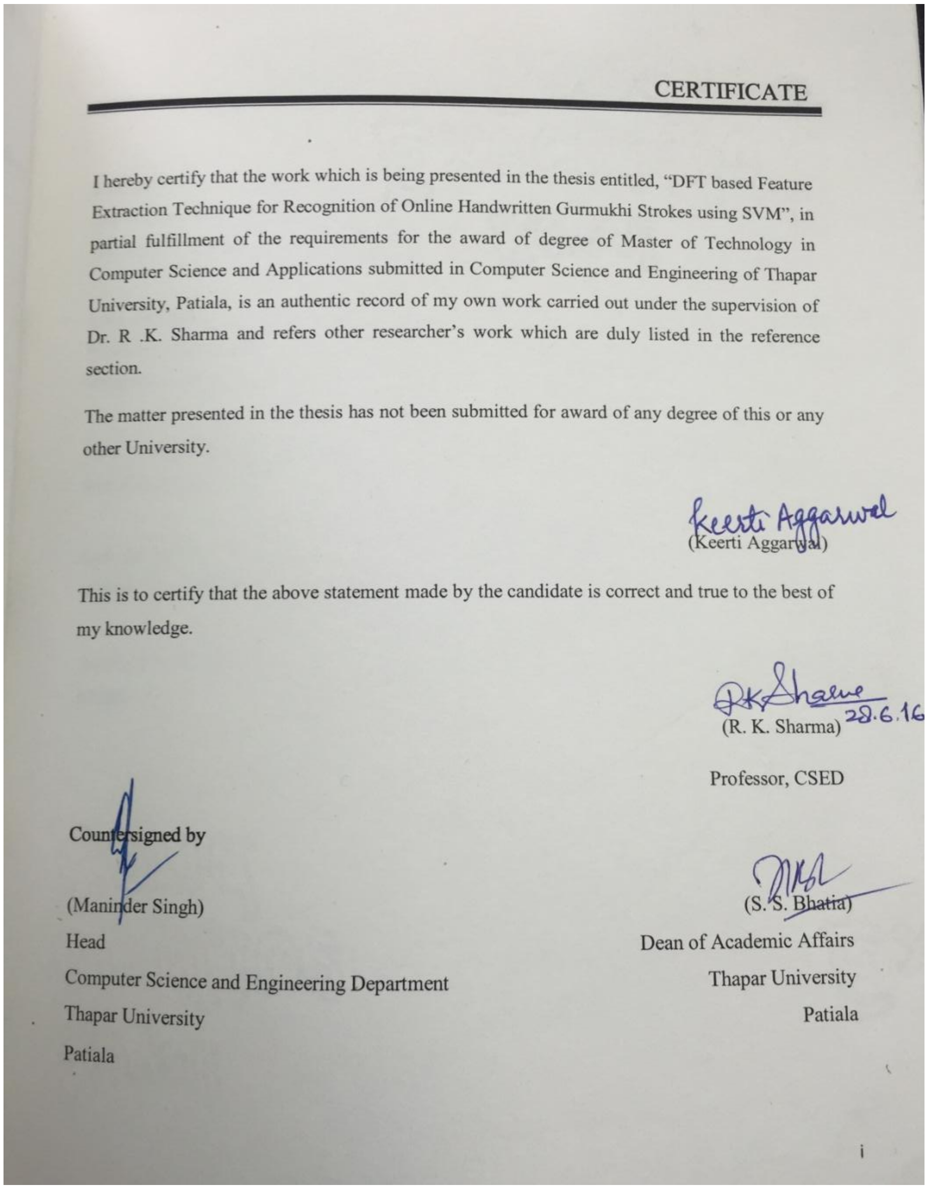
Professor

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

THAPAR UNIVERSITY

PATIALA - 147004

JUNE 2016



CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled, "DFT based Feature Extraction Technique for Recognition of Online Handwritten Gurmukhi Strokes using SVM", in partial fulfillment of the requirements for the award of degree of Master of Technology in Computer Science and Applications submitted in Computer Science and Engineering 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 degree of this or any other University.

Keerti Aggarwal
(Keerti Aggarwal)

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

RKSharma 28.6.16
(R. K. Sharma)

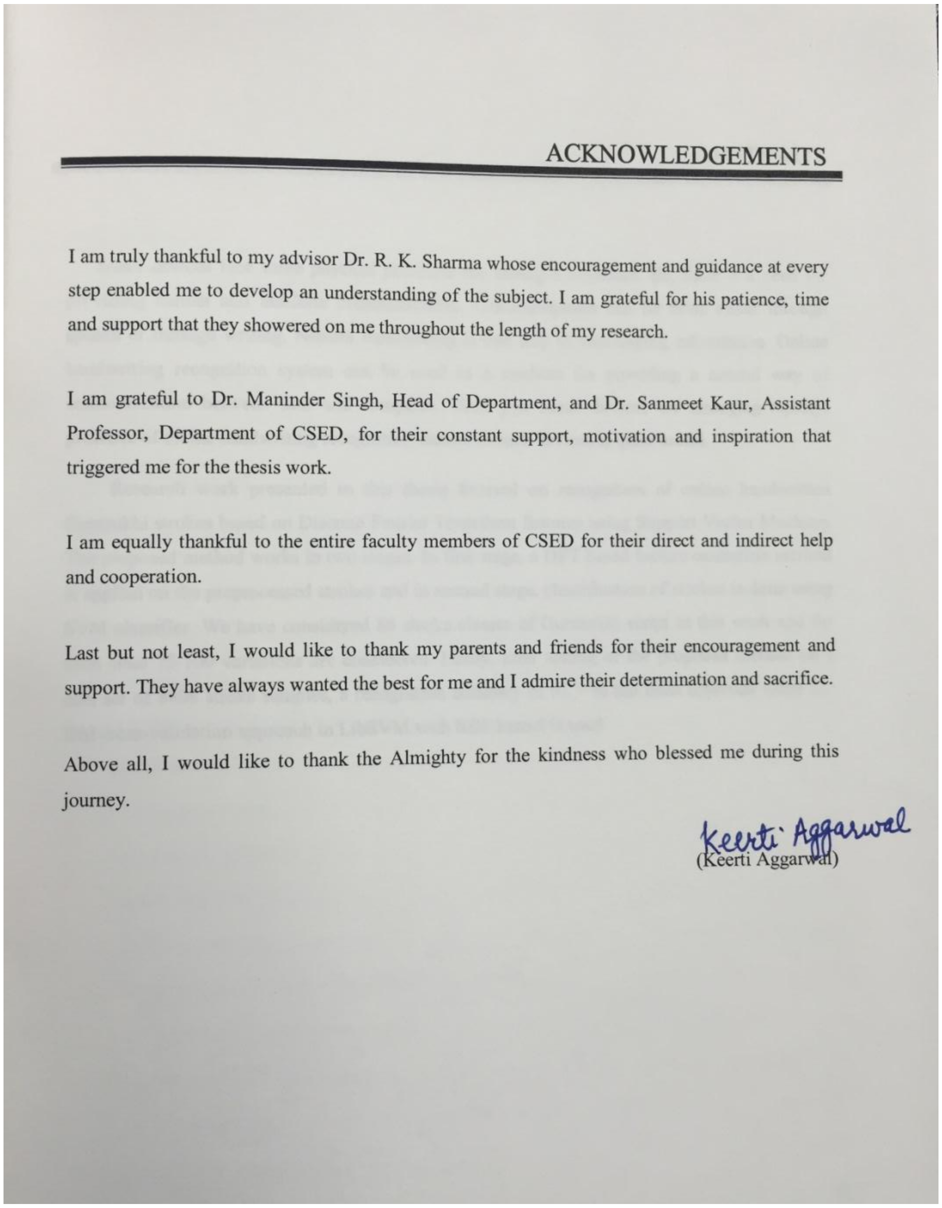
Professor, CSED

Countersigned by

(Maninder Singh)
Head
Computer Science and Engineering Department
Thapar University
Patiala

(S. S. Bhatia)
Dean of Academic Affairs
Thapar University
Patiala

i



ACKNOWLEDGEMENTS

I am truly thankful to my advisor Dr. R. K. Sharma whose encouragement and guidance at every step enabled me to develop an understanding of the subject. I am grateful for his patience, time and support that they showered on me throughout the length of my research.

I am grateful to Dr. Maninder Singh, Head of Department, and Dr. Sanmeet Kaur, Assistant Professor, Department of CSED, for their constant support, motivation and inspiration that triggered me for the thesis work.

I am equally thankful to the entire faculty members of CSED for their direct and indirect help and cooperation.

Last but not least, I would like to thank my parents and friends for their encouragement and support. They have always wanted the best for me and I admire their determination and sacrifice.

Above all, I would like to thank the Almighty for the kindness who blessed me during this journey.

Keerti Aggarwal

(Keerti Aggarwal)

ABSTRACT

Small devices face some physical problems for having a keyboard. So, there is a need for providing human and machine communication. Communication can be done either through speech or through writing. Natural handwriting is one way of exchanging information. Online handwriting recognition system can be used as a medium for providing a natural way of communication between user and computer. Since pen based devices are emerging rapidly, presence of online handwriting recognition feature in such devices is quite useful.

Research work presented in this thesis focused on recognition of online handwritten Gurmukhi strokes based on Discrete Fourier Transform features using Support Vector Machine. The proposed method works in two stages. In first stage, a DFT based feature extraction method is applied on the preprocessed strokes and in second stage, classification of strokes is done using SVM classifier. We have considered 86 stroke classes of Gurmukhi script in this work and for each class 75-100 variations are considered. Lastly, after testing of the proposed method on a data set of 8408 stroke samples, a recognition accuracy of 91.7 % has been achieved when 11-fold cross-validation approach in LibSVM with RBF kernel is used.

TABLE OF CONTENTS

CHAPTER 1

CHAPTER 6

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

DFT	Discrete Fourier Transform
DTW	Dynamic Time Warping
HMM	Hidden Markov Model
MCDF	Modified Centre Distance Feature
OHWR	Online Handwriting Recognition
PC	Personal Computer
PCA	Principal Component Analysis
PDA	Personal Digital Assistant
SVM	Support Vector Machine

CHAPTER 1

INTRODUCTION

As portable computers turn out to be more personal and are made smaller in size, they reach some physical problems for having a keyboard (Beigi and Watson, 1993). Also, there are some scripts like Chinese and Japanese for which inputting data to computers is quite difficult as they have a large number of alphabets. So, there is a need for providing human and machine communication. There are two ways of communication that can occur between human and machine that is through speech and through writing. In speech recognition, users who are unable to write can easily communicate via speech with computers. But speech recognition does not perform well in noisy environment. Here, in this work, we have discussed about the communication with computers via handwriting. There are various issues in handwriting recognition. There are variations in writing styles of different users and also variation may occur in one's own writing style which is a major issue in handwriting recognition. Attaining high recognition accuracy in such handwriting recognition systems is quite a difficult task.

Since touch screen devices like tablet PCs or PDAs are emerging rapidly, the presence of OHWR feature in touch screen devices is quite useful. It may provide a natural form of communication between user and these devices. The present work focused on online handwriting recognition of Gurmukhi script.

1.1 Foundation

Handwriting recognition is the capability of a computer to accept and interpret handwritten input from sources like paper documents, touch screens and other devices. Handwriting recognition is of two types: one is offline and other is online. In offline handwriting recognition system, handwritten text is first scanned and then scanned image is passed to the recognition system (Ghosh and Ghosh, 2005). In online handwritten recognition system, automatic conversion of input text takes place as it is written on a writing device. Online handwriting recognition is done with the help of pen computing devices that consist of a sensor which can track the position of pen tip movements and pen-up/pen-down switching information. The present work focused on the recognition of online handwritten strokes of Gurmukhi script.

1.2 Online Handwriting Recognition System

OHWR system automatically converts the text as it is written on pen computing devices like digitizer, tablet PC. OHWR recognition systems have five phases, namely, data collection, preprocessing, feature extraction, recognition and postprocessing (Sharma *et al.*, 2009). These phases are shown in Figure 1.1.

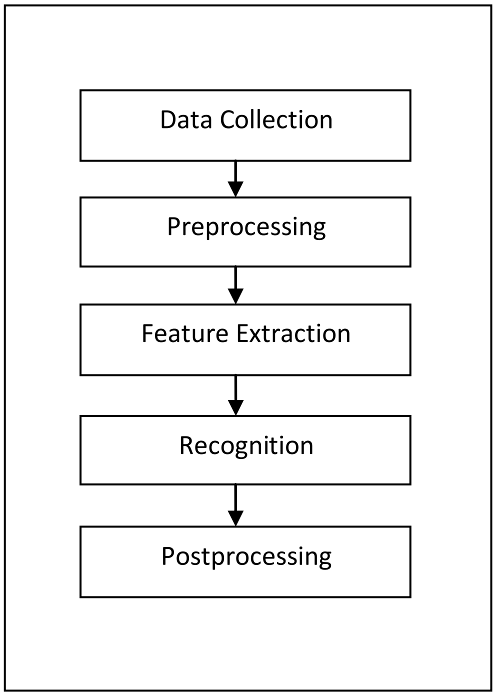


Figure 1.1: Phases of OHWR system.

1.2.1 Data collection

In this phase, data is collected from user using a pen computing device like tablet PC or digitizer. Pen computing device contains a sensor which can monitor the pen-tip positions and pen-up/pen-down switching. These devices capture the sequence of coordinates of moving pen. The pen trace captured between pen-up/pen-down switching is called a stroke. Pen computing devices are shown in Figure 1.2.

Figure 1.2: Pen computing devices.

Some commonly used pen computing devices are Nextbook Flexx 11, Lenovo Tab 2A8-50, Asus ZenPad S 8.0, Hp Pro Slate 8, Vaio Z Canvas, Apple 12.9-inch ipad Pro Review.

1.2.2 Preprocessing

In preprocessing phase of online handwriting recognition, distortions and noise present in input data are eliminated which may occur due to software and hardware limitations. These distortions include missing points, irregular size of input data *etc.* . Preprocessing incorporates mainly five steps that are discussed in (Sharma *et al.*, 2009), namely, size normalization and centering, interpolation missing points, smoothing, slant correction, resampling of points.

1.2.3 Feature extraction

Selection of a feature extraction technique is an important task. Efficiency of OHWR system highly relies on the features which are considered as input, given to a classifier and also complexity of classification problem can be decreased by selecting suitable features. There is no standard strategy for extracting features. Features that provide good results for one script may not provide good results for other scripts. Features are of two types: one is low level features and other is high level features. Low level features include slope, curliness, width *etc*. and high level features include head line, straight line, dot *etc.* (Sharma *et al.*, 2008).

1.2.4 Recognition

In recognition phase, classifier classifies the feature vector of input stroke to one of the defined classes. Classifier compares the input feature vector with stored patterns and chooses the best matching class for input stroke. Selection of a suitable classifier ensures low misclassification and rejection rates. There are various handwriting recognition classifier such as SVM, HMM, neural networks, nearest neighbor algorithm.

1.2.5 Postprocessing

In postprocessing phase, misclassified results are corrected with the help of linguistic knowledge. All the possibilities of an individual character are studied and best suitable character is depicted.

1.3 Issues in Online Handwriting Recognition System

OHWR systems enhance the communication between user and computer. But there are various issues in online handwriting recognition systems such as variations in writing styles of different users and also, variations may occur in one’s own writing. Due to these variations and distortions that may occur during digitizing process, even the best handwriting recognition system may give unreliable results.

1.3.1 Variations in handwriting styles

Variations in writing styles of different users may occur due to varying speed of writing, varied sizes and styles of characters. Also, variations may occur in one’s own writing due to various reasons: different mood of user; the user is writing from long time; writing on dissimilar kinds of hardware devices. Figure 1.3 shows the variations in few samples of Gurmukhi script written by five different users. Figure 1.4 shows the variations in few samples of Gurmukhi script written by an individual user (Sharma *et al.*, 2009).

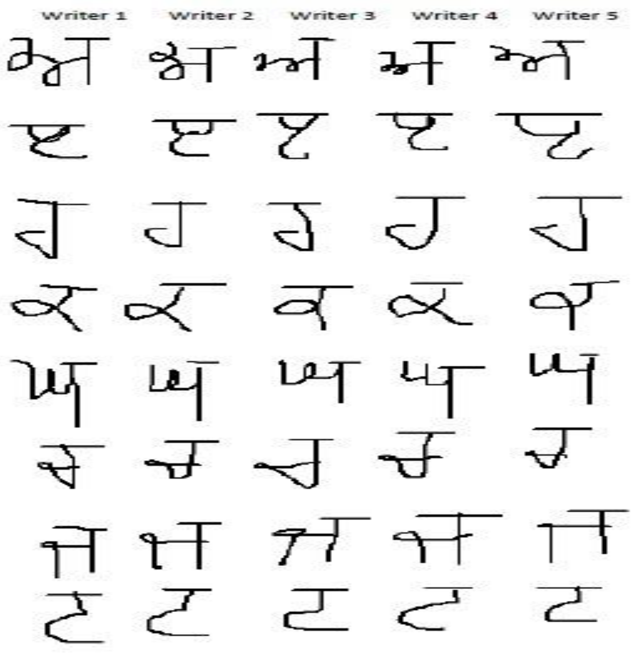


Figure 1.3: Variations in Gurmukhi characters written by five users.

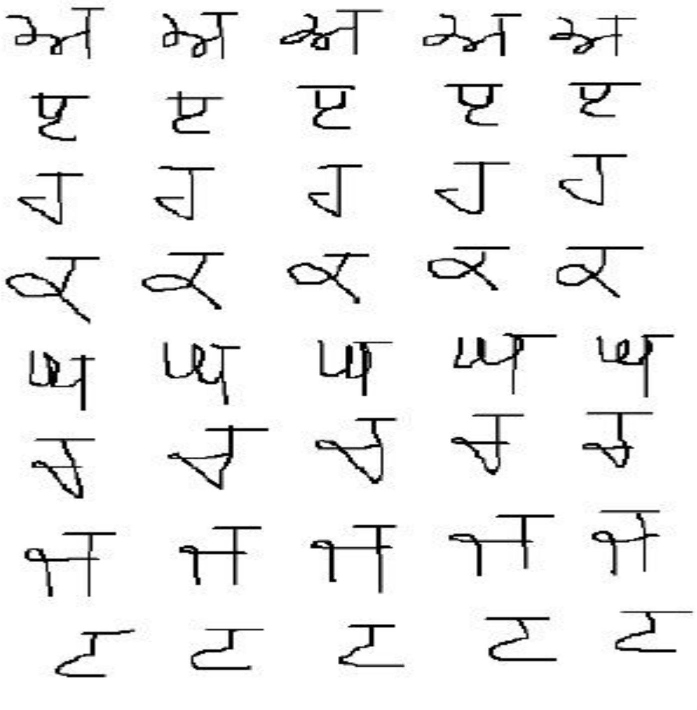


Figure 1.4: Variations in Gurmukhi characters written by individual user.

1.3.2 Personal and material factors

In personal factors, handedness of a writer is considered whether a writer is left handed or right handed. Different positions and directions are used by different handed writers while writing. Material factors depend on the hardware device that is used for writing.

1.3.3 Writer dependent vs. writer independent recognition systems

There are two categories of handwriting recognition system, namely, writer dependent and writer independent. In writer dependent handwriting recognition system, system is trained with the data samples of writers whose handwriting will be recognized in the future and in writer independent recognition system, unknown handwriting styles will be recognized. In comparison to writer independent systems, writer dependent systems have attained better accuracy rate. Also, writer independent systems are more complicated to develop as all common aspects of handwriting are needed to study.

1.4 Gurmukhi, the Script

Gurmukhi is the script of Punjabi language which is widely spoken across the globe. Punjabi is spoken by around 130 million people. The word "Gurmukhi" means "*from the mouth of Guru*".

Some salient features of this script (Sharma *et al.*, 2008) are:

1. Gurmukhi script is written from left to right.
2. Characters of Gurmukhi script contain a horizontal line at upper part, called headline.
3. A word in Gurmukhi can be divided into three zones. First zone is called upper zone, it indicates the section above the headline where some of the vowels and sub-parts of some other vowels dwell. Second zone is the middle zone, it represents the region underneath the headline where consonants and some sub-parts of vowels reside. This is the most occupied zone. Third zone is called lower zone, it denotes the region below the middle zone. These zones are shown in Figure 1.5.
4. Gurmukhi script has 41 characters and 12 *matras* (Bahri, 2011). These Gurmukhi *matras* and characters are shown in Table 1.1 and Table 1.2, respectively.

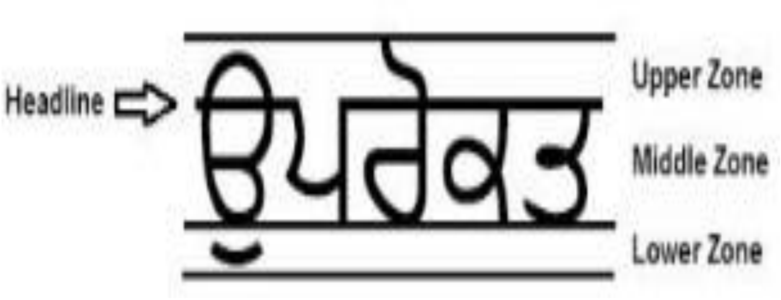


Figure 1.5: Upper, Middle and Lower zones in Gurmukhi script.

Table 1.1: Gurmukhi *matras*.

◌ਾ	ਿ◌	◌ੀ	◌ੁ	◌ੂ	◌ੇ
◌ੈ	◌ੋ	◌ੌ	◌ੰ	◌ਂ	◌ੱ

In the present study, we have considered 35 characters and 10 *matras* of Gurmukhi script. The characters ਸ਼, ਖ਼, ਗ਼, ਜ਼, ਫ਼ and ਲ਼ have not been considered in this work. The *matras* ◌ਾ and ◌ਂ have also not been considered in this work. The first three symbols shown in Table 1.2 are vowels. They are called *matra* bearers as they are used to carry *matras*.

Table 1.2: Gurmukhi characters.

ੳ	ਅ	ੲ	ਸ	ਹ
ਕ	ਖ	ਗ	ਘ	ਙ
ਚ	ਛ	ਜ	ਝ	ਞ
ਟ	ਠ	ਡ	ਢ	ਣ
ਤ	ਥ	ਦ	ਧ	ਨ
ਪ	ਫ	ਬ	ਭ	ਮ
ਯ	ਰ	ਲ	ਵ	ੜ
ਸ਼	ਖ਼	ਗ਼	ਜ਼	ਫ਼
ਲ਼				

1.5 Thesis Outline

The purpose of this thesis is to develop a recognition system for online handwritten Gurmukhi strokes. The thesis is organized in following manner.

In the present Chapter, introduction of OHWR system, issues in OHWR system and overview of Gurmukhi script is presented. Chapter 2 describes the literature survey on handwriting recognition systems. Chapter 3 describes the problem statement. Chapter 4 illustrates the various phases of preprocessing, DFT based feature extraction technique and recognition of preprocessed strokes using LibSVM. Results obtained by applying methodology described in Chapter 4 are discussed in Chapter 5. The conclusion drawn from the results obtained is discussed in Chapter 6.

CHAPTER 2

LITERATURE SURVEY

In this Chapter, a brief literature survey on the topics included in this work is presented.

Bahlmann *et al.* (2002) have proposed a new SVM kernel, called as Gaussian DTW by combining DTW and SVM techniques for online handwriting recognition. Authors have tested the proposed methodology on UNIPEN handwriting data.

Cho and Kim (2003) have proposed a system for recognition of online handwritten Hangul characters, based on Bayesian network. They have trained the system with a data set of 49049 characters written by 48 writers and tested the system using three data sets: Nation, High School, jungang. Number of samples in Nation, High school and jungang data sets are 3127, 15250, 16427. They have achieved an average accuracy of 95.7% on these three data sets.

Vinciarelli and Perrone (2003) have implemented a technique for the recognition of digits by combining offline recognizer with online handwriting recognition system. They have considered a total of 7500 samples of isolated digits. Out of which, 5000 samples are used to train the system and 2500 samples are used to test the system and using this system error rate is reduced by 43.0% as compare to purely online recognizer.

Gorgevik and Cakmakov (2004) have proposed a classifier for recognizing of handwritten digits. The three stage classifier proposed by them is based on neural network and SVM classifiers. In this classifier, classification is done by two neural networks and one SVM. First neural network is designed in such a way that it provides a low misclassification rate using strong rejection criterion. From the patterns rejected by first neural network, complex features are extracted and are forwarded to second neural network. At last, patterns rejected from second neural network are forwarded to SVM which considers only *k* top-ranked classes.

Sundaram and Ramakrishnan (2007) have proposed a three level hierarchical classification scheme for the recognition of online handwritten Tamil characters. The first level of classifier has been built using the knowledge of Tamil character’s writing rules. At the second level, number of strokes in the preprocessed character is used for classification and at the final level, *k*-Nearest Neighbor classifier has been used. Authors have achieved an average accuracy of 96.0%.

Prasanth *et al.* (2007) have proposed character based elastic matching using local features for the recognition of online handwritten Tamil and Telugu Script. DTW has been employed with four distinct feature sets, namely, x-y features, Generalized Shape context feature, Tangent Angle and Shape Context features and normalized first and second derivatives and curvature features. Recognition of data has been done using nearest neighborhood classifier with DTW distance. Authors have achieved an accuracy of 90.6% on Telugu data.

Sharma *et al.* (2008) have implemented elastic matching technique in online handwritten Gurmukhi character recognition. Recognition of character is done in two stages. In first stage, recognition of strokes is done and in second stage, recognition of character is done based on the recognized strokes. Authors have attained an accuracy of 90.1% in consideration of 60 writers and a set of 41 Gurmukhi characters.

Parui *et al.* (2008) have implemented online handwritten recognition system for Bangla characters using HMM. In the system proposed by the authors classification of characters is done in two stages. In first stage, strokes of Bangla script is recognized using HMM and in second stage, characters is recognized using stroke classification results. They have considered 24,500 samples of Bangla characters as data set and achieved an accuracy of 84.6% at stroke level and 87.7% at character level.

Sharma *et al.* (2009) have proposed a novel stride as rearrangement of recognized stroke for the recognition of online handwritten Gurmukhi characters. The proposed technique consists: stroke's identification as dependent and major dependent strokes, the rearrangement of strokes according to their positions, the combination of strokes to recognize characters. For a dataset of 2567 Gurmukhi dictionary words they had attained a recognition rate of 81.0%.

Vamvakas *et al.* (2009) have implemented a new feature extraction technique and classification method for recognition of historical documents. The method proposed by them involves feature extraction technique which is based on recursive subdivisions of the image and computation of centre of masses of each sub-image with sub-pixel accuracy. After feature extraction, hierarchical classification scheme is applied which is based on the level of granularity of feature extraction method. In their work, authors have achieved an accuracy of 97.7% and 94.5% for historical typewritten and handwritten characters respectively.

Kunwar *et al.* (2010) have proposed a method for the recognition of Kannada words using statistical dynamic space warping. Authors have proposed a heuristic approach to segment recognizable symbols from Kannada word and perform recognition of entire word. Two different estimates of first derivative, extracted from preprocessed strokes are used as features. Authors have achieved accuracies of 88.0% at akshara level and 80.0% at word level in their work.

Ghods and Kabir (2010) have demonstrated a feature extraction technique for recognizing online handwritten Farsim characters using decision tree. Authors have implemented feature extraction technique on main body of Farsi characters. They have attained an accuracy of 94.0% in their work.

Fink *et al.* (2010) have proposed a system for recognition of online handwritten Bangla characters based on HMM. They have considered cursively written words. In their work, authors have combined sub-stroke level feature representation and an HMM based writing model using context-dependent sub word units. They have achieved an accuracy of 93.1%.

Ahmed and Azeem (2011) have implemented a system for recognition of online handwritten Arabic characters using HMM. In the system implemented by them, they have removed the delayed strokes in the training and testing phases to avoid confusion in the recognition process. They have attained a recognition accuracy of 95.3% on ADAB database.

Addakiri and Bahaj (2012) have proposed a recognition system for online handwritten Arabic characters using Artificial Neural Networks. The system proposed by them involves three phases, namely, preprocessing of original image, training neural networks with feed-forward back propagation algorithm and recognition of characters using neural network technique. Authors have tested the proposed system on a data set of 1400 samples of Arabic characters written by ten users. They have attained an average accuracy of 83.0% using their approach.

Prasad *et al.* (2012) have proposed a system for the recognition of online handwritten Kannada characters using PCA and DTW. They considered 51 Kannada characters in their work. A total of 2550 samples of Kannada characters has been considered in order to train the system and 765 samples are used for testing the system. They have obtained an overall recognition rate of 87.5% using PCA and 63.7% recognition rate using DTW.

Prasad *et al.* (2012) have used PCA and DTW for the recognition of online handwritten Hindi characters. They considered 49 Hindi characters in their work. They have constructed a data set of 2450 samples for training and 735 samples for testing in order to check the system efficiency. They have obtained an overall recognition rate of 87.5% using PCA and 68.7% recognition rate using DTW.

Alijila and Kwaik (2012) have proposed a system for recognition of online handwritten Arabic character recognition system using feed forward back propagation neural network. They achieved an average recognition rate of 99.1% for trained writers and 95.7% for non-trained writers.

Gupta *et al.* (2012) have proposed a model for preprocessing the Gurmukhi strokes. In preprocessing, the input Gurmukhi stroke is processed through various transformations, namely, size normalization and centering, interpolation, uniformation, smoothing and resampling. 2197 strokes out of 2433 strokes were correctly preprocessed and authors have attained an accuracy of 90.3%.

Singh *et al.* (2012) have used Gabor filter based feature extraction technique for the recognition of Gurmukhi characters. They have considered 35 Gurmukhi characters in their work and for each character, they have considered 200 samples taken from distinct writers. These samples are first pre-processed and normalized to a size of 30*30. A cross-validation accuracy of 94.3% has been attained when 5-fold cross-validation approach in LibSVM with RBF kernel is used.

Jumanal and Holi (2013) have proposed a methodology for recognizing English characters, based on spatial and temporal features using genetic algorithm. They have considered a data set of 5200 samples, both lowercase and uppercase characters. Out of which 3120 samples are used to train the system and 2080 samples are used for testing the system. Authors have attained an overall recognition accuracy of 81.3%.

Kumar and Rao (2013) have proposed a system for the recognition of online handwritten Telugu characters using SVM. In the system proposed by them, first of all they pre-classified the Telugu strokes into four categories, namely, main stroke, baseline auxiliary stroke, top stroke and bottom stroke. After that various feature extraction techniques have been employed for best results and finally stroke recognition is done using SVM. For main stroke, they have achieved an overall stroke recognition accuracy of 96.7%.

Hamdulla *et al.* (2014) have proposed Modified Centre Distance Feature (MCDF) for Uyghur characters. They have presented three implementations of MCDF (MCDF-2, MCDF-4 and MCDF-8). Along with MCDF, they have also considered some low level features such as stroke number feature, shape feature and additional part's location feature. They have taken 12800 distinct samples of Uyghur characters. Out of 12800 samples, 8960 samples are used to train the system and remaining 3840 samples are used to test the system. Authors have achieved a recognition rate of 98.8% for 32 isolated forms of Uyghur characters.

Joseph and Hameed (2014) have proposed an experimental technique for recognizing the Malayalam handwritten character using LibSVM. They have considered 160 samples of Malayalam vowels with different styles. They have achieved a good recognition rate of above 90.0% by using LibSVM classifier with polynomial kernel and Gaussian kernel.

CHAPTER 3

PROBLEM STATEMENT

It has been seen that much work has not been accomplished in the field of recognition of online handwritten Gurmukhi script, in general and extraction of features for Gurmukhi script recognition, in particular. Efficiency of OHWR system highly relies on the features which are considered as input to a classifier and if suitable features are extracted then it may reduce the complexity of recognition process and high recognition accuracy can be achieved.

The present work focused on the feature extraction and recognition of Gurmukhi strokes. In this work, feature extraction technique has been employed on the preprocessed strokes of Gurmukhi and then recognition of strokes is done. We have employed a DFT based feature extraction technique and recognized the Gurmukhi strokes using LibSVM, RBF kernel of LibSVM has been explored in this work.

CHAPTER 4

DFT BASED FEATURE EXTRACTION TECHNIQUE FOR RECOGNITION OF ONLINE HANDWRITTEN GURMUKHI STROKES

In the present study, we have applied feature extraction technique on the preprocessed Gurmukhi strokes and after that recognition of strokes is done with the help of LibSVM classifier. Development stages of online handwritten stroke recognition system are shown in Figure 4.1. Section 4.1 illustrates the preprocessing of input Gurmukhi strokes. Section 4.2 demonstrates the DFT based feature extraction and Section 4.3 discusses recognition of Gurmukhi strokes.

4.1 Preprocessing

In online handwritten character recognition system, preprocessing is the first and very important stage. Preprocessing of input data is done to eliminate distortions present in the input data which may occur due to software or hardware limitations. These distortions include missing points, irregular size of input data and uneven separations between points at adjoining position. Preprocessing incorporates mainly four steps that are discussed in (Sharma *et al.*, 2009), and are described as follows for the purpose of completeness.

4.1.1 Size normalization and centering

Writing style of every writer is varied. Some writers write a character large in size and some writers write the same character in small size. So, in order to make all the inputs uniform, size normalization is a necessary step. Centering of input data is also needed when the writer writes along the boundary of writing device. In this work, preprocessed samples that are used for experiment are normalized to a size of 300×300 pixels (Sharma *et al.*, 2009).

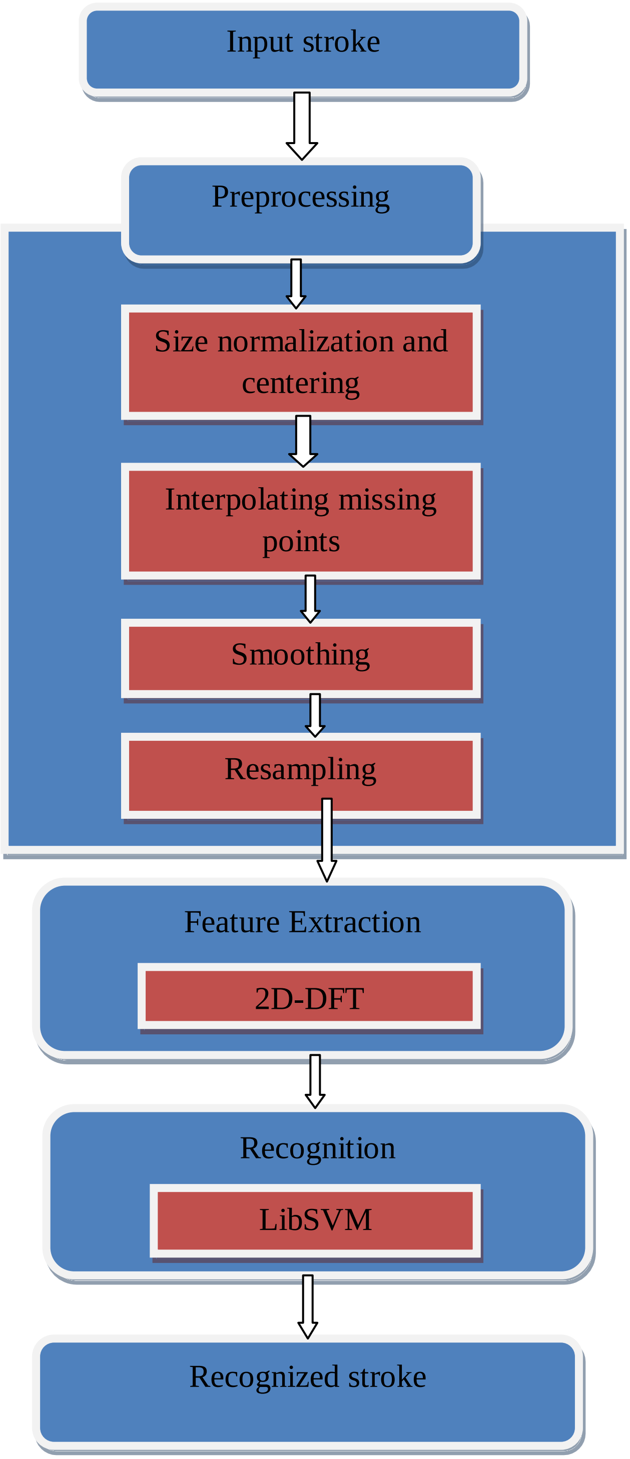


Figure 4.1: Development stages in online handwritten stroke recognition system.

4.1.2 Interpolating missing points

The number of x, y coordinates collected on writing pad totally depends upon the writing speed of user. When user writes with high speed on writing device there may be a possibility of missing some points. Bezier curve method is used to interpolate these missing points (Sharma *et al.*, 2009).

4.1.3 Smoothing

Smoothing is done to eliminate jitter in handwriting. Jitter is removed by changing every point of the stroke with mean estimation of *t* neighbors and the angle subtended at t^{th} position at each end. Mean estimation of two neighbors (*i.e.*, *t* = 2) has been considered in this work (Sharma *et al.*, 2009)

4.1.4 Resampling of points

In resampling, adjacent points of the stroke are placed at equal distances. We have considered 64 equidistant points of the input stroke (Sharma *et al.*, 2009). These resampled points have further been processed for extracting DFT features.

One of the preprocessed samples of character 'ਹ' *hāhā* is shown in Figure 4.2.

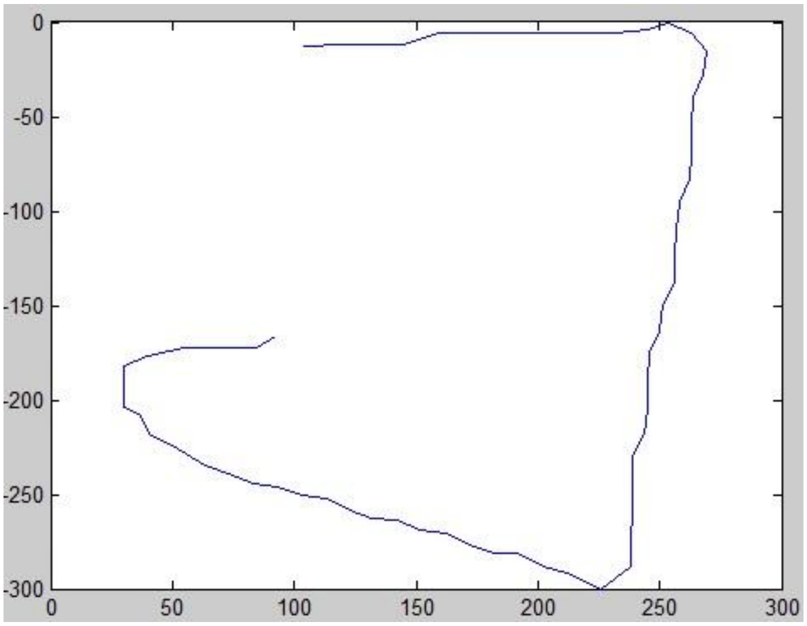


Figure 4.2: Preprocessed Gurmukhi character *hāhā*.

4.2 Feature Extraction

Feature extraction is one of the significant phases in online handwritten character recognition. Selection of a feature extraction technique is an important task. Efficiency of OHWR system highly relies on the features which are considered as input to a classifier. There is no standard strategy for extracting features. Features that provide good results for one script may not provide good results for other scripts.

Features are of two types: one is low level features and other is high level features. Low level features include slope, curliness, width *etc*. and high level features include head line, straight line, dot, crossing *etc*. (Sharma *et al.*, 2008).

In the present study, we have applied 2D-DFT on the preprocessed points of the input stroke. After applying 2D-DFT, we get complex numbers as output. We have used real part coefficients of these complex numbers as features and stored in a file, called feature file and this feature file is taken as input to the classifier. Equation 1 defines the 2D-DFT when applied on f (Gonzalez, 2006).

$$F[k,l] = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f[m,n]\,.\,e^{-2\pi j\left(\frac{km}{M}+\frac{ln}{N}\right)} \tag{1}$$

where

$f[m,n]$ is a discrete input signal,

$F[k,l]$ is the 2D-DFT of $f[m,n]$,

M and N are the number of samples in x and y directions respectively,

$0 \leq m,k \leq M-1$,

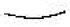
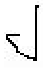
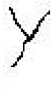
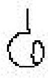
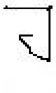
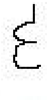
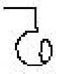
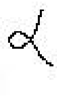
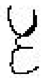
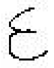
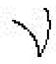
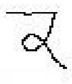
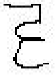
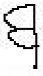
$0 \leq n,l \leq N-1$.

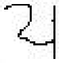
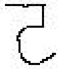
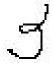
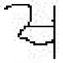
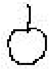
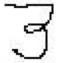
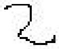
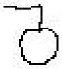
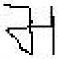
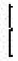
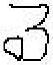
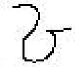
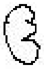
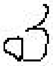
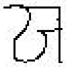
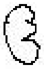
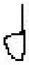
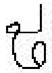
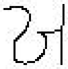
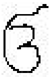
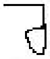
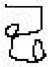
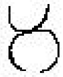
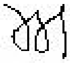
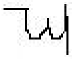
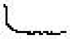
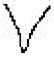
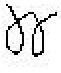
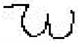
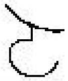
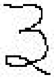
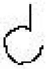
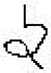
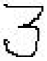
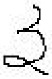
4.3 Recognition of Gurmukhi Strokes

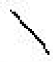
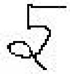
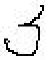
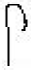
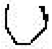
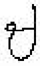
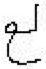
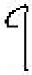
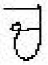
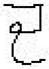
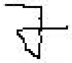
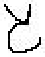
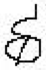
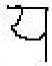
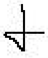
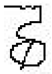
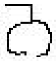
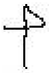
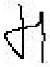
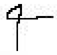
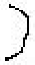
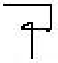
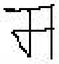
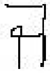
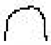
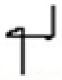
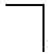
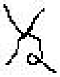
For recognizing online handwritten Gurmukhi strokes, we have used SVM classifier for the classification of strokes. In SVM, supervised learning algorithm is used for classification. Here, each data item is plotted as a point in *n*-dimensional space where *n* is the number of features. In SVM, classification is done by estimating the hyper-plane which differentiates the two classes. SVM employs kernel functions, namely, linear kernel, polynomial kernel, RBF kernel and sigmoid (hyperbolic tangent) kernel. A classification task generally includes division of data into two sets that are training and testing sets. Based on the training data, a model file is generated. The generated model file is further used for classifying the test data.

A character in Gurmukhi script can be written as a combination of one or more strokes, depending upon the user. In this study, a class id is associated with each stroke that can form a Gurmukhi character. Class ids associated with each stroke are shown in Table 4.1. We have considered 86 stroke classes in my work.

Table 4.1: Stroke ids and corresponding stroke shapes.

Stroke id	Stroke Shape	Stroke id	Stroke Shape	Stroke id	Stroke Shape	Stroke id	Stroke Shape
101		155		177		200	
106		156		179		201	
122		157		180		202	
123		158		181		203	

124		159		182		204	
125		160		183		205	
126		161		184		206	
128		162		185		207	
141		163		186		208	
142		164		187		209	
143		165		188		210	
144		166		189		211	
145		167		190		212	
146		168		191		214	

147		169		192		215	
148		170		193		216	
149		171		194		217	
150		172		195		218	
151		173		196		220	
152		174		197		222	
153		175		198		223	
154		176					

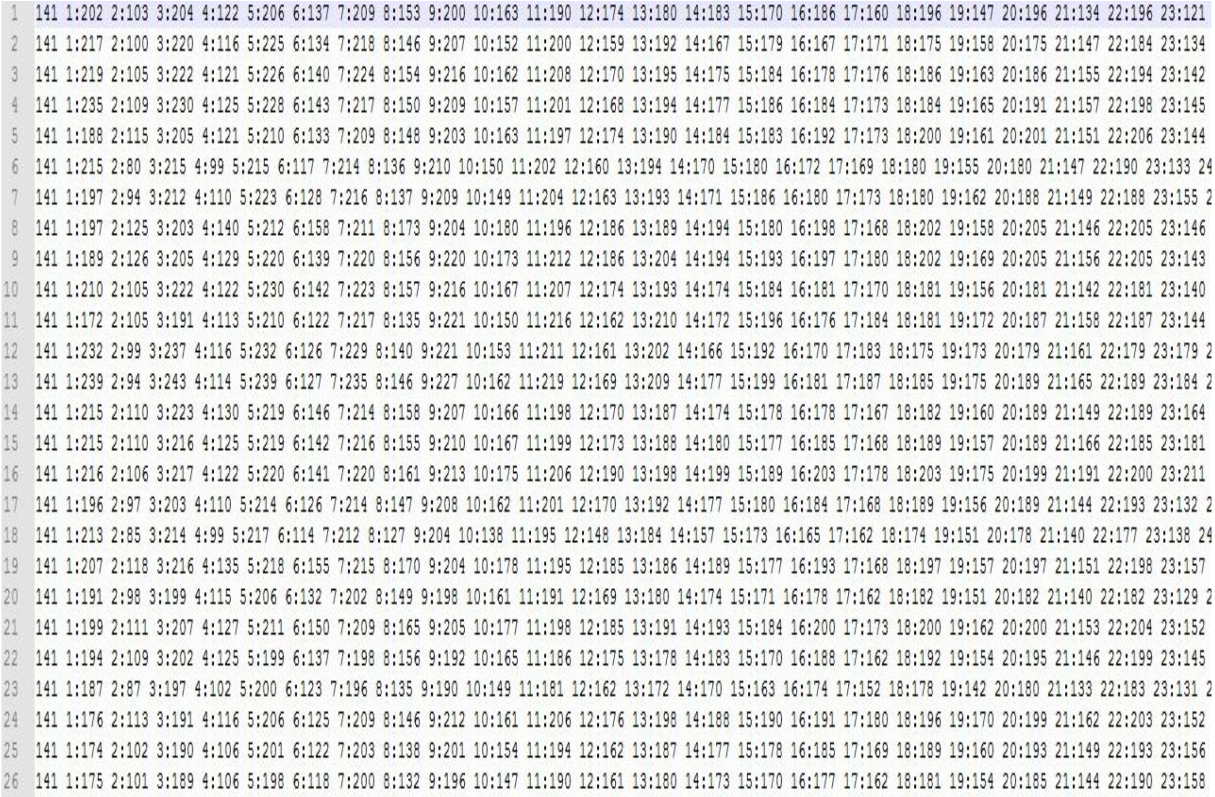
For recognition of Gurmukhi strokes, we have applied DFT based feature extraction technique on the preprocessed strokes. Each Preprocessed stroke has 64 points, *i.e.*, 64 *x*-coordinates and 64 *y*-coordinates, so a total of 128 points are there. These 128 points are stored in a feature file. This feature file is taken as input to the SVM classifier. The format of this feature file is given below:

<Class id> <index 1> : <value 1> <index 2> : <value 2> …

So, each preprocessed stroke is stored as follows:

Stroke id 1: *x*-coordinate 2: *y*-coordinate 3: *x*-coordinate 4: *y*-coordinate … 127: *x*-coordinate 128: *y*-coordinate.

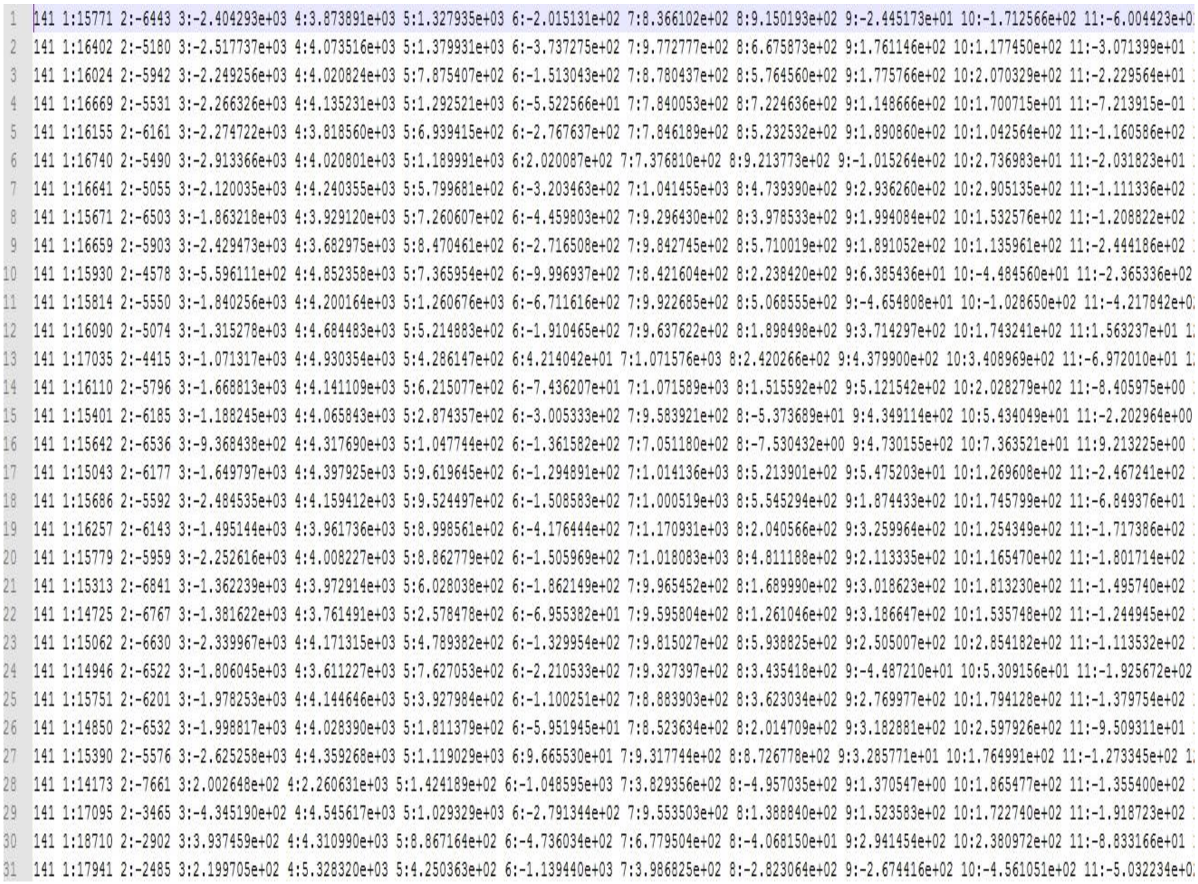
These preprocessed points in the SVM format is shown in Figure 4.3. We have read these preprocessed points from the feature file and then applied 2D-DFT on these preprocessed points. After applying 2D-DFT, the output obtained is used as features and stored into feature file in the above mentioned format. Feature file containing real part coefficients of 2D-DFT as features is shown in Figure 4.4.



```
141 1:202 2:103 3:204 4:122 5:206 6:137 7:209 8:153 9:200 10:163 11:190 12:174 13:180 14:183 15:170 16:186 17:160 18:196 19:147 20:196 21:134 22:196 23:121
141 1:217 2:100 3:220 4:116 5:225 6:134 7:218 8:146 9:207 10:152 11:200 12:159 13:192 14:167 15:179 16:167 17:171 18:175 19:158 20:175 21:147 22:184 23:134
141 1:219 2:105 3:222 4:121 5:226 6:140 7:224 8:154 9:216 10:162 11:208 12:170 13:195 14:175 15:184 16:178 17:176 18:186 19:163 20:186 21:155 22:194 23:142
141 1:235 2:109 3:230 4:125 5:228 6:143 7:217 8:150 9:209 10:157 11:201 12:168 13:194 14:177 15:186 16:184 17:173 18:184 19:165 20:191 21:157 22:198 23:145
141 1:188 2:115 3:205 4:121 5:210 6:133 7:209 8:148 9:203 10:163 11:197 12:174 13:190 14:184 15:183 16:192 17:173 18:200 19:161 20:201 21:151 22:206 23:144
141 1:215 2:80 3:215 4:99 5:215 6:117 7:214 8:136 9:210 10:150 11:202 12:160 13:194 14:170 15:180 16:172 17:169 18:180 19:155 20:180 21:147 22:190 23:133 24
141 1:197 2:94 3:212 4:110 5:223 6:128 7:216 8:137 9:209 10:149 11:204 12:163 13:193 14:171 15:186 16:180 17:173 18:180 19:162 20:188 21:149 22:188 23:155 2
141 1:197 2:125 3:203 4:140 5:212 6:158 7:211 8:173 9:204 10:180 11:196 12:186 13:189 14:194 15:180 16:198 17:168 18:202 19:158 20:205 21:146 22:205 23:146
141 1:189 2:126 3:205 4:129 5:220 6:139 7:220 8:156 9:220 10:173 11:212 12:186 13:204 14:194 15:193 16:197 17:180 18:202 19:169 20:205 21:156 22:205 23:143
141 1:210 2:105 3:222 4:122 5:230 6:142 7:223 8:157 9:216 10:167 11:207 12:174 13:193 14:174 15:184 16:181 17:170 18:181 19:156 20:181 21:142 22:181 23:140
141 1:172 2:105 3:191 4:113 5:210 6:122 7:217 8:135 9:221 10:150 11:216 12:162 13:210 14:172 15:196 16:176 17:184 18:181 19:172 20:187 21:158 22:187 23:144
141 1:232 2:99 3:237 4:116 5:232 6:126 7:229 8:140 9:221 10:153 11:211 12:161 13:202 14:166 15:192 16:170 17:183 18:175 19:173 20:179 21:161 22:179 23:179 2
141 1:239 2:94 3:243 4:114 5:239 6:127 7:235 8:146 9:227 10:162 11:219 12:169 13:209 14:177 15:199 16:181 17:187 18:185 19:175 20:189 21:165 22:189 23:184 2
141 1:215 2:110 3:223 4:130 5:219 6:146 7:214 8:158 9:207 10:166 11:198 12:170 13:187 14:174 15:178 16:178 17:167 18:182 19:160 20:189 21:149 22:189 23:164
141 1:215 2:110 3:216 4:125 5:219 6:142 7:216 8:155 9:210 10:167 11:199 12:173 13:188 14:180 15:177 16:185 17:168 18:189 19:157 20:189 21:166 22:185 23:181
141 1:216 2:106 3:217 4:122 5:220 6:141 7:220 8:161 9:213 10:175 11:206 12:190 13:198 14:199 15:189 16:203 17:178 18:203 19:175 20:199 21:191 22:200 23:211
141 1:196 2:97 3:203 4:110 5:214 6:126 7:214 8:147 9:208 10:162 11:201 12:170 13:192 14:177 15:180 16:184 17:168 18:189 19:156 20:189 21:144 22:193 23:132 2
141 1:213 2:85 3:214 4:99 5:217 6:114 7:212 8:127 9:204 10:138 11:195 12:148 13:184 14:157 15:173 16:165 17:162 18:174 19:151 20:178 21:140 22:177 23:138 24
141 1:207 2:118 3:216 4:135 5:218 6:155 7:215 8:170 9:204 10:178 11:195 12:185 13:186 14:189 15:177 16:193 17:168 18:197 19:157 20:197 21:151 22:198 23:157
141 1:191 2:98 3:199 4:115 5:206 6:132 7:202 8:149 9:198 10:161 11:191 12:169 13:180 14:174 15:171 16:178 17:162 18:182 19:151 20:182 21:140 22:182 23:129 2
141 1:199 2:111 3:207 4:127 5:211 6:150 7:209 8:165 9:205 10:177 11:198 12:185 13:191 14:193 15:184 16:200 17:173 18:200 19:162 20:200 21:153 22:204 23:152
141 1:194 2:109 3:202 4:125 5:199 6:137 7:198 8:156 9:192 10:165 11:186 12:175 13:178 14:183 15:170 16:188 17:162 18:192 19:154 20:195 21:146 22:199 23:145
141 1:187 2:87 3:197 4:102 5:200 6:123 7:196 8:135 9:190 10:149 11:181 12:162 13:172 14:170 15:163 16:174 17:152 18:178 19:142 20:180 21:133 22:183 23:131 2
141 1:176 2:113 3:191 4:116 5:206 6:125 7:209 8:146 9:212 10:161 11:206 12:176 13:198 14:188 15:190 16:191 17:180 18:196 19:170 20:199 21:162 22:203 23:152
141 1:174 2:102 3:190 4:106 5:201 6:122 7:203 8:138 9:201 10:154 11:194 12:162 13:187 14:177 15:178 16:185 17:169 18:189 19:160 20:193 21:149 22:193 23:156
141 1:175 2:101 3:189 4:106 5:198 6:118 7:200 8:132 9:196 10:147 11:190 12:161 13:180 14:173 15:170 16:177 17:162 18:181 19:154 20:185 21:144 22:190 23:158
```

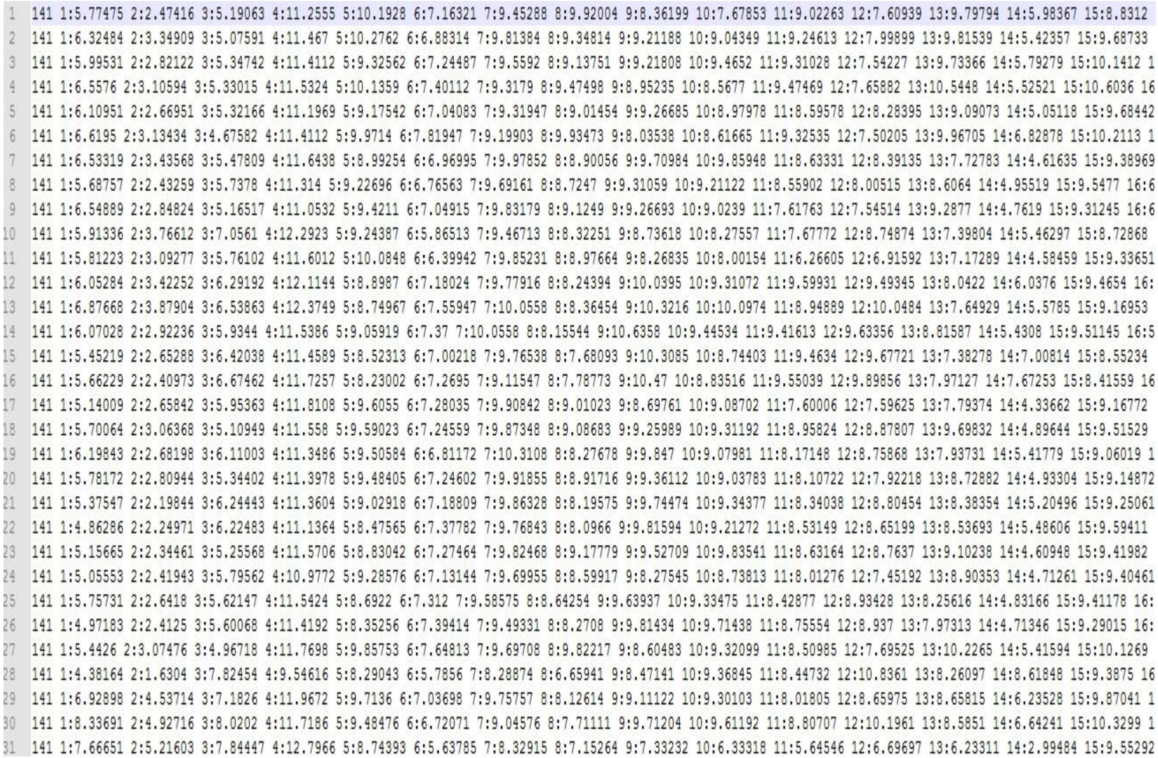
Figure 4.3: Preprocessed points of strokes in SVM format.

Before training the system with the obtained feature file, feature file should be scaled to achieve high recognition accuracy. Scaling can be done with any lower and higher limits, it totally depends upon the input data. In the present study, DFT based feature file is scaled to different ranges that is discussed in next Chapter. DFT based feature file that is scaled from 1 to 14 is shown in Figure 4.5.



```
141 1:15771 2:-6443 3:-2.404293e+03 4:3.873891e+03 5:1.327935e+03 6:-2.015131e+02 7:8.366102e+02 8:9.150193e+02 9:-2.445173e+01 10:-1.712566e+02 11:-6.004423e+0
141 1:16402 2:-5180 3:-2.517737e+03 4:4.073516e+03 5:1.379931e+03 6:-3.737275e+02 7:9.772777e+02 8:6.675873e+02 9:1.761146e+02 10:1.177450e+02 11:-3.071399e+01
141 1:16024 2:-5942 3:-2.249256e+03 4:4.020824e+03 5:7.875407e+02 6:-1.513043e+02 7:8.780437e+02 8:5.764560e+02 9:1.775766e+02 10:2.070329e+02 11:-2.229564e+01
141 1:16669 2:-5531 3:-2.266326e+03 4:4.135231e+03 5:1.292521e+03 6:-5.522566e+01 7:7.840053e+02 8:7.224636e+02 9:1.148666e+02 10:1.700715e+01 11:-7.213915e-01
141 1:16155 2:-6161 3:-2.274722e+03 4:3.818560e+03 5:6.939415e+02 6:-2.767637e+02 7:7.846189e+02 8:5.232532e+02 9:1.890860e+02 10:1.042564e+02 11:-1.160586e+02
141 1:16740 2:-5490 3:-2.913366e+03 4:4.020801e+03 5:1.189991e+03 6:2.020087e+02 7:7.376810e+02 8:9.213773e+02 9:-1.015264e+02 10:2.736983e+01 11:-2.031823e+01
141 1:16641 2:-5055 3:-2.120035e+03 4:4.240355e+03 5:5.799681e+02 6:-3.203463e+02 7:1.041455e+03 8:4.739390e+02 9:2.936260e+02 10:2.905135e+02 11:-1.111336e+02
141 1:15671 2:-6503 3:-1.863218e+03 4:3.929120e+03 5:7.260607e+02 6:-4.459803e+02 7:9.296430e+02 8:3.978533e+02 9:1.994084e+02 10:1.532576e+02 11:-1.208822e+02
141 1:16659 2:-5903 3:-2.429473e+03 4:3.682975e+03 5:8.470461e+02 6:-2.716508e+02 7:9.842745e+02 8:5.710019e+02 9:1.891052e+02 10:1.135961e+02 11:-2.444186e+02
141 1:15930 2:-4578 3:-5.596111e+02 4:4.852358e+03 5:7.365954e+02 6:-9.996937e+02 7:8.421604e+02 8:2.238420e+02 9:6.385436e+01 10:-4.484560e+01 11:-2.365336e+02
141 1:15814 2:-5550 3:-1.840256e+03 4:4.200164e+03 5:1.260676e+03 6:-6.711616e+02 7:9.922685e+02 8:5.068555e+02 9:-4.654808e+01 10:-1.028650e+02 11:-4.217842e+0
141 1:16090 2:-5074 3:-1.315278e+03 4:4.684483e+03 5:5.214883e+02 6:-1.910465e+02 7:9.637622e+02 8:1.898498e+02 9:3.714297e+02 10:1.743241e+02 11:1.563237e+01 1
141 1:17035 2:-4415 3:-1.071317e+03 4:4.930354e+03 5:4.286147e+02 6:4.214042e+01 7:1.071576e+03 8:2.420266e+02 9:4.379900e+02 10:3.408969e+02 11:-6.972010e+01 1
141 1:16110 2:-5796 3:-1.668813e+03 4:4.141109e+03 5:6.215077e+02 6:-7.436207e+01 7:1.071589e+03 8:1.515592e+02 9:5.121542e+02 10:2.028279e+02 11:-8.405975e+00
141 1:15401 2:-6185 3:-1.188245e+03 4:4.065843e+03 5:2.874357e+02 6:-3.005333e+02 7:9.583921e+02 8:-5.373689e+01 9:4.349114e+02 10:5.434049e+01 11:-2.202964e+00
141 1:15642 2:-6536 3:-9.368438e+02 4:4.317690e+03 5:1.047744e+02 6:-1.361582e+02 7:7.051180e+02 8:-7.530432e+00 9:4.730155e+02 10:7.363521e+01 11:9.213225e+00
141 1:15043 2:-6177 3:-1.649797e+03 4:4.397925e+03 5:9.619645e+02 6:-1.294891e+02 7:1.014136e+03 8:5.213901e+02 9:5.475203e+01 10:1.269608e+02 11:-2.467241e+02
141 1:15686 2:-5592 3:-2.484535e+03 4:4.159412e+03 5:9.524497e+02 6:-1.508583e+02 7:1.000519e+03 8:5.545294e+02 9:1.874433e+02 10:1.745799e+02 11:-6.849376e+01
141 1:16257 2:-6143 3:-1.495144e+03 4:3.961736e+03 5:8.998561e+02 6:-4.176444e+02 7:1.170931e+03 8:2.040566e+02 9:3.259964e+02 10:1.254349e+02 11:-1.717386e+02
141 1:15779 2:-5959 3:-2.252616e+03 4:4.008227e+03 5:8.862779e+02 6:-1.505969e+02 7:1.018083e+03 8:4.811188e+02 9:2.113335e+02 10:1.165470e+02 11:-1.801714e+02
141 1:15313 2:-6841 3:-1.362239e+03 4:3.972914e+03 5:6.028038e+02 6:-1.862149e+02 7:9.965452e+02 8:1.689990e+02 9:3.018623e+02 10:1.813230e+02 11:-1.495740e+02
141 1:14725 2:-6767 3:-1.381622e+03 4:3.761491e+03 5:2.578478e+02 6:-6.955382e+01 7:9.595804e+02 8:1.261046e+02 9:3.186647e+02 10:1.535748e+02 11:-1.244945e+02
141 1:15062 2:-6630 3:-2.339967e+03 4:4.171315e+03 5:4.789382e+02 6:-1.329954e+02 7:9.815027e+02 8:5.938825e+02 9:2.505007e+02 10:2.854182e+02 11:-1.113532e+02
141 1:14946 2:-6522 3:-1.806045e+03 4:3.611227e+03 5:7.627053e+02 6:-2.210533e+02 7:9.327397e+02 8:3.435418e+02 9:-4.487210e+01 10:5.309156e+01 11:-1.925672e+02
141 1:15751 2:-6201 3:-1.978253e+03 4:4.144646e+03 5:3.927984e+02 6:-1.100251e+02 7:8.883903e+02 8:3.623034e+02 9:2.769977e+02 10:1.794128e+02 11:-1.379754e+02
141 1:14850 2:-6532 3:-1.998817e+03 4:4.028390e+03 5:1.811379e+02 6:-5.951945e+01 7:8.523634e+02 8:2.014709e+02 9:3.182881e+02 10:2.597926e+02 11:-9.509311e+01
141 1:15390 2:-5576 3:-2.625258e+03 4:4.359268e+03 5:1.119029e+03 6:9.665530e+01 7:9.317744e+02 8:8.726778e+02 9:3.285771e+01 10:1.764991e+02 11:-1.273345e+02 1
141 1:14173 2:-7661 3:2.002648e+02 4:2.260631e+03 5:1.424189e+02 6:-1.048595e+03 7:3.829356e+02 8:-4.957035e+02 9:1.370547e+00 10:1.865477e+02 11:-1.355400e+02
141 1:17095 2:-3465 3:-4.345190e+02 4:4.545617e+03 5:1.029329e+03 6:-2.791344e+02 7:9.553503e+02 8:1.388840e+02 9:1.523583e+02 10:1.722740e+02 11:-1.918723e+02
141 1:18710 2:-2902 3:3.937459e+02 4:4.310990e+03 5:8.867164e+02 6:-4.736034e+02 7:6.779504e+02 8:-4.068150e+01 9:2.941454e+02 10:2.380972e+02 11:-8.833166e+01
141 1:17941 2:-2485 3:2.199705e+02 4:5.328320e+03 5:4.250363e+02 6:-1.139440e+03 7:3.986825e+02 8:-2.823064e+02 9:-2.674416e+02 10:-4.561051e+02 11:-5.032234e+0
```

Figure 4.4: DFT based feature file.



```
141 1:5.77475 2:2.47416 3:5.19063 4:11.2555 5:10.1928 6:7.16321 7:9.45288 8:9.92004 9:8.36199 10:7.67853 11:9.02263 12:7.60939 13:9.79794 14:5.98367 15:8.8312
141 1:6.32484 2:3.34909 3:5.07591 4:11.467 5:10.2762 6:6.88314 7:9.81384 8:9.34814 9:9.21188 10:9.04349 11:9.24613 12:7.99899 13:9.81539 14:5.42357 15:9.68733
141 1:5.99531 2:2.82122 3:5.34742 4:11.4112 5:9.32562 6:7.24487 7:9.5592 8:9.13751 9:9.21808 10:9.4652 11:9.31028 12:7.54227 13:9.73366 14:5.79279 15:10.1412 1
141 1:6.5576 2:3.10594 3:5.33015 4:11.5324 5:10.1359 6:7.40112 7:9.3179 8:9.47498 9:8.95235 10:8.5677 11:9.47469 12:7.65882 13:10.5448 14:5.52521 15:10.6036 16
141 1:6.10951 2:2.66951 3:5.32166 4:11.1969 5:9.17542 6:7.04083 7:9.31947 8:9.01454 9:9.26685 10:8.97978 11:8.59578 12:8.28395 13:9.09073 14:5.05118 15:9.68442
141 1:6.6195 2:3.13434 3:4.67582 4:11.4112 5:9.9714 6:7.81947 7:9.19903 8:9.93473 9:8.03538 10:8.61665 11:9.32535 12:7.50205 13:9.96705 14:6.82878 15:10.2113 1
141 1:6.53319 2:3.43568 3:5.47809 4:11.6438 5:8.99254 6:6.96995 7:9.97852 8:8.90056 9:9.70984 10:9.85948 11:8.63331 12:8.39135 13:7.72783 14:4.61635 15:9.38969
141 1:5.68757 2:2.43259 3:5.7378 4:11.314 5:9.22696 6:6.76563 7:9.69161 8:8.7247 9:9.31059 10:9.21122 11:8.55902 12:8.00515 13:8.6064 14:4.95519 15:9.5477 16:6
141 1:6.54889 2:2.84824 3:5.16517 4:11.0532 5:9.4211 6:7.04915 7:9.83179 8:9.1249 9:9.26693 10:9.0239 11:7.61763 12:7.54514 13:9.2877 14:4.7619 15:9.31245 16:6
141 1:5.91336 2:3.76612 3:7.0561 4:12.2923 5:9.24387 6:5.86513 7:9.46713 8:8.32251 9:8.73618 10:8.27557 11:7.67772 12:8.74874 13:7.39804 14:5.46297 15:8.72868
141 1:5.81223 2:3.09277 3:5.76102 4:11.6012 5:10.0848 6:6.39942 7:9.85231 8:8.97664 9:8.26835 10:8.00154 11:6.26605 12:6.91592 13:7.17289 14:4.58459 15:9.33651
141 1:6.05284 2:3.42252 3:6.29192 4:12.1144 5:8.8987 6:7.18024 7:9.77916 8:8.24394 9:10.0395 10:9.31072 11:9.59931 12:9.49345 13:8.0422 14:6.0376 15:9.4654 16:
141 1:6.87668 2:3.87904 3:6.53863 4:12.3749 5:8.74967 6:7.55947 7:10.0558 8:8.36454 9:10.3216 10:10.0974 11:8.94889 12:10.0484 13:7.64929 14:5.5785 15:9.16953
141 1:6.07028 2:2.92236 3:5.9344 4:11.5386 5:9.05919 6:7.37 7:10.0558 8:8.15544 9:10.6358 10:9.44534 11:9.41613 12:9.63356 13:8.81587 14:5.4308 15:9.51145 16:5
141 1:5.45219 2:2.65288 3:6.42038 4:11.4589 5:8.52313 6:7.00218 7:9.76538 8:7.68093 9:10.3085 10:8.74403 11:9.4634 12:9.67721 13:7.38278 14:7.00814 15:8.55234
141 1:5.66229 2:2.40973 3:6.67462 4:11.7257 5:8.23002 6:7.2695 7:9.11547 8:7.78773 9:10.47 10:8.83516 11:9.55039 12:9.89856 13:7.97127 14:7.67253 15:8.41559 16
141 1:5.14009 2:2.65842 3:5.95363 4:11.8108 5:9.6055 6:7.28035 7:9.90842 8:9.01023 9:8.69761 10:9.08702 11:7.60006 12:7.59625 13:7.79374 14:4.33662 15:9.16772
141 1:5.70064 2:3.06368 3:5.10949 4:11.558 5:9.59023 6:7.24559 7:9.87348 8:9.08683 9:9.25989 10:9.31192 11:8.95824 12:8.87807 13:9.69832 14:4.89644 15:9.51529
141 1:6.19843 2:2.68198 3:6.11003 4:11.3486 5:9.50584 6:6.81172 7:10.3108 8:8.27678 9:9.847 10:9.07981 11:8.17148 12:8.75868 13:7.93731 14:5.41779 15:9.06019 1
141 1:5.78172 2:2.80944 3:5.34402 4:11.3978 5:9.48405 6:7.24602 7:9.91855 8:8.91716 9:9.36112 10:9.03783 11:8.10722 12:7.92218 13:8.72882 14:4.93304 15:9.14872
141 1:5.37547 2:2.19844 3:6.24443 4:11.3604 5:9.02918 6:7.18809 7:9.86328 8:8.19575 9:9.74474 10:9.34377 11:8.34038 12:8.80454 13:8.38354 14:5.20496 15:9.25061
141 1:4.86286 2:2.24971 3:6.22483 4:11.1364 5:8.47565 6:7.37782 7:9.76843 8:8.0966 9:9.81594 10:9.21272 11:8.53149 12:8.65199 13:8.53693 14:5.48606 15:9.59411
141 1:5.15665 2:2.34461 3:5.25568 4:11.5706 5:8.83042 6:7.27464 7:9.82468 8:9.17779 9:9.52709 10:9.83541 11:8.63164 12:8.7637 13:9.10238 14:4.60948 15:9.41982
141 1:5.05553 2:2.41943 3:5.79562 4:10.9772 5:9.28576 6:7.13144 7:9.69955 8:8.59917 9:8.27545 10:8.73813 11:8.01276 12:7.45192 13:8.90353 14:4.71261 15:9.40461
141 1:5.75731 2:2.6418 3:5.62147 4:11.5424 5:8.6922 6:7.312 7:9.58575 8:8.64254 9:9.63937 10:9.33475 11:8.42877 12:8.93428 13:8.25616 14:4.83166 15:9.41178 16:
141 1:4.97183 2:2.4125 3:5.60068 4:11.4192 5:8.35256 6:7.39414 7:9.49331 8:8.2708 9:9.81434 10:9.71438 11:8.75554 12:8.937 13:7.97313 14:4.71346 15:9.29015 16:
141 1:5.4426 2:3.07476 3:4.96718 4:11.7698 5:9.85753 6:7.64813 7:9.69708 8:9.82217 9:8.60483 10:9.32099 11:8.50985 12:7.69525 13:10.2265 14:5.41594 15:10.1269
141 1:4.38164 2:1.6304 3:7.82454 4:9.54616 5:8.29043 6:5.7856 7:8.28874 8:6.65941 9:8.47141 10:9.36845 11:8.44732 12:10.8361 13:8.26097 14:8.61848 15:9.3875 16
141 1:6.92898 2:4.53714 3:7.1826 4:11.9672 5:9.7136 6:7.03698 7:9.75757 8:8.12614 9:9.11122 10:9.30103 11:8.01805 12:8.65975 13:8.65815 14:6.23528 15:9.87041 1
141 1:8.33691 2:4.92716 3:8.0202 4:11.7186 5:9.48476 6:6.72071 7:9.04576 8:7.71111 9:9.71204 10:9.61192 11:8.80707 12:10.1961 13:8.5851 14:6.64241 15:10.3299 1
141 1:7.66651 2:5.21603 3:7.84447 4:12.7966 5:8.74393 6:5.63785 7:8.32915 8:7.15264 9:7.33232 10:6.33318 11:5.64546 12:6.69697 13:6.23311 14:2.99484 15:9.55292
```

Figure 4.5: Scaled DFT based feature file.

4.3.1 Training and testing using *k*-fold cross-validation

In *k*-fold cross-validation, the training data is divided into *k* parts. Out of *k* parts, *k*-1 parts are used for training and remaining one part is used for testing. 3-fold cross-validation working is shown in Figure 4.6. RBF kernel has been employed for recognition of Gurmukhi strokes in this work. While training the system using RBF kernel, two parameters c and γ are taken into consideration. The c parameter provides tradeoff between decision surface and misclassification of training samples. A lower value of c forms the decision surface smooth and higher value of c tries to classify all training samples correctly. The other parameter γ describes the influence of single training sample. The best values of parameters c and γ are not known in advance for a given problem. So, some sort of parameter search must be done. To find the optimal values of c and γ, we have performed grid search on c and γ. Different sets of (c, γ) values are tried and one set of (c, γ) is chosen which gives the best cross-validation accuracy.

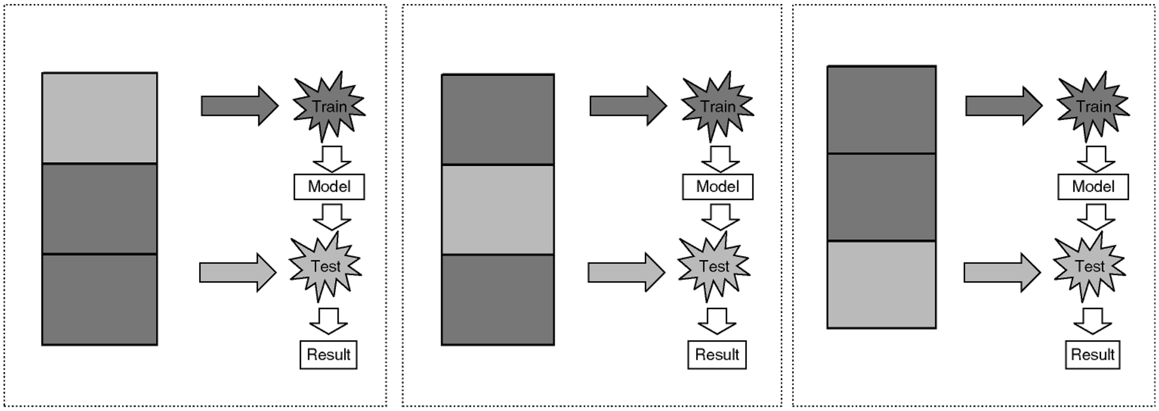


Figure 4.6: Working of 3-fold cross-validation.

In the present work, we have considered 86 classes of Gurmukhi strokes and took 75-100 samples for each class. A total of 8408 strokes of Gurmukhi has been taken as dataset. After applying 2D-DFT on the preprocessed strokes, these strokes are stored into a text file: training.txt. The file training.txt is then scaled and a new file training.txt.scale is generated. This scaled file is passed to a program which gives the optimal values of c and γ on a given fold value and provides the corresponding cross-validation accuracy obtained on c and γ values.

CHAPTER 5

EXPERIMENTAL RESULTS

As discussed in Chapter 4, we have applied 2D-DFT on the preprocessed strokes for recognizing online handwritten Gurmukhi strokes. Present study focuses on recognition of 35 Gurmukhi characters and 10 Gurmukhi *matras*. 86 stroke classes of Gurmukhi are considered, out of which 75 classes are of Gurmukhi characters and 11 classes are of Gurmukhi *matras* and for each class we have taken 75-100 variations. A total of 8408 strokes of Gurmukhi has been considered as data set. We have applied 2D-DFT on the preprocessed strokes. After applying 2D-DFT, we get complex numbers as output. We have performed various experiments by using real part and imaginary part coefficients of complex numbers as features separately. Experimented by scaling feature file to three different ranges has also been performed in this work.

LibSVM classifier has been used for recognition of Gurmukhi strokes. RBF kernel of LibSVM has been employed in this work. We have performed experiments with *k*-fold cross-validation. The advantage of using *k*-fold cross-validation is that it can help in avoiding over-fitting problem. RBF kernel has two parameters *c* and *γ*. Initially, the value of parameters *c* and *γ* are fixed manually and we have to try a lot of combinations of *c* and *γ*. This approach is very time consuming. So, we have used a program based on python script language which gives the optimal values of *c* and *γ* for RBF kernel and corresponding cross-validation accuracy on a given fold value. The value of *γ* varies from 2^{-15} to 2^{3} and value of *c* varies from 2^{-5} to 2^{15}. Six experiments have been conducted in this work that have been discussed further in this Chapter.

5.1 Experiment 1

Experiment 1 has been performed for recognition of Gurmukhi strokes. 86 stroke classes have been considered in this experiment. For each class, 75-100 samples are taken. There are a total of 8408 stroke samples in the dataset. Real part coefficients of complex numbers, obtained as output after applying DFT is used as features and the feature file that is taken as input to

LibSVM is scaled from 1 to 14. The values of parameters *c*, *γ* and cross-validation accuracy obtained from python program for Gurmukhi strokes on different values of fold is shown in Table 5.1.

Table 5.1: *k*-fold cross-validation accuracy of Gurmukhi strokes using real part coefficients as features and scaling feature file from 1 to 14.

***k*-fold**	**Penalty factor (*c*)**	**Gamma (*γ*)**	**Cross-validation accuracy (%)**
2-fold	8	0.00781	89.6
3-fold	8	0.00781	90.3
4-fold	8	0.00781	90.8
5-fold	8	0.00781	91.1
6-fold	8	0.00781	91.0
7-fold	8	0.00781	91.4
8-fold	8	0.00781	91.2
9-fold	8	0.00781	91.4
10-fold	8	0.00781	91.4
11-fold	8	0.00781	91.7
12-fold	8	0.00781	91.5

5.2 Experiment 2

Experiment 2 has been performed for recognition of Gurmukhi strokes on the same dataset of 8408 stroke samples. Also, in this experiment real part coefficients of complex numbers are used as features. But the feature file is scaled from 1 to 9. The values of parameters *c*, *γ* and cross-validation accuracy obtained for Gurmukhi strokes using python program on different values of fold is shown in Table 5.2.

Table 5.2: *k*-fold cross-validation accuracy of Gurmukhi strokes using real part coefficients as features and scaling feature file from 1 to 9.

***k*-fold**	**Penalty factor (*c*)**	**Gamma(*γ*)**	**Cross-validation accuracy (%)**
2-fold	8	0.03125	89.4
3-fold	8	0.03125	90.4
4-fold	8	0.03125	90.8
5-fold	8	0.03125	91.1
6-fold	8	0.03125	91.0
7-fold	8	0.03125	91.2
8-fold	8	0.03125	91.1
9-fold	8	0.03125	91.3
10-fold	8	0.03125	91.3
11-fold	8	0.03125	91.4
12-fold	8	0.03125	91.4

5.3 Experiment 3

Experiment 3 has also been conducted for recognition of Gurmukhi strokes and on the same dataset but scaling used for feature file is from -5 to 5. Also, in this experiment real part coefficients of complex numbers are used as features. The values of parameters *c*, *γ* and cross-validation accuracy obtained on different values of fold for Gurmukhi strokes using python program is shown in Table 5.3.

Table 5.3: *k*-fold cross-validation accuracy of Gurmukhi strokes using real part coefficients as features and scaling feature file from -5 to 5.

***k*-fold**	**Penalty factor (*c*)**	**Gamma(*γ*)**	**Cross-validation accuracy (%)**
2-fold	8	0.00781	89.4
3-fold	8	0.00781	90.1
4-fold	32	0.00781	90.7
5-fold	8	0.00781	90.9
6-fold	32	0.00781	90.9
7-fold	32	0.00781	91.1
8-fold	8	0.00781	91.1
9-fold	32	0.00781	91.2
10-fold	8	0.00781	91.2
11-fold	8	0.00781	91.3
12-fold	8	0.00781	91.2

5.4 Experiment 4

Experiment 4 has also been performed for the recognition of Gurmukhi strokes and on the same dataset. But in this experiment imaginary part coefficients of complex numbers obtained after applying 2D-DFT are used as features and feature file is scaled from 1 to 14. The values of parameters *c*, *γ* and cross-validation accuracy obtained from python program for Gurmukhi strokes on different values of fold is shown in Table 5.4.

Table 5.4: *k*-fold cross-validation accuracy of Gurmukhi strokes using imaginary part coefficients as features and scaling feature file from 1 to 14.

***k*-fold**	**Penalty factor (*c*)**	**Gamma(*γ*)**	**Cross-validation accuracy (%)**
2-fold	32	0.00781	82.4
3-fold	8	0.00781	83.6
4-fold	8	0.00781	84.1
5-fold	8	0.00781	84.8
6-fold	128	0.00781	84.6
7-fold	8	0.00781	84.9
8-fold	512	0.00781	85.0
9-fold	128	0.00781	85.0
10-fold	8	0.00781	85.0
11-fold	2048	0.00781	85.0
12-fold	32	0.00781	85.0

5.5 Experiment 5

Experiment 5 has been performed for recognition of Gurmukhi strokes on the same dataset of 8408 stroke samples. Also, in this experiment imaginary part coefficients of complex numbers are used as features. But the feature file is scaled from 1 to 9. The values of parameters *c*, *γ* and cross-validation accuracy obtained for Gurmukhi strokes by using python program on different values of fold is shown in Table 5.5.

Table 5.5: *k*-fold cross-validation accuracy of Gurmukhi strokes using imaginary part coefficients as features and scaling feature file from 1 to 9.

***k*-fold**	**Penalty factor (*c*)**	**Gamma(*γ*)**	**Cross-validation accuracy (%)**
2-fold	8	0.00781	82.4
3-fold	32	0.00781	83.4
4-fold	8	0.00781	84.2
5-fold	8	0.00781	85.0
6-fold	8	0.00781	84.6
7-fold	8	0.00781	84.6
8-fold	8	0.00781	84.8
9-fold	8	0.00781	85.0
10-fold	8	0.00781	85.0
11-fold	8	0.00781	85.0
12-fold	8	0.00781	84.8

5.6 Experiment 6

Experiment 6 has also been performed for recognition of Gurmukhi strokes. Also, in this experiment we have used imaginary part coefficients of complex numbers as features. Scaling used in this experiment is different. Feature file is scaled from -5 to 5. Values of *c*, *γ* and cross-validation accuracy for Gurmukhi strokes obtained from python program on different values of fold is shown in Table 5.6.

Table 5.6: *k*-fold cross-validation accuracy of Gurmukhi strokes using imaginary part coefficients as features and scaling feature file from -5 to 5.

***k*-fold**	**Penalty factor (*c*)**	**Gamma(*γ*)**	**Cross-validation accuracy (%)**
2-fold	8	0.00781	82.5
3-fold	8	0.00781	83.7
4-fold	8	0.00781	84.3
5-fold	8	0.00781	85.0
6-fold	512	0.00781	84.6
7-fold	32	0.00781	84.6
8-fold	2048	0.00781	84.9
9-fold	2048	0.00781	84.8
10-fold	8	0.00781	84.9
11-fold	8	0.00781	84.8
12-fold	128	0.00781	84.9

Comparison between *k*-fold cross-validation accuracy obtained by using real part coefficients of 2D-DFT as features and by using imaginary part coefficients of 2D-DFT as features at different scaling is shown in Figures 5.1, 5.2 and 5.3.

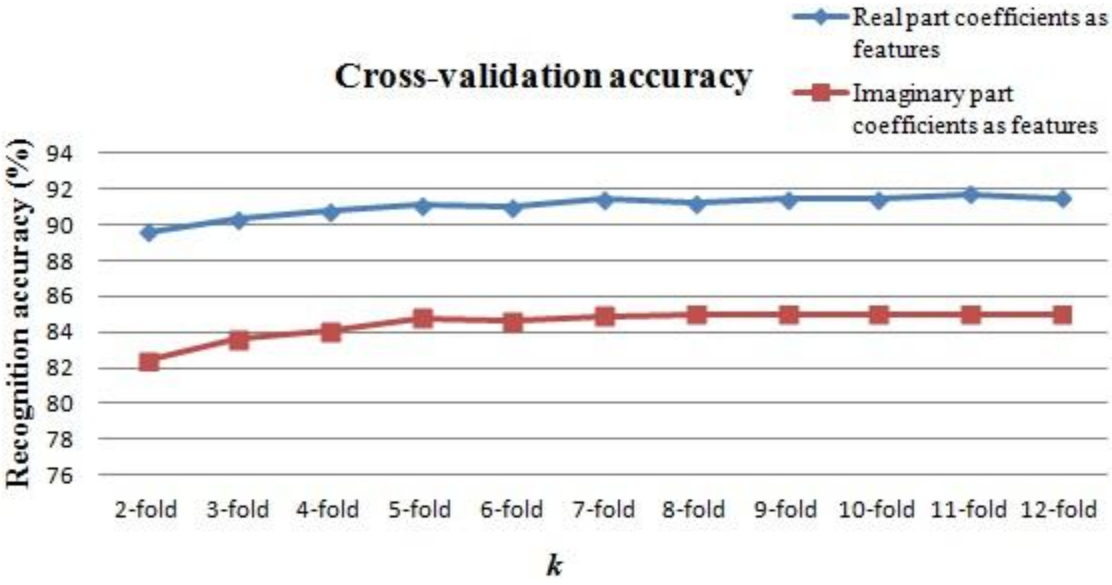


Figure 5.1: Comparison between *k*-fold cross-validation accuracy obtained using real part and imaginary part coefficients as features scaled to 1 to 14.

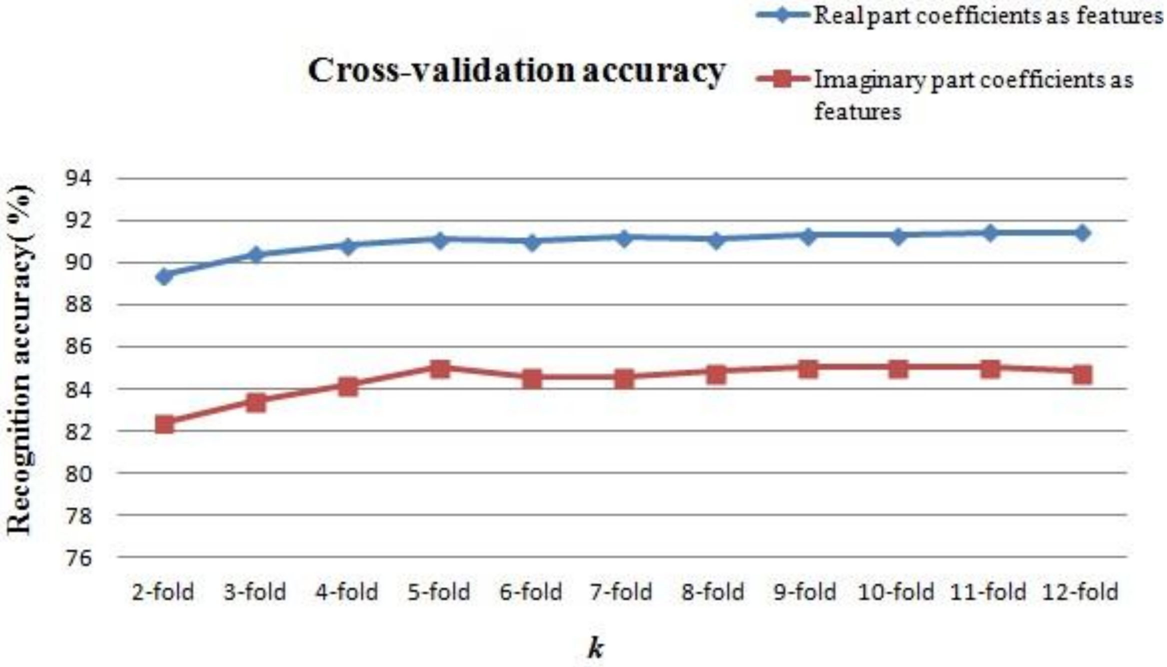


Figure 5.2: Comparison between *k*-fold cross-validation accuracy obtained using real part and imaginary part coefficients as features scaled to 1 to 9.

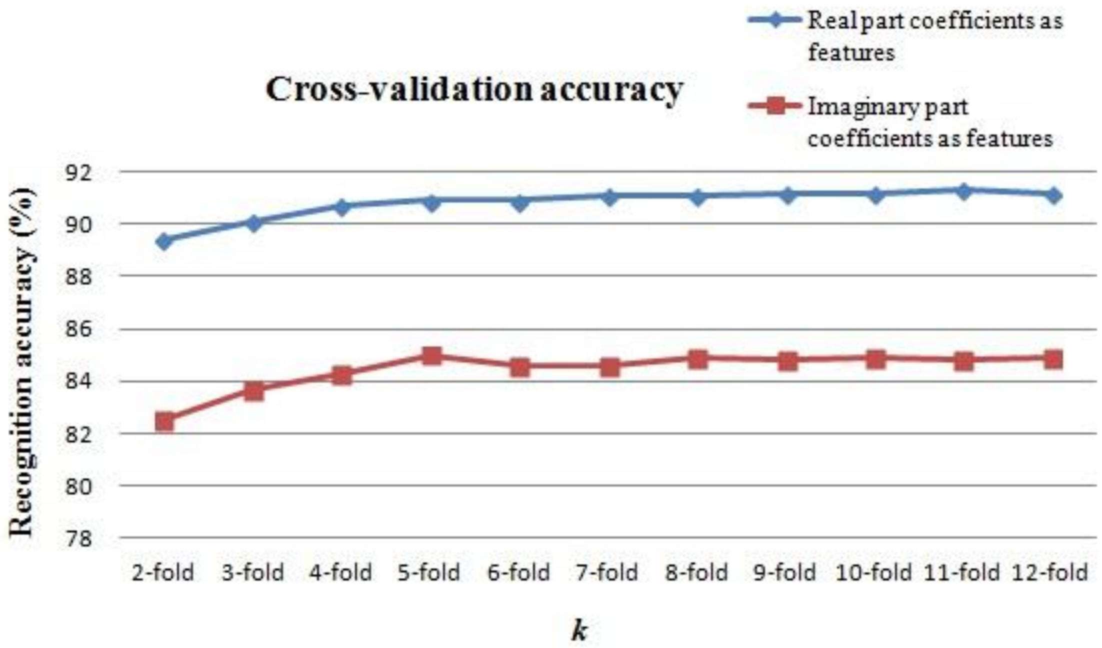
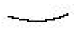
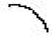
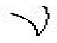
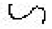
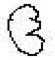
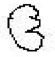
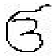
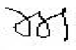


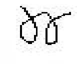
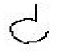
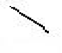
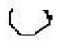
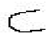
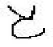
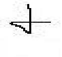
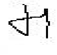
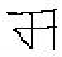
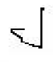
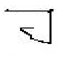
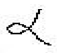
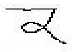
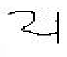
Figure 5.3: Comparison between *k*-fold cross-validation accuracy obtained using real part and imaginary part coefficients as features scaled to -5 to 5.

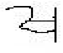
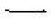
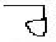
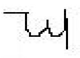
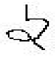
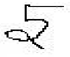
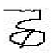
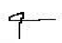
5.7 Testing Accuracy of Each Class

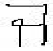
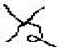
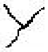
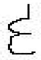
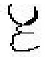
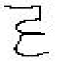
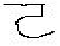
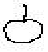
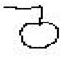
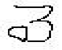
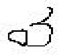
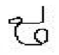
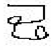
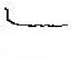
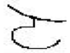
Testing accuracy obtained for 86 strokes classes that have been considered in this work is shown in table 5.7. For each class, 25 samples are taken for testing the engine. A total of 8408 stroke samples of Gurmukhi are taken in this work. Out of which 6258 stroke samples are used to train the system and 2150 stroke samples are used to test the system. Number of stroke samples recognized for each class and confusion of classes with other classes is shown in Table 5.7.

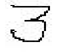
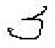
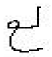
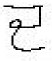
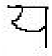
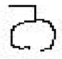
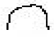
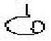
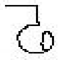
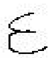
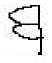
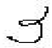
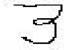
Table 5.7: Testing accuracy obtained for each class.

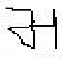
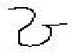
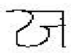
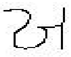
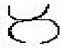
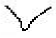
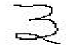
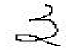
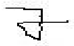
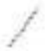
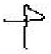
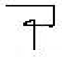
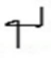
Class Id	Shape	Sample Tested	Recognized Samples	Confusion with Class Id	Accuracy (% age)
101		25	25		100.0
106		25	22	101(1),122(2)	88.0
122		25	21	125(1), 124(2), 207(1)	84.0
123		25	25		100.0
124		25	20	122(5)	80.0
125		25	24	122(1)	96.0
126		25	25		100.0
128		25	23	215(2)	92.0
141		25	25		100.0
142		25	24	141(1)	96.0
143		25	23	142(2)	92.0
144		25	24	166(1)	96.0

145		25	25		100.0
146		25	21	149(4)	84.0
147		25	23	182(2)	92.0
148		25	23	211(2)	92.0
149		25	24	215(1)	96.0
150		25	24	202(1)	96.0
151		25	24	164(1)	96.0
152		25	22	182(2),157(1)	88.0
153		25	6	144(4),159(2),166(2),173(2), 196(3),201(1),203(2),206(3)	24.0
154		25	24	197(1)	96.0
155		25	25		100.0
156		25	24	165(1)	96.0
157		25	23	149(1),223(1)	92.0
158		25	24	157(1)	96.0
159		25	22	154(1),158(1),182(1)	88.0

160		25	25		100.0
161		25	24	147(1)	96.0
162		25	25		100.0
163		25	25		100.0
164		25	25		100.0
165		25	25		100.0
166		25	19	209(1),194(2),176(1),159(1), 144(1)	76.0
167		25	20	145(1),161(1),211(3)	80.0
168		25	18	214(2),162(1),158(1),159(1), 179(1),147(1)	72.0
169		25	22	212(3)	88.0
170		25	25		100.0
171		25	24	184(1)	96.0
172		25	20	200(1),214(4)	80.0
173		25	24	212(1)	96.0
174		25	25		100.0

175		25	23	191(1),205(1)	92.0
176		25	25		100.0
177		25	25		100.0
179		25	23	202(2)	92.0
180		25	25		100.0
181		25	22	182(2),147(1)	88.0
182		25	24	147(1)	96.0
183		25	25		100.0
184		25	25		100.0
185		25	24	212(1)	96.0
186		25	22	185(3)	88.0
187		25	24	203(1)	96.0
188		25	25		100.0
189		25	25		100.0
190		25	23	182(1),203(1)	92.0

191		25	21	205(4)	84.0
192		25	23	162(1),191(1)	92.0
193		25	21	146(4)	84.0
194		25	23	182(1),181(1)	92.0
195		25	24	148(1)	96.0
196		25	12	205(5),166(3),182(3),159(1), 222(1)	48.0
197		25	24	156(1)	96.0
198		25	12	149(7),157(1),194(1),216(2), 182(1),152(1)	48.0
200		25	25		100.0
201		25	23	203(1),188(1)	92.0
202		25	21	190(2),150(2)	84.0
203		25	21	206(1),144(2),201(1)	84.0
204		25	24	191(1)	96.0
205		25	20	191(5)	80.0

206		25	23	159(2)	92.0
207		25	24	189(1)	96.0
208		25	19	148(3),215(2),195(1)	76.0
209		25	14	206(1),181(2),182(1),210(1), 159(1),163(3),144(2)	56.0
210		25	22	203(1),147(2)	88.0
211		25	20	148(5)	80.0
212		25	23	185(2)	92.0
214		25	24	186(1)	96.0
215		25	25		100.0
216		25	25		100.0
217		25	25		100.0
218		25	25		100.0
220		25	24	215(1)	96.0
222		25	24	158(1)	96.0
223		25	25		100.0

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

6.1 Conclusion

This thesis presents an implementation of recognition system for online handwritten Gurmukhi strokes based on 2D-DFT feature extraction technique. The approach used in this thesis and results obtained can be summarized as follows:

- 86 stroke classes of Gurmukhi have been considered in this work and for each class 75-100 samples are taken into consideration. Recognition experiments have been conducted on a data set of 8408 strokes of Gurmukhi.
- The feature file consisted of DFT coefficients, is given as input to the LibSVM classifier.
- Based on experimental results, we can examine best recognition results are obtained when real part coefficients of complex numbers (2D-DFT output) are used as features and feature file is scaled from 1 to 14. Using this approach a recognition accuracy of 91.7% has been attained for Gurmukhi when 11-fold cross-validation approach in LibSVM with RBF kernel is used.

6.2 Future Scope

An interested researcher may implement a few more things to this work:

- One may include features like direction, crossing *etc*. to attain higher recognition accuracy.
- Classes have confusion between them which has degraded the recognition accuracy, so one can work on reducing the class confusion.
- One may extend the work to find character level accuracy for Gurmukhi script and also to word level recognition.

- Improving the recognition accuracy to a higher level is always there as an attraction for a motivated researcher.

REFERENCES

[1] Addakiri K. and Bahaj M., "On-line Handwritten Arabic Character Recognition using Artificial Neural Network," *International Journal on Computer Applications*, vol. 55, no. 13, 2012, pp. 42-46.

[2] Ahmed H. and Azeem S.A., "On-line Arabic Handwriting Recogntion System based on HMM," in *International Conference on Document Analysis and Recognition*, 2011, pp. 1324-1328.

[3] Alijila B. and Kwaik K., "Online Isolated Arabic Handwritten Character Recognition Using Neural Network," *International Arab Journal of Information Technology*, vol. 9, no. 4, 2012, pp. 343-351.

[4] Bahlmann C., Haasdonk B. and Burkhardt H., "On-line Handwriting Recognition with Support Vector Machines- A Kernel Approach," in *Proceedings of the Eighth International Workshop on Frontiers in Handwriting Recognition*, 2002.

[5] Bahri H., "Teach Yourself Panjabi," in *Publication Bureau*, Punjabi University, Patiala, 2011.

[6] Beigi H.S.M. and Watson T.J., "An overview of handwriting recognition." in *Proeedings of The 1st Annual Conference on Technological Advancements*, 1993.

[7] Cho S.J. and Kim J.H., "Bayesian Network Modeling of Hangul Characters for On-line Handwriting Recognition," in *Proceedings of the Seventh International conference on Document Analysis and Recognition*, 2003.

[8] Fink G.A., Vajda S., Bhattacharya U., Parui K. and Chaudhuri B.B. "Online Bangla Word Recognition Using Sub-Stroke Level Features and Hidden Markov Models," in *Proceedings of 12th International Conference on Frontiers in Handwriting Recognition*, 2010.

[9] Ghods V. and Kabir E., "Feature Extraction for Online Farsi Characters," in *Proceedings of 12th International Conference on Frontiers in Handwriting Recognition*, 2010.

[10] Ghosh R. and Ghosh M., "An Intelligent Offline Handwriting Recognition System Using Evolutionary Neural Learning Algorithm and Rule Based Over Segmented Data Points," in *Journal of Research and Practice in Information Technology*, vol. 37, no. 1, 2005.

[11] Gonzalez R.C. and Woods R.E., "Digital Image Processing," in *Prentice-Hall,* New Jersey, USA, 2006.

[12] Gorgevik D. and Cakmakov D., "An Efficient Three-Stage Classifier for Handwriiten Digit Recognition," in *Proceedings of the 17th International Conference on Pattern Recognition*, 2004.

[13] Gupta N., Gupta M. and Agrawal R., "Preprocessing of Gurmukhi Strokes in Online Handwriting Recognition,"in *Proceedings of 3rd International Conference on Information Technology and Artificial Intelligence,* Singapore, vol. 56, no. 30, 2012.

[14] Hamdulla A., Simayi W., Ibrayim M. and Tursun D., "On-line Uyghur Handwritten Character Recognition Technology Based on Modified Center Distance Feature," *International Journal of Signal Processing and Pattern Recognition*, vol. 7, no. 5, 2014, pp. 409-424.

[15] Joseph S.M. and Hameed A., "Online Handwritten Malayalam Character Recognition using LIBSVM in Matlab," in *National Conference on Communication, Signal Processing and Networking*, 2014, pp. 1-5.

[16] Jumanal S. and Holi G., "On-line Handwritten English Character Recognition Using Genetic Algorithm," *International Journal of Computer Trends and Technology*, vol. 6, no. 6, 2013.

[17] Kumar K.V. and Rao R.R., "Online Handwriiten Characetr Recognition for Telugu Language Using Support Vector Machine," *International Journal of Engineering and Advanced Technology*, vol. 3, no. 2, 2013.

[18] Kunwar R., K.S. and Ramakrishnan A.G., "Online handwritten Kannada word recognizer with unrestricted vocabulary," *Proceedings of 12th International Conference on Frontiers in Handwriting Recognition*, 2010.

[19] Parui S.K., Guin K., Bhattacharya U. and Chaudhuri B.B., "Online Handwritten Bangla Character Recognition Using HMM," in *Proceedings of International Conference on Pattern Recognition*, 2008.

[20] Prasad G.K., Khan I., Chanukotimath N. R. and Khan F., "On-line Handwritten Character Recogntion for Kannada using Principal Component Analysis Approach," in *World Congress on Information and Communication Technologies*, 2012, pp. 675-678.

[21] Prasad G.K., Khan I. and Chanukotimath N., "On-line Hindi Handwritten Character Recogntion for Mobile Devices," in *International Conference on Advances in Computing, Communications and Informatics,* Chennai, India, 2012, pp. 1074-1078.

[22] Prasanth L., Babu V.J., Sharma R.R., Rao G.V.P. and M. D., "Elastic Matching of Online Handwritten Tamil and Telugu Scripts Using Local Features," in *Proceedings of International Conference on Document Analysis and Recognition*, 2007.

[23] Sharma A., Kumar R. and Sharma R.K., "Online Handwritten Gurmukhi Character Recognition Using Elastic Matching," in *Congress on Image and Signal Processing*, vol. 2, 2008, pp. 391-396.

[24] Sharma A., Kumar R. and Sharma R.K., "Rearrangement of Recognized Strokes in Online Handwritten Gurmukhi Words Recognition," in *Proceedings of 10th International Conference on Document Analysis and Recognition*, 2009.

[25] Sharma A., Kumar R. and Sharma R.K. (2009). "Online Handwritten Gurmukhi Character Recognition," Ph.D Thesis, Thapar University [Online]. Available:http://dspace.thapar.edu:8080/dspace/bitstream/10266/1057/3/Thesis_AnujSharma_SMCA_9041451.pdf.

[26] Singh S., Aggarwal A. and Dhir R., "Use of Gabor Filters for Recognition of Handwritten Gurmukhi Characters," in *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 2, no. 5, 2012.

[27] Sundaram S. and Ramakrishnan A.G., "A Novel Hierarchical Classification Scheme for Online Tamil Character Recognition," in *Proceedings of International Conference on Document Analysis and Recognition*, 2007.

[28] Vamvakas G., Gatos B. and Perantonis S.J., "A Novel Feature Extraction and Classification Methodology for the Recognition of Historical Documents," in *proceedings of 10th International Conference on Document Analysis and Recognition*, 2009.

[29] Vinciarelli A. and Perrone M., "Combining Online and Offline Handwriting Recognition," in *Proceedings of the Seventh International conference on Document Analysis and Recognition*, 2003.

APPENDIX A

PUBLICATION

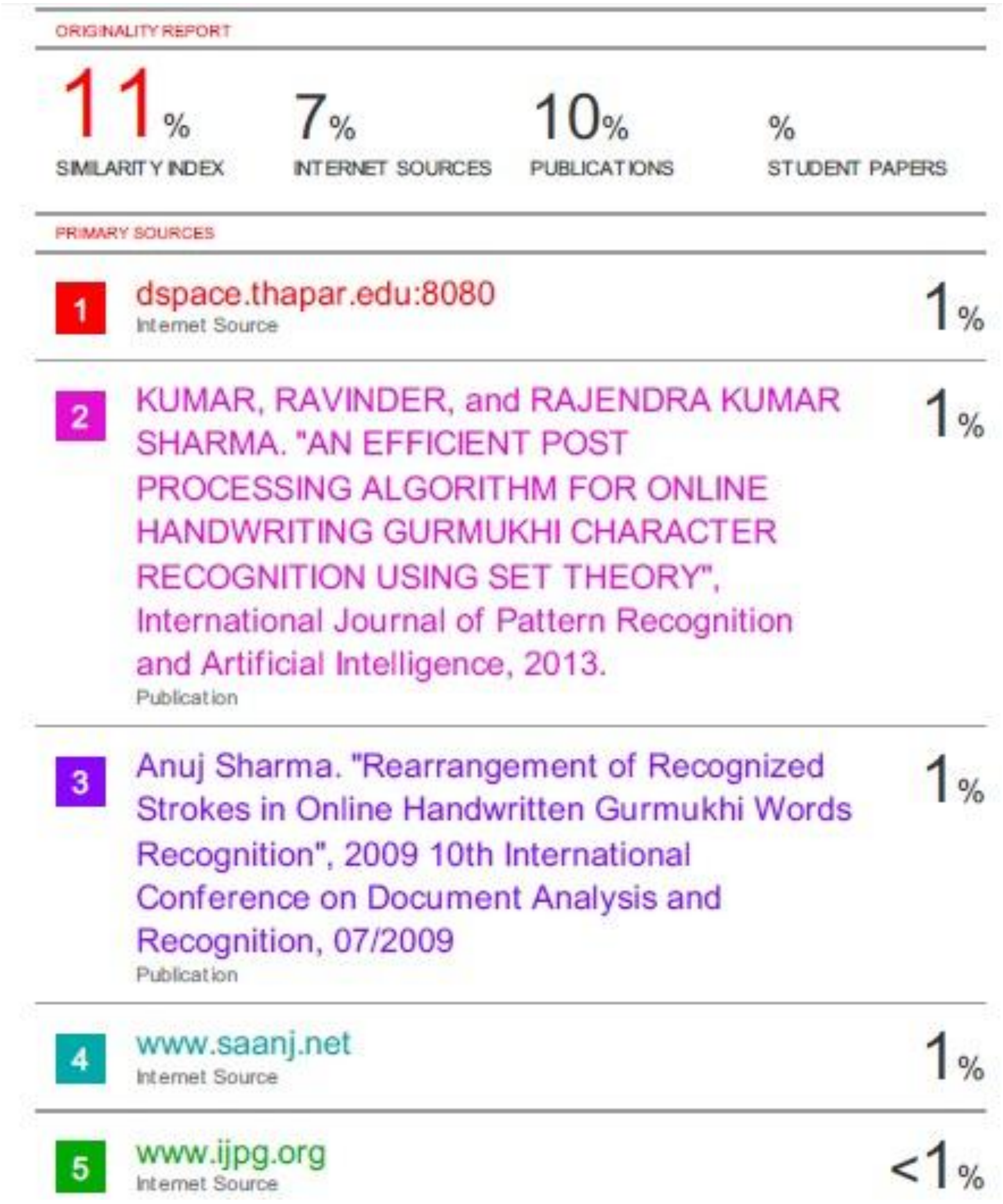
K. Aggarwal and R.K. Sharma, "DFT Based Feature Extraction Technique for Recognition of Online Handwritten Gurmukhi Strokes," in International Conference on Inventive Computation Technologies, Coimbatore, 2016 (**communicated**).

APPENDIX B

VIDEO PRESENTATION LINK

APPENDIX C

PLAGIARISM REPORT



ORIGINALITY REPORT

11%	7%	10%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	dspace.thapar.edu:8080 Internet Source	1%
2	KUMAR, RAVINDER, and RAJENDRA KUMAR SHARMA. "AN EFFICIENT POST PROCESSING ALGORITHM FOR ONLINE HANDWRITING GURMUKHI CHARACTER RECOGNITION USING SET THEORY", International Journal of Pattern Recognition and Artificial Intelligence, 2013. Publication	1%
3	Anuj Sharma. "Rearrangement of Recognized Strokes in Online Handwritten Gurmukhi Words Recognition", 2009 10th International Conference on Document Analysis and Recognition, 07/2009 Publication	1%
4	www.saanj.net Internet Source	1%
5	www.ijpg.org Internet Source	<1%