LBG METHOD AND VECTOR QUNATIZATION FOR RECOGNITION OF PUNJABI HANDWRITTEN STROKES IN HIDDEN MARKOV MODELS

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

Master of Engineering

in

Computer Science and Engineering

Submitted By

Aditi Gupta

(801432002)

Under the supervision of:

Mr. Karun Verma

Assistant Professor, CSED

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

THAPAR UNIVERSITY

PATIALA – 147004

July 2016

CERTIFICATE

I hereby certify that the work which is being presented in thesis entitled, "*LBG Method and Vector Quantization for recognition of Punjabi Handwritten Strokes in Hidden Markov Models*", in partial fulfillment of the requirements for the award of degree of Master of Engineering in *Computer Science and Engineering* submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of Mr. Karun Verma and refers other researchers work which are duly listed in the reference section.

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

Signature:

Aditi Gupta

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

Mr. Karun Verma

Assistant Professor, CSED

Countersigned by

(Dr. Manmder Singh)

Associate Professor & Head

Computer Science and Engineering Department

Thapar University, Patiala

(Dr. S.S. Bhatia)

Dean (Academic Affairs)

Thapar University, Patiala

i

ACKNOWLEDGEMENT

First of all I would like to thank the Almighty, who has always guided me to work on the right path of the life. This work would have not been possible without the encouragement and able guidance of my supervisor **Mr. Karun Verma.** I thank my supervisor for his time, patience, discussions and valuable comments. His enthusiasm and optimism made this experience both rewarding and enjoyable.

I am equally grateful to **Dr. Maninder Singh**, Associate Professor and Head of Computer Science & Engineering Department, a nice person, an excellent teacher and a well credited researcher, who always encouraged me to keep going with the work and always advised me with his valuable suggestions.

I am equally thankful to **Dr. Deepak Garg**, Professor, Thapar University, Patiala, excellent teacher and mentor, who gave motivated me to keep going with good work.

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

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

Last but not the least, I would like to thank my family whom I dearly miss and without whose blessings, this would have not been possible. To my parents, I own thanks for their wonderful love and encouragement. I would also like to thank my brother and my friends for their consistent support.

Date: June 2016 Aditi Gupta

Place: Thapar University, Patiala 801432002

ME (CSE)

ABSTRACT

Handwritten stroke recognition problem is being solved in the work specifically for Punjabi language. Fifty three different types of strokes, which constitute most of the Punjabi characters, are considered here. 242 different instances of each such stroke drawn by various people are used for the training and testing experiments. The recognition system proposed in this report performs quite well yielding high levels of recognition accuracy. This system will help in solving handwritten character recognition which would be a great tool for conversion of handwritten documents into computerized textual documents and recognizing handwritten phrases. Out of many machine learning models we have used hidden markov models (HMMs) for solving this problem which is a doubly embedded stochastic model. HMM performs fairly well in recognizing Punjabi strokes. We have created model for each stroke in our system

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

Character recognition is a well known problem. A lot of work has been done on this problem and many commercially used OCRs are available today. However most of work has been focused on languages like English, Arabic, Japanese or Roman numerals but Indian languages still require lot of work to match up. In this paper, we move a step closer to handwritten character recognition. We design a system to recognize the distinct handwritten strokes constituting the letters of a famous Indian language -Punjabi.

Optical Character recognition is the very popular research area these days. It is the process in which typewritten, handwritten characters or printed documents are scanned first and then translate to machine encoded form. The machine encoded text we get can be edited and also very concise. Handwritten text, printed text or typewritten all can use optical character recognition. Every person has different style of writing so character recognition is even more difficult for handwritten texts. Optical character recognition can be divided into two parts namely online recognition and offline recognition. The offline recognition is the one in which first whole document is scanned and then it's processed with recognition and in online recognition it is done simultaneously in no time.

Skill of drawing symbols, words, or letters with pencil or pen is handwriting. A handwritten character is captured by digital handwritten system strokes sequences. First strokes are analyzed in online handwriting recognition and then particular character is determined with high recognition rate. As there are increasing number of pen/stylus based devices, there are a lot of researchers who have shown their inclination in the field of online handwriting recognition [25, 35]. But in recent past very limited amount of work is done by researchers in online character recognition of Punjabi [38, 39]. However, for languages like English, traditional Chinese, Chinese, Korean, and Japanese languages much of research work is done [36, 37].

Out of many Machine Learning models of computer science we have chosen Hidden Markov Models (HMMs) for solving this problem. HMM is a doubly-embedded stochastic model which seems suitable for this kind of problem. HMMs and their application to speech has been elaborated in [1, 5, 6] papers and the idea seems very much applicable to our problem in handwriting domain as well. Solving this problem successfully has great practical benefits associated. Solution to this problem can help in automating the conversion of handwritten files into electronic documents at regional offices (government or non-government) of Punjab and Eastern Pakistan. These documents can then be stored, accessed and manipulated in computers with much ease and will lead towards the digitization of the office work. Also this can be a major breakthrough for searching the large collection of manually written files stored in cupboards. Many software applications can be developed using stroke recognition to make work easier for example words can be entered manually on touch-screen devices using finger or pen devices conveniently.

The system is developed in C++ from scratch and trained for Punjabi language. Punjabi language is an Indo-Aryan language. 130 million people approximately voice this language moat in west Punjab in Pakistan and east Punjab in India.

Punjabi is a popular language in north India and eastern Pakistan. It is the language which is written from left to right. It comprises of 41 consonants, 9 vowels (laga matras), addak which is used to duplicate sound of a consonant, tippi and bindi as two symbols for nasal sounds.

A segment of the character which is drawn without lifting the pen/pencil up from the paper is a stroke. Thus each Punjabi character can be supposed to be composed of one or many sequence of strokes. Here we aim to recognize and identify the individual strokes. Any continuous segment taken from a stroke is called its sub-stroke. In simple words, sub-stroke is just a part of a stroke and may be of any length. Characters in Punjabi language can be written in different styles due to which there arises a difficulty in character recognition. Strokes in Punjabi can be drawn in three zones namely, middle, upper and lower zones as shown in Figure 1.1. In middle zone most Punjabi characters are written. Upper zone is the region above the headline. Area below the headline and

above baseline is middle zone. In the middle zone sub part of some vowels and the consonants are present. Area below middle zone is lower zone. In lower zone lie some half characters and few vowels. Detection of baseline and headline is the major problem in character recognition in Punjabi. There are also some strokes in these three zones which are quite similar in looks. Such strokes further increase the difficulty of character recognition process.

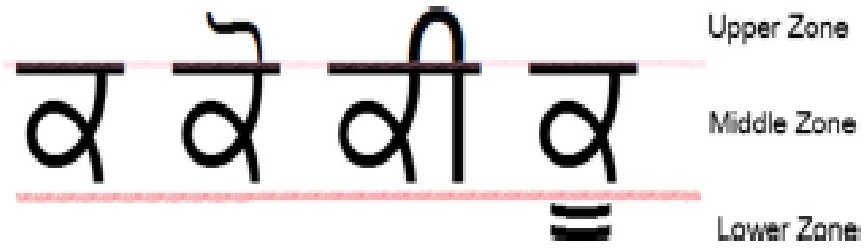


Figure 1.1: Zones in Punjabi Language [2]

1.1 Dataset

Our dataset is based on Punjabi Character-Set on which very less work has been done till date, but instead of recording the actual characters we have recorded the 'strokes' used recurringly in formation of Punjabi characters. A stroke is defined as the part of the character which is formed without shifting the pen up from the paper. Thus each character can be supposed to be composed to multiple strokes. Dataset contains collection of multiple instances of each stroke for maintaining variety and building a robust system. In this paper we are only concerned with the recognition and identification of the individual strokes. There are 53 strokes (or classes) numbered from 141 to 223 excluding 142, 154, 155, 158, 160, 162, 163, 164, 170, 173, 178, 181, 182, 183, 188, 189, 192, 194, 196, 199, 201, 204, 206, 209, 212, 213,217, 219, 220, 221 for no specific reasons.

1.1.1 Dataset Collection

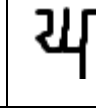
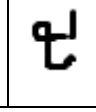
A total of 53 strokes for building Punjabi letters are considered here. For each such stroke, 242 hand-drawn instances of it are recorded using touch-screen application. All

such instances are normalized to fit into the same sized square. Thus, we end up having a corpus of 242 different normalized drawings for each of the 53 different strokes that is total of 12826 instances of strokes.

Strokes are represented by stroke ID which is shown in Table 1.1. Various strokes taken for experiment with their respective strokes ID

Table 1.1: Stroke ID with their Representation

Stroke ID	**Representation**	**Stroke ID**	**Representation**	**Stroke ID**	**Representation**
141		167		195	
143		168		197	
144		169		198	
145		171		200	
146		172		202	
147		174		203	
148		175		205	
149		176		207	
150		177		208	
151		179		210	
152		180		211	

153		184		214	
156		185		215	
157		186		216	
159		187		218	
161		190		222	
165		191		223	
166		193			

1.2 Feature Selection and Stroke Representation

In this step, each recorded stroke instance is broken into 64 equal lengthed-parts taking 65 equidistant points on that instance-curve. Now on joining the contiguous points to each other, the stroke instance gets reduced to just a sequence of 64 lines. Now the sequence of angles these 64 lines make with horizontal axis are recorded[2]. This sequence of 64 angles represents the corresponding stroke instance in our dataset.

Instead of noting the precise angle, we divided the 360 degree range into 20 different sectors (namely A-T) with each sector covering 18 degrees such way. Here angle A means that the angle lies between the range 0-17 degrees with horizontal, angle B means that angle lies between the range 18-35 degree with horizontal and so on[2].

1.3 What is HMM

1.3.1 Elements of HMM

"Hidden Markov Model is a doubly embedded stochastic process with an underlying stochastic process that is hidden or that is not observable, and it can only be observed through another set of stochastic processes." Those processed produce the sequence of observed symbols [1].

- N - Number of states in the system [1].
- M - Number of distinct observation symbols per state i.e. the codebook size [1].
- State Transition Probability Distribution[1]:

$$A = [a_{ij}] = P(q_{t+1} = s_j / q_t = s_i)$$

- Observation Symbol Probability Distribution in state-j

  $B_j[k] = P(b_k \; at \; time = t / q_t = s_j) \; where \; 1 <= j <= n \; and \; 1 <= k <= m$[1].

- Initial State Probability Distribution: $\pi = \{\pi_i\}$ where $\pi_i = P[q_1 = s_i]$ and where $1 <= i <= m$ [1].
- Topology chosen for the particular application domain.

1.3.2 Three Problems of HMM

1) Evaluation problem [1] - Given a model and sequence of observations how to compute the probability that the observation sequence was produced from the model. It is also called scoring problem. In other words it means how well a given model matches a given observation sequence. We may have several models, we need to find maximum of $P(O/\lambda i)$ where i = 1….w and w is number of models. Solution to evaluation problem is Forward Backward Algorithm.

2) Uncovering problem [1] - given an observation sequence $O_1,O_2,O_3,...O_T$ and $\lambda = A, B, \pi$. How would you chose a corresponding state sequence q = q_1,q_2,q_3, ….q_T in a meaningful manner. We are actually measuring the quality of the system here we will never know the actual sequence. Solution to this problem is Viterbi Algorithm given in [4, 6].

3) Re-estimation problem [1]- How to adjust the model parameters (A, B, π) to maximize the probability $P(O/\lambda)$. We use Baum Welch Procedure or Expectation Maximization Procedure given in [6,7] to solve this problem as best as possible.

Chapter 2
Literature Survey

2.1 Brief about Character Recognition

Human computer interaction is generally done using keyboards and mouse. We are advancing in computational technology as more and more touch devices are being introduced and now we are progressing towards more natural way of communication. Handwriting recognition has been area of interest for many researchers across the globe since 4 decades. Great advances have been made in this field and also efficiency and reliability of handwriting based devices has been improved over the years. There are Various recognition methods proposed by various researchers in the world for handwriting/character recognition of languages for example hidden markov model (HMM), support vector machine (SVM), elastic matching, rule-based methods and dynamic time warping (DTW).

In our literature survey we have studied many research approaches practiced on many languages and scripts particularly in foreign languages, Indian languages and Punjabi language.

Languages like Chinese and Japanese have very huge number of alphabets, so it becomes extremely tough to input such type of data to computers. Also for languages like Punjabi and Hindi it is also difficult for computers to input as these scripts are of complex nature. Speech recognition and handwriting recognition are two natural ways for inputting the data into computer. Speech recognition has to deal with problems like noises. In our thesis we are limited to handwriting recognition. Major problems in handwriting recognition is variations that is different people write differently and thus yielding high accuracy is difficult task. Every writer has different speed of writing, different style, different position, slant and different size of text. Also one person may write differently at different times due to using different hardware for writing, or different mood of the writer, varied situations of the writer, or writing same characters with different shapes in made in recent years different situations.

2.1.1 Research done in foreign languages

Below discussed is the various advances made in character recognition in recent years in foreign languages like English, Japanese, Chinese and Greek.

Lazzerini and Marcelloni [8] have introduced EYE a classifier which is based on fuzzy logic. It is used for offline recognition of handwritten characters. They introduced classification method based on the linguistic description of the shape of the character. They yielded 68.2% recognition rate.

Hanmandlu et al. [9] gave us an innovate method which is used for feature extraction. He uses this for recognition of handwritten characters. First image of character is partitioned into sub images which are called boxes. They got features which consist of angle from each box to a fixed point and normalized vector distance. The recognition schemes used are fuzzy logic and back propagation neural network (BPNN). They got 100% accuracy rate. They used fuzzy based method on the standard database.

Feature for handwritten character recognition is proposed by Zhang et al [10] which is combination of coefficient of gradient feature and wavelet transform. The character had local characteristic which is represented by gradient feature, and is very sensitive to the distortion of handwritten character. The character image in multi resolution analysis is represented by wavelet transform which keeps adequate global characteristic of a character image in different scales.

Plamondon and Srihari [11] did a survey on online and offline handwriting recognition systems. They explained full process which starts from inputting the character and goes till final characters recognition for a particular language. Recognition methods can be divided into two categories provided input is based on pen based devices that is rule/ structural based methods and statistical methods. Rule/ structural -based methods have more reliable and robust rules which are used in character recognition. And statistical methods, involves strokes which is the shape of the character which has some features and the multidimensional probability distribution are used to describe classes of these shapes.

Hu et all [33] proposed a model which is based on Hidden Markov model (HMM) handwriting recognition system. There is a lot development in speech processing field, so they took advantage of this point in order to get more sophisticated recognition system of handwriting. HMMs model some sub character stroke types which are the pattern elements of handwriting model. Letter models are prepared by HMMs, which are further used in a stochastic language model. This model represent various new features regarding handwriting recognition out of which some features include invariant properties and some covers a large portion of the input pattern. Their model has achieved a writer independent accuracy of 94.5% on 3,823 unconstrained handwritten word samples from 18 writers that covers vocabulary of 32 words.

Salicetti [34] proposed a system that is designed for online characters recognition to words. It is also extension of neural predictive system. Digitizing tablet records the pen trajectory information and feature extraction is done after its re-sampling. Modeling of each letter is done with the help of concatenation of letter-models. Different neural network models successive parts of word trajectory, moreover only those transitions are permitted which are to be done to its right neighbors. A dynamical and holistic segmentation permits adjusting of letter-models so that large variations of handwriting can be done in the words. In this, combination of dynamic programming and multilayer neural networks is done with hidden Markov model (HMM) which works from left to right. Training is done on 7000 words which are written by 9 writers. In the letter-labeling procedure they achieved good results, without any help of language models.

Bellegarda et all [35] handled problem for Automatically recognition of handwritten text. First the data (text) which was to recognize was captured through online medium and was presented in the form of temporal data sequence. Approach used was left-to-right HMM (Hidden Markov Model) on every character. The output probabilities distributions on each arc of HMM was represented through the Gaussian distributions and their mixtures. There have been many approaches to model and re estimating the parameters. Experimentally it has been shown that the error rates has been reduced for discrete characters while comparing results with elastic matching techniques. It has been concluded that recognition using HMM techniques performs better than elastic matching

techniques for both types of recognition tasks whether they are writer-dependent or writer-independent.

Liu et all [36] proposed a online character recognition. With the increase of new pen input devices and pen computing applications character recognition gained popularity. The western handwriting recognition and Chinese handwriting recognition are quite different. Chinese recognition system is quite tough. Authors reviewed the advances in online Chinese character recognition (OLCCR), with emphasized on the research works from the 1990s. The efforts in research in the 1990s aimed to relax the constraints of handwriting as compared to research efforts in 1980s. The sticking to standard stroke orders, the limitation of recognition to isolated characters only and stroke orders were concentrated in research during 1990s. In order to fulfill the needs of practical applications the objective of handwriting recognition has moved from regular script to fluent script. The research is evaluated in terms of character classification, pattern representation, contextual processing and adaptation/learning.

Nakagawa et all [37] compared current ongoing research and methods for the online character recognition of Japanese script with handwriting recognition techniques for western techniques. The developments of various method used in classification, preprocessing and post processing for character recognition in Japanese in recent years has been discussed with recognition of western handwriting. After comparing recognition from both eastern and western handwriting, there have been different aspects of learning and understanding features which are common in handwriting recognition. This is so helpful and significant thing when thinking for developing various concise and efficient modules for integrated systems which support many writing systems which are capable of identifying multi language documents.

The machine recognition techniques proposed by Kana, Latin and Chinese characters have been in progress since past two decades. Apart from that the machine recognition technique of Arabic characters has not been touched. In this context a method of structure recognition has been proposed by Almuallim et all. [25] which is of Arabic cursively handwritten words. They converted the words into segments of strokes and classified these strokes based on the two properties which are geometrical and topological. After

strokes classification their relative position was examined. Then strokes were grouped or combined applying several steps and a string of character was formed that represent a word. After experiments on text handwritten performed by two persons it has been observed that the results show high accuracy for character recognition.

2.1.2 Research done in Indian Languages

Researchers who work on Indian scripts perform their research mainly on Bangla script and Devnagri script.

Pal and Chaudhary (2004) [21] did a survey on Indian Scripts Character Recognition. They introduced features of Indian scripts, work done and methodologies used so far in various Indian scripts. They have studies work done so far foe many Indian languages such as Hindi, Punjabi, Bangla, Tamil, Gujarati, Kannada.

Wakabayashi *et al*. (2009) [22] performed comparative study of Devnagri characters using four feature sets which are based on gradient information and curvature obtained from binary as gray scale images. Also they studies recognition using 12 different classifiers namely Modified projection distance, Projection distance, Linear Discriminant function, subspace method, modified quadratic Discriminant function, support vector machines, Euclidean distance, mirror image learning, nearest neighbor, k-Nearest neighbor, compound modified quadratic Discriminant function and compound projection distance. Mirror Image Learning (MIL) classifier gave the best results that is 94.94% and 95.19% for features extracted from binary and gray image respectively. For many classifiers curvature features yield better efficiency than gradient features.

Dungree et al (2010) [23] have surveyed the methodologies currently available for character recognition. They presented computer vision techniques like thresholding, normalization of size, skew identifying and its improvement for use in character recognition. They also did review the feature obtaining with the help of mathematical series and transformations like Discrete Fourier Transform (DFT), Gabor Transformation, projection, distance and few geometrical features are commonly used. The classification using various other models like NNs, SVM, matching the templates

and using ensemble of classifiers for better accuracy for recognition has also been reviewed.

A statistics based technique for character recognition using Bayesian Filter was presented by Araki et.al. (2008) [26]. They obtained nice recognition rate despite the elegance of Bayesian procedure. The author proposed a Bayesian Filter based brand new character recognition procedure. Handwritten characters pictures scanned from the paper are used in the process. Preprocessing, learning and recognition are the 3 main steps which constitute the method. During first step, scanned image's threshold processing is preformed and its size normalization is also done. Then, in learning, the black pixel's appearance count is found out. At last the recognition step determines the probability of a given character to match with the learned one using Bayesian Filter technique with the help of appearance count. Experiments showed that this method has recognition accuracy of 90% despite it using only low learning data. Nozomu Araki et al presented a statistics based method on similar grounds. They also reported nice accuracy rate despite its simplified process.

Mukherji and Rege, (2009) [24] proposed a method for offline Devanagari character recognition which used fuzzy logic and shape features. Characters were divided into strokes by making use of structural features like cross-point, endpoint, thinning and junction points. Strokes or segmented shapes were classified as right curve, left curve, vertical stroke, horizontal stroke, slanted lines *etc*., accuracy of 86.4 is achieved using fuzzy and tree classifier.

Pauri et all [14] worked on handwritten isolated numerals of Devanagari which is most widespread language in India and third most common language in the world. They presented a two stage grouping system which recognized handwritten Devanagari numerals. First a character image was given as input and then a shape feature vector was determined from certain directional-view-based strokes. Then these shape feature vectors were utilized by 2 classifiers namely HMM and ANN. After this step they got an output by using these two classifiers. These outputs were two set of posterior probabilities which were combined using ANN classifier. ANN of the second stage provided maximum score

according to which numeral image is classified. They developed large database of character images. They achieved accuracy of 92.83% in their proposed scheme.

For Indian languages character recognition has been done on various languages for example in Bangla language Biswas et all [12] presented a paper HMM based recognizer using dirichlet distributions. His method was based on two stages approach. First probability distribution is estimated for each class. In second stage every stroke class in considered as state and character classifier is designed which is based on HMM. The character level recognition rate obtained by the proposed method on the test set having 8,616 samples, is 91.85%.

Gernot A et all. [13] gave a new method to recognize online Bangla handwriting. They were first one who considered cursive words in lieu of isolated words. Their technique used a substroke level feature representation of the script and a writing model which was based on hidden Markov models. Bangla is a highly compositional script so to define model structures they investigated different approaches. In this paper they showed that if context dependent sub word units are used they showed better results than any alternatively structured models.

Bangla language is an Indian language which is spoken by over 200 million people and Guin et all [15] proposed a scheme on online handwriting recognition of Bangla characters. They have created a database of 50 basic characters of Bangla which were written by 70 different people in different ways so they got total of 24,500 samples of online handwritten isolated characters. Using shape similarity these strokes are classified into 54 different classes manually. Then hidden markov model is applied to these classes and for each class hidden markov model is created. Characters are recognized using stroke classification in second stage of classification.

Jayaraman et all [16] proposed a system for Online Handwritten recognition of Telugu characters, a famous language in Southern part of India. This script has around 500 characters. Each character is composed of sequence of strokes. There are 253 unique strokes. According to the relative location of the stroke in character, the set can be divided into 3 categories – baselines, bottom and top strokes namely. SVMs – Support

Vector Machines are used for building the respective classifiers for these categories. They analyzed the performance and proposed ways to improve it. They proposed that SVM performed more accurately than HMMs in comparison.

In this paper Aparna et all [17] addressed the framework for developing online recognition system for handwritten Tamil characters. In general, a sequence of strokes constructs a character. Here sequences of shape features are used to depict a stroke. Using this representation, a given stroke can be identified on making comparisons with all the strokes in our corpus with the help of a flexible string matching method. Thus, the complete character can also be recognized by recognizing the individual strokes utilized to build it and their order. The end of character is detected with the help of Automaton (FSAs). The ideology of building systems for other Indian scripts is also discussed.

After the success of Hidden markov Models in Speech, these have also became very famous for English cursive handwriting recognition. For Chinese, Korean and Japanese also HMMs are extensively used for modeling of characters' substrokes. However very less work is done for Indian scripts in this direction and current work is only on individual character recognitions. So for this, an online handwritten recognition system for Tamil words of Indian script was presented by Bharath et all [18]. High accuracies were obtained which were between 92.2% to 98% and results were quite inspiring to use HMMs on Indian scripts also.

2.1.3 Work on Punjabi Character Recognition

There are many researchers who have worked on recognition of Indian languages in general and Gurmukhi in particular. A detailed survey has been done on research work done so far on Indian languages. Researchers also discuss properties of Indian scripts, approaches and methods applied to recognize are discussed.

G. S. Lehal and C. Singh focused on Gurmukhi script have performed major research in the character recognition for Gurmukhi characters only. They built a full system for recognition of printed Gurmukhi characters [27]. Specific to Gurmukhi, previous major works were done by Chandan Singh and G.S.Lehal. They presented recognition system for Gurmukhi Script in their paper Singh and Lehal, 2000 [28]. After this, they

constructed a whole Gurmukhi Optical Character Recognition system in 2006 [29]. In other works of their related to Gurmukhi Script, they presented approaches like extraction of features, post-processing and classification.

Sharma *et al*. (2008) and Sharma *et al*. (2009) [30] proposed the implementation of three approaches: elastic matching technique, HMM based technique and small line segment to recognize online handwritten Gurmukhi character and reported 90.8%, 91.59% and 94.59% recognition rates respectively.

Anuj Sharma et.al [30] [31] proposed the 3 techniques - HMM based, elastic matching and small line segments technique for recognition of handwritten Gurmukhi characters online. He also found of the recognition rates of 91.95%, 90.08% and 94.59% respectively.

Dharam V. Sharma et all. [32] Withdrew digits from Gurmukhi script from hardcopies of documents and then tried the recognition process on those. Several structure based features were implemented like curve, line segment, entry, image aspect ratio and other features based on statistics such as directional distance, zoning, etc. A very good accuracy of 92.6% was observed in this approach for Gurmukhi digits.

Kumar et all [19] proposed a time-efficient procedure for Gurumukhi character construction and recognition. A specially constructed data set was used to train the SVM (Support Vector Machines) classifier for classification. The software tool LibSVM has been used for experimental results. Once the preprocessing is done, LibSVM is used to scale the coordinates from 1 to 9. A corpus of 46,772 words was built out of which 2,47,697 strokes were extracted and then annotated. They then tested their methodology on 4,310 instances of Gurmukhi words gathered from various people. Testing results display high recognition accuracy for various Gurmukhi strokes and vowel combinations.

Sharma et all [20] presented a paper named hidden Markov model-based online handwritten character recognition for Gurmukhi script. So as to recognize Gurumukhi character they proposed a procedure to develop a hidden Markov model database. Hidden markov model database was prepared in XML technologies with 5330 Gurumukhi characters. Accuracy rate was 91.95% when 60 handwritten samples were

taken and each sample included 41 Gurumukhi characters. And also an average recognition speed of 0.112 seconds per stroke was achieved. This work can be useful to implement a Hidden Markov Model in online handwriting recognition and its software development.

Kumar R, Sharma RK (2013) [38] proposed a post processor method for recognition of character on online Gurmukhi script taken in real-time scenario. They performed experiments on dataset which consists of 184 samples and is of each 45 characters and they took it from Gurmukhi script. They took script of four different categories. Hence using extensive and deep study they proposed an efficient algorithm to recognize handwritten Gurmukhi character. This algorithm promised a recognition accuracy of 95.6 % for a single character sequence stroke. Apart from summarization the paper contributed in following two manners.

1) The stroke sequencing was resolved by the proposed scheme.

2) Technique identified and resolved overwritten strokes. The complexity was $O(n)$ for adding a new stroke of Gurmukhi character.

Verma K, Sharma R (2015) [39] proposed a method in offline character recognition techniques, where features based on Zone has been used. These features have been implemented for data collected from online source for 30 users of set of 82 unique middle zero strokes taken from Gurmukhi script. They achieved an optimal performance using Support Vector Machine (SVM) and SVM uses kernel methods and considers various parameters like learning rate (ε), umber of folds (k) and tolerance limit of termination (γ). They observed that Zone base Feature for recognition gave 92.09% accuracy with SVM or the online handwritten text and considering diagonal features.

Chapter 3
Problem Statement

3.1 Problem Statement

The aim is to build a system capable of recognizing an input stroke with a fair precision i.e. a system able to tell the actual class correctly which that stroke belongs to. Our problem is majorly inspired from the problem of Punjabi Handwritten-Character Recognition. As solution to our problem (stroke recognition) can majorly help in solving the bigger (character recognition) problem, so we chose to solve this problem first.

3.1.1 Definition of Strokes

A stroke is a segment of the character which is drawn without lifting the pen/pencil up from the paper. Thus each Punjabi character can be supposed to be composed of one or many sequence of strokes. In this paper we aim to recognize and identify the individual strokes.

3.1.2 Sub-Strokes

Any continuous segment taken from a stroke is called its sub-stroke. In simple words, sub-stroke is just a part of a stroke and may be of any length.

3.2 Objectives

Objective of this thesis is to give a minor contribution to character recognition using Hidden markov models for Punjabi Script which can help in converting handwritten documents into digital documents.

Another objective is to speed up the processing of the data and speeding the process of searching millions of documents. The System is capable of isolated recognition of the strokes in Punjabi language. Markov model for each stroke class is made and it is trained until recognition rate is achieved higher.

Chapter 4
Implementation

Our formulation for implementation has been inspired from the way HMMs are applied to the speech recognition problem explained in [1, 5, 6]. The strokes in handwriting domain in this paper can be seen analogous to speech utterances in the speech domain.

4.1 Modeling Strokes Using HMM

Stroke is the basic building unit of any character. We chose to represent it with a 5-state left-to-right HMM model in which state number increases with time always i.e. system goes from a lower indexed state to a higher indexed state only. The loop on the state indicates the system may even stay in the same state. Figure 4.1 depicts state transition diagram from left to right.

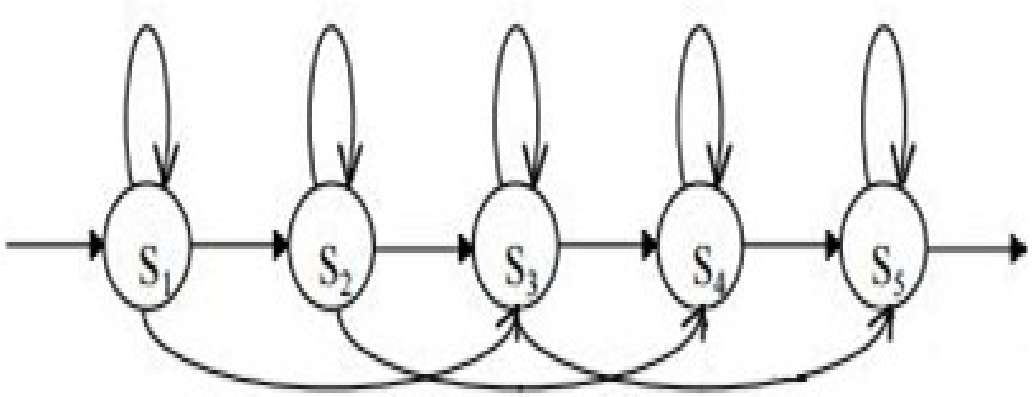


Figure 4.1: State Transition Diagram

4.2 Integral Representation of Stroke

In the whole dataset, the angular ranges denoted by A to T were replaced with integral values 1 to 20 respectively for having data in integral form. This helps in Euclidean distance measure being applicable between two sub-strokes and calculations to be meaningful and quantitative. For example :- The sub stroke - BCEKJHGGAA will be represented as - 2,3,5,11,10,8,7,7,1,1.

Each stroke instance thus in our dataset is an integral vector of size 64. In Figure 4.2 A) represents a stroke which is divided into small parts using points. Figure 4.2 B) represents a stroke in which points are connected using straight line. In Figure 4.2 C) angle between horizontal and straight line is calculated and Figure 4.2 D) represents the corresponding character given to an angle.

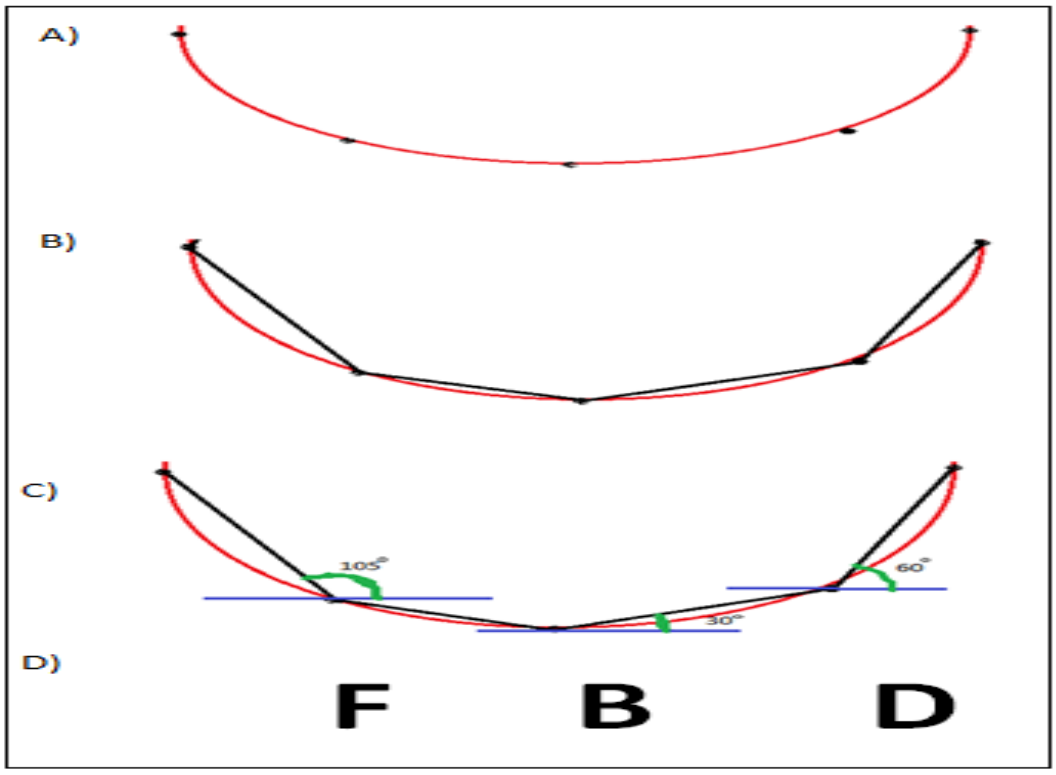


Figure 4.0.1: Stroke Representation

4.3 Parameters Values for our System

Parameter values for our system: We have taken following values of parameter for our experiment. Value of N is taken as 5 which are number of states in the system. Value of M is 32 which are number of distinct observation symbols per state i.e. the codebook size. Matrix A shows probability of transition from previous state to next state. Matrix B shows probability of observation on some given state. π is initial state probability of all the five states. And topology we have taken is left to right it means states will progress in left to right direction.

Table 4.1: Showing Parameters and their Values

Parameters	**Values Taken**
N	5
M	32
A	$\begin{bmatrix} 0.8 & 0.2 & 0 & 0 & 0 \\ 0 & 0.8 & 0.2 & 0 & 0 \\ 0 & 0 & 0.8 & 0.2 & 0 \\ 0 & 0 & 0 & 0.8 & 0.2 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
B	$\begin{bmatrix} 1/32 & 1/32 & 1/32 & .. & 1/32 \\ 1/32 & 1/32 & 1/32 & .. & 1/32 \\ 1/32 & 1/32 & 1/32 & .. & 1/32 \\ 1/32 & 1/32 & 1/32 & .. & 1/32 \\ 1/32 & 1/32 & 1/32 & .. & 1/32 \end{bmatrix}$
π	[1 0 0 0 0]
Topology	Left -to -Right

4.4 Universe & Codebook Creation

From each instance of handwritten stroke vectors of size n are taken by shifting window size by k.

An analysis window of size n is chosen. The window is applied on each handwritten stroke instance vector and shifted by size k every time. The values of n and k selected in our experiment were 8 and 1 respectively Thus each stroke instance gives us a sequence of sub-strokes (which are themselves integral vectors of size n).

Each stroke instance (being vector of size 64) gets converted into a sequence of 57 sub-stroke vectors. The sub-stroke vectors from all the strokes' instances are obtained and dumped into a file called Universe.

Universe file has many similar vectors in it which are clustered using LBG clustering algorithm. This is actually the process of vector quantization used for compression of data and elimination of redundant vectors as elaborated by Rabiner, Lavinson and Sodhi et.al. in [3]**.** The number of clusters is chosen as 32, thus total 32 clusters gets created. For each cluster there is one representative vector. Hence we get 32 different vectors (possibly) which we store in a file called Codebook. Codebook of 32 vectors is fundamental block for our handwritten recognition system. Here 32 codebook vectors represent 32 different sub-strokes. Vector quantization is the central concept used in the generation of codebook and is explained in E subsection.

4.4.1 LBG - Linde Buzo and Gray Algorithm

It is also known as modified k-means algorithm or Binary Split Algorithm

Step1: Design a one vector codebook. This is centroid of the whole data i.e. average of all training vectors.

Step2: Double the size of the codebook by splitting each current codebook vector y_n according to the rule.

$$y_n^+ = y_n(1 + epsilon)$$

$$y_n^- = y_n(1 - epsilon)$$

where n varies from one to the current size of the code and epsilon is the splitting parameter.

The value of the splitting parameter epsilon lies between 0.01 and 0.05

Step3: Use *k*-means algorithm to get the best set of centroids for the split codebook - codebook of twice the size.

Step4: Iterate Step2 and Step3 until a codebook of desired size is reached.

Figure 4.3 shows flow chart of Linde Buzo & Gray Algorithm.

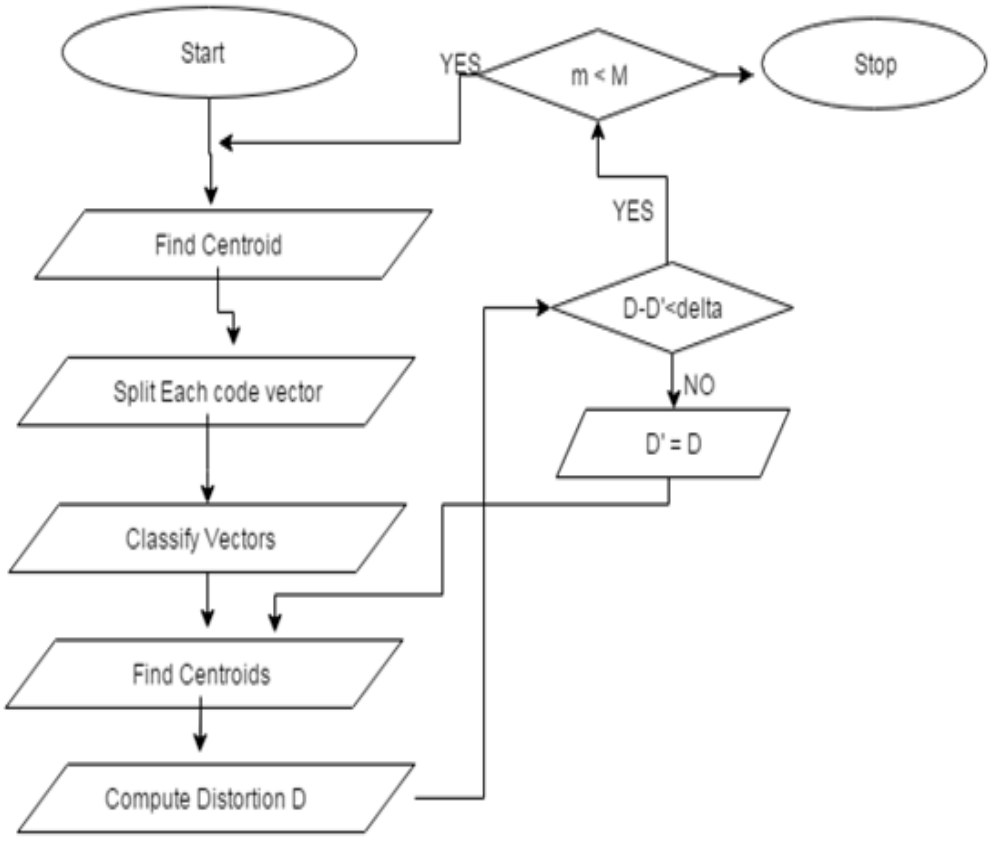


Figure 4.3: Flow Chart of LBG Algorithm

4.5 Observation Sequence Generation

In this part an observation sequence is generated for each instance of each stroke or class. For each stroke, a sequence of vectors of size n is generated. Then this sequence of vectors get converted into a sequence of indices based on the principle that the vector is replaced with the index of that codebook vector with which its distance is minimum out of all codebook vectors. We use distance metric as Tokhura distance which is distance between two vectors. For example if we have 2 vectors (a,b) & (c,d) then Tokhura distance is calculated as

$$\sqrt{(w_1(a-b)^2 + w_2(c-d)^2)}$$

Where w_1 and w_2 are weights associated with first and second dimensions resp. As the observation sequences of each stroke instance are generated, the amount of compression gets even better. Initial Hidden Markov Models are then trained to generate representative models for each stroke class.

4.6 Vector Quantization

Vector quantization is a process in which multiple vectors which are similar to each other are represented by single representative vector that is mapping multiple vectors to a single vector. So large set of vectors can be broken into k smaller subsets each having vectors similar to each other. These k subsets of vectors are then represented by only k distinct vectors. Vector quantization technique used in this paper in the form of codebook generation has been inspired largely from the paper [3] of Rabiner and Lavinson et.al.

Definition: Let x be a k dimensional vector whose components are real valued random variable. A vector x is mapped onto another k dimensional vector y and written as

$$(y) = q(x)$$

1) Advantages of Vector Quantization

   a) It requires less storage for analysis

   b) It reduces computation for determining similarity

c) It helps in discrete representation of sub-strokes

2) Disadvantage of Vector Quantization

a) Quantization or Distortion Error

4.7 Training

Figure 4.4 explains the training phase of the experiment. It depicts the training process happening on each observation sequence or stroke instance. Initially an observation sequence and an initial model $\lambda_0\,(A_0, B_0, \pi_0)$ is fed to the viterbi algorithm. This algorithm is dynamic programming algorithm which solves second problem (Uncovering Problem) of HMM. It outputs the maximum probable state sequence q^* for the given observation sequence and that maximum probability is denoted by say p^* . Then this state sequence and its corresponding observation sequence are fed into the Baum Welch Algorithm which outputs the model that best explains this relation. This process is repeated for several iterations until the maximum probability p^* get stabilized and the result of this process is obtaining an improved model λ for that particular observation sequence or stroke instance. This process performed for each observation sequence of a particular class and thus the final model for that class is obtained by taking average of all these models.

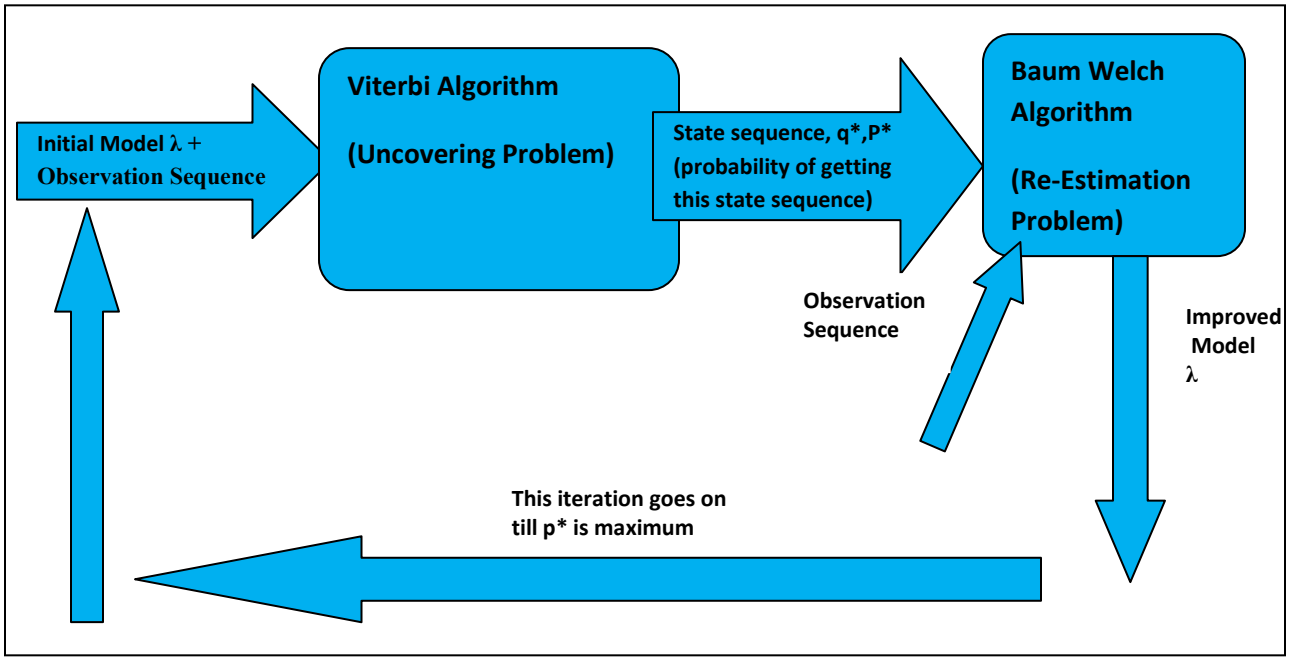


Figure 4.4: Training process

We have 53 classes each containing 242 instances. We have total of 12826 strokes. We trained our model on three different types on training data and tested our model on three different combinations of data. Table for which is shown in results section.

Initial Hidden Markov Models (containing A, B and π matrices) for each stroke class are assumed as shown in table above and then these models are trained on the instances of the corresponding stroke. A Model is represented by 3 matrices namely A, B, π.

A-matrix is state transition matrix. It is C X C matrix with C being the number of hidden states of system. Each entry in matrix represents-

$$[a_{ij}] = P(q_{t+1} = s_j / q_t = s_i)$$

i.e. probability of going into sj state in the next time instant, when the current state is s_i

B-matrix is observation output probability at a state. It is a C X 32 matrix.

π matrix- i^{th} entry of π matrix denotes the probability that initial state is si. So at the end of this phase, HMMs for all required stroke classes are generated.

4.8 Testing

Figure 4.5 explains Testing Phase in which observation sequences of testing data along with the HMM models we have found out in training module are given as input and these input are fed to forward and backward algorithm.

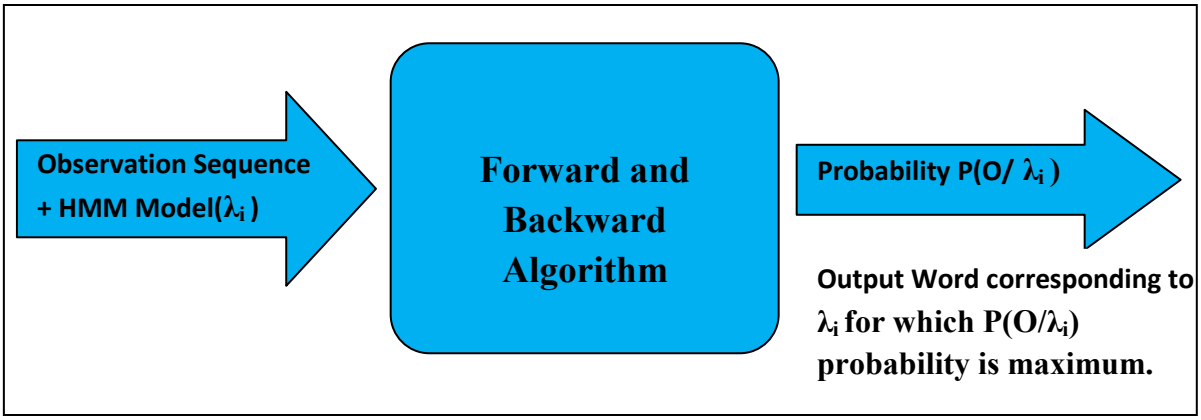


Figure 4.5: Testing Process

Probability of each observation sequence given each model i.e. $P(O/\lambda_i)$ is calculated and output stroke is fetched corresponding to HMM model for which probability of observation sequence given i-th HMM model $P(O/\lambda_i)$ is maximum.

Chapter 5

Simulation Results

Below are the screenshots of how this character recognition process works. It shows how we are training and testing on Punjabi strokes. Figure 5.1, Figure 5.2 shows system is reading the strokes successfully.

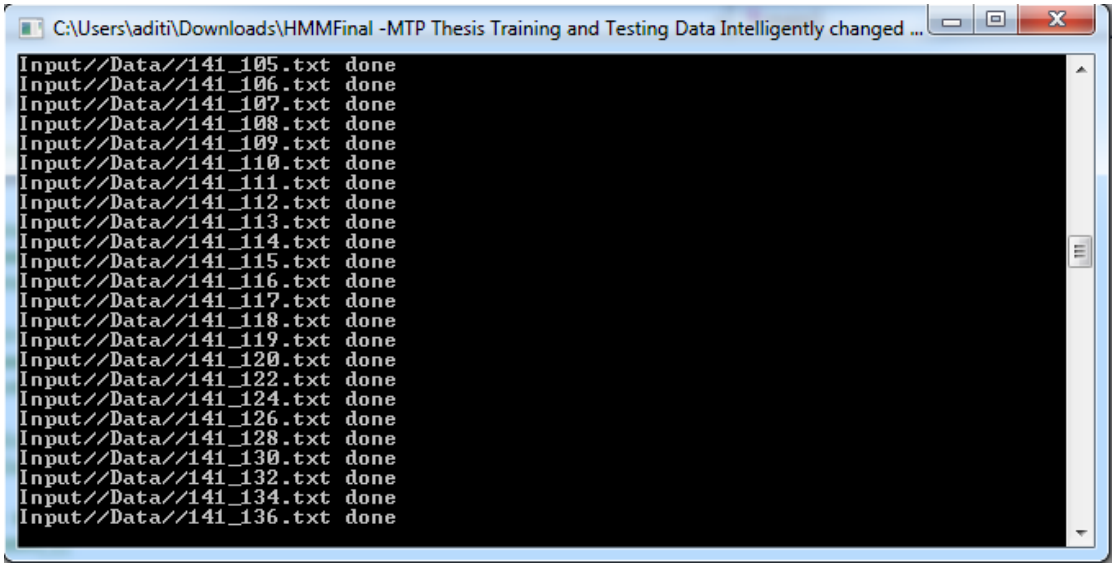


Figure 5.1: Reading Strokes Files

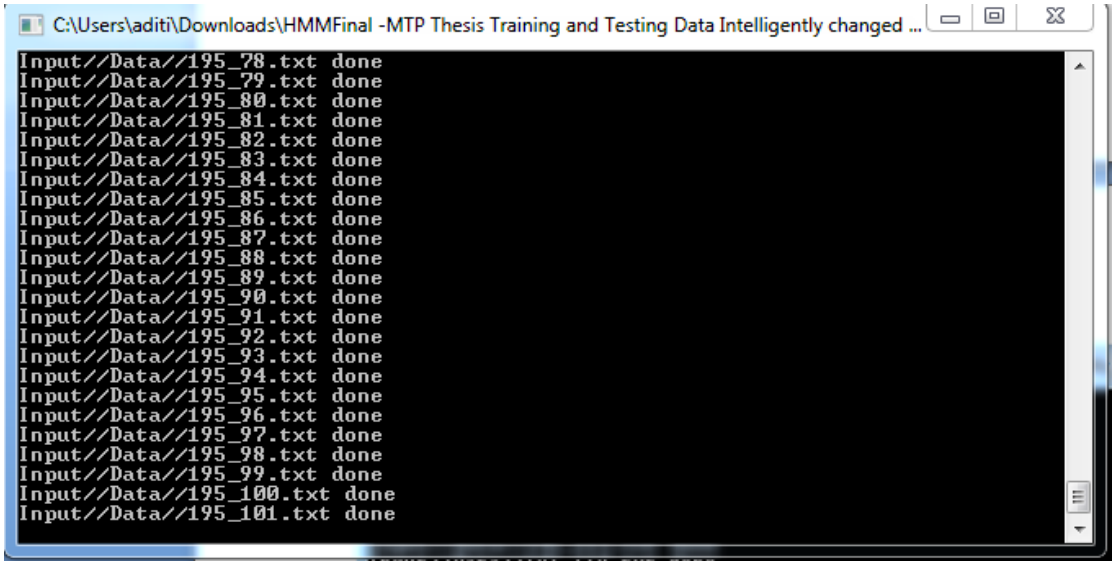


Figure 5.2: Reading Stroke Files

There are 12826 instances of strokes which are read successfully. Figure 5.3 shows after all the files have been read successfully Universe is created which is a file which contains vectors as explained in implementation part.

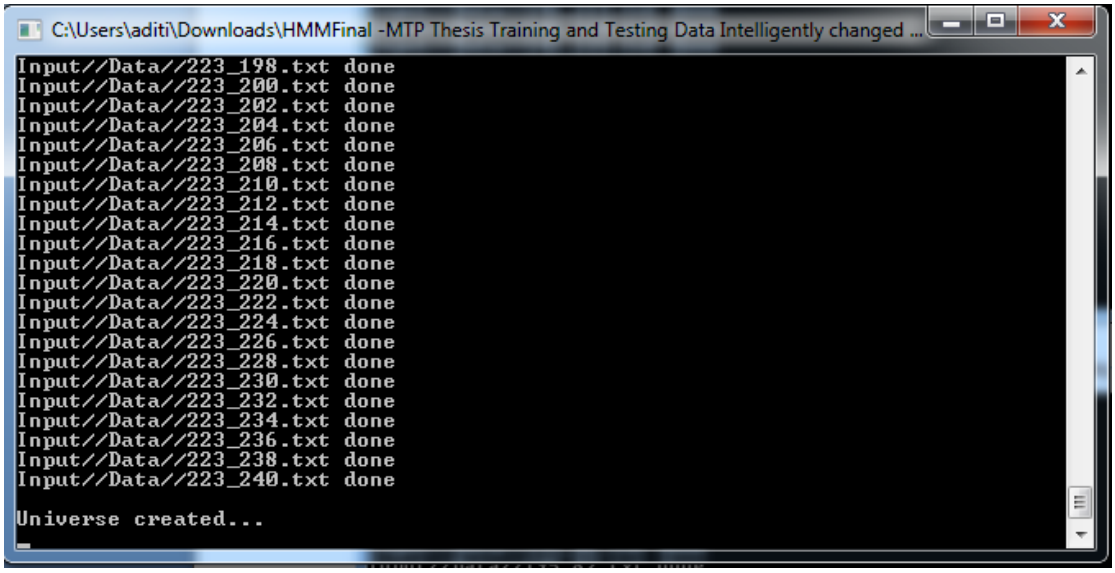


Figure 5.3: Universe Creation

After Universe is created, processing goes on and codebook is generated of 32 size. Figure 5.4 is the screenshot when codebook is created successfully.

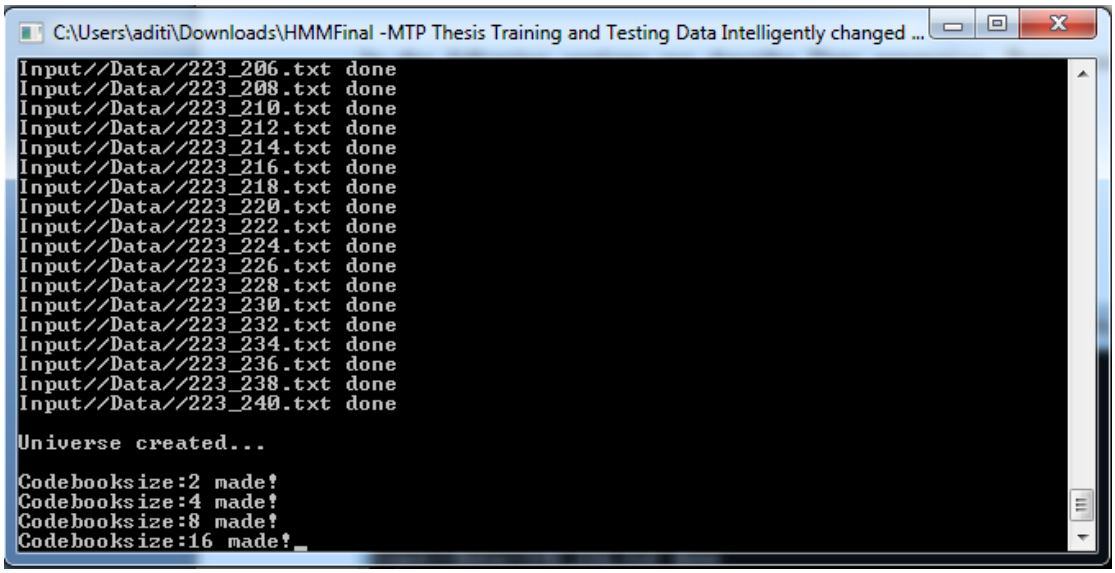


Figure 5.4: Codebook Created Successfully

Next step is creation of observation Sequences for all the instances of data and thus figure 5.5 shows observation sequence being created.

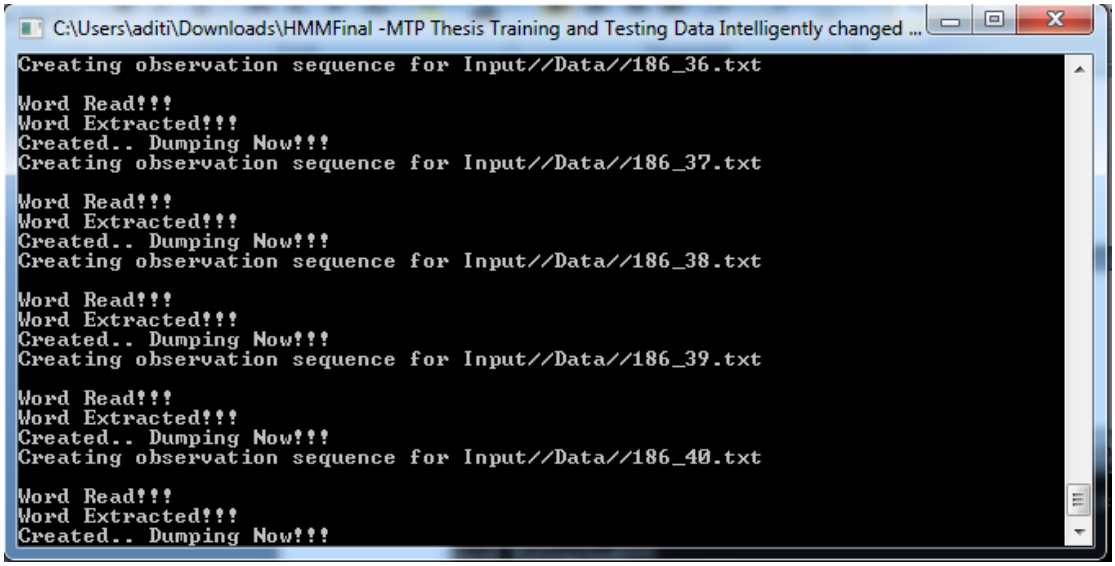


Figure 5.5: Creation of Observation Sequences

Next step is the Training Part. First observation sequences created in last part are being read. For each stroke class hidden markov models are generated after several iterations using Viterbi algorithm and Baum Welch algorithm. Figure 5.6 shows training on data

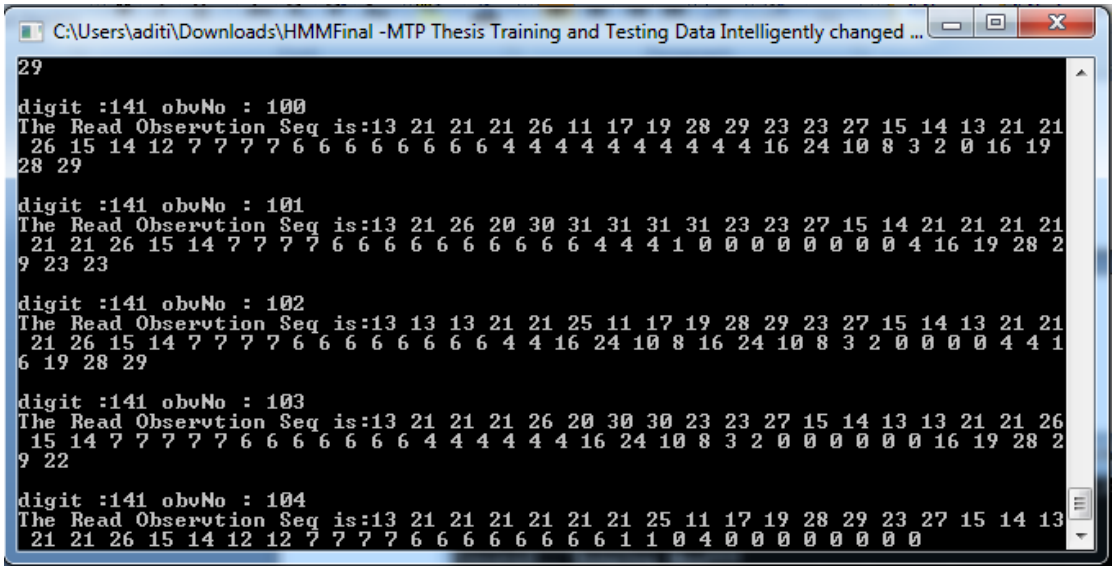


Figure 5.6: training happening on data

Figure 5.7 and Figure 5.8 shows how we proceed to testing. File for testing is fetched and then tested against the hidden markov model generated for respective stroke. Whether is its success or failure both are printed to console.

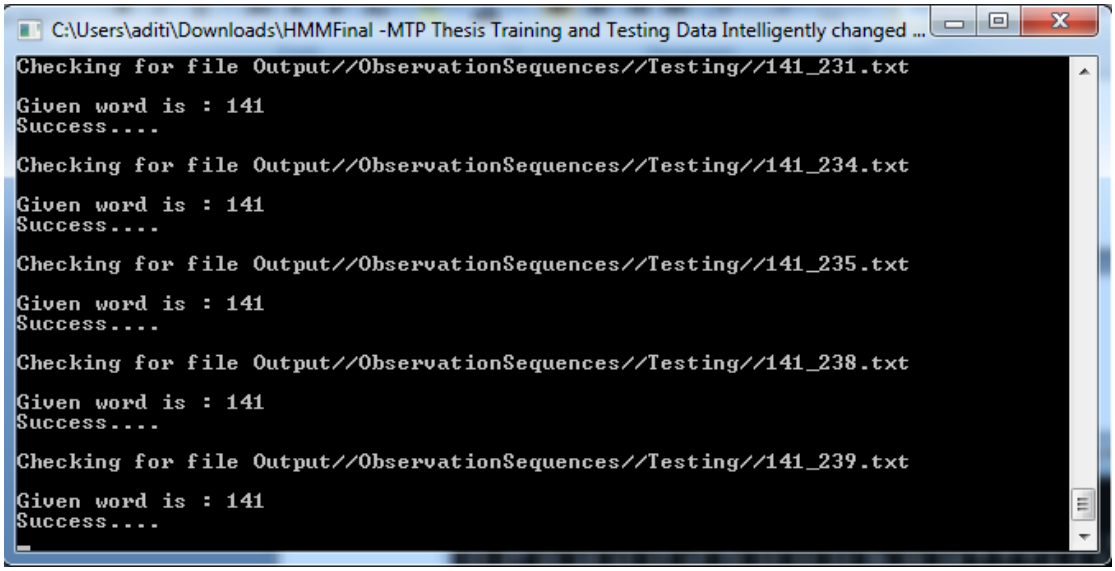


Figure 5.7: Testing happening on Stroke Data

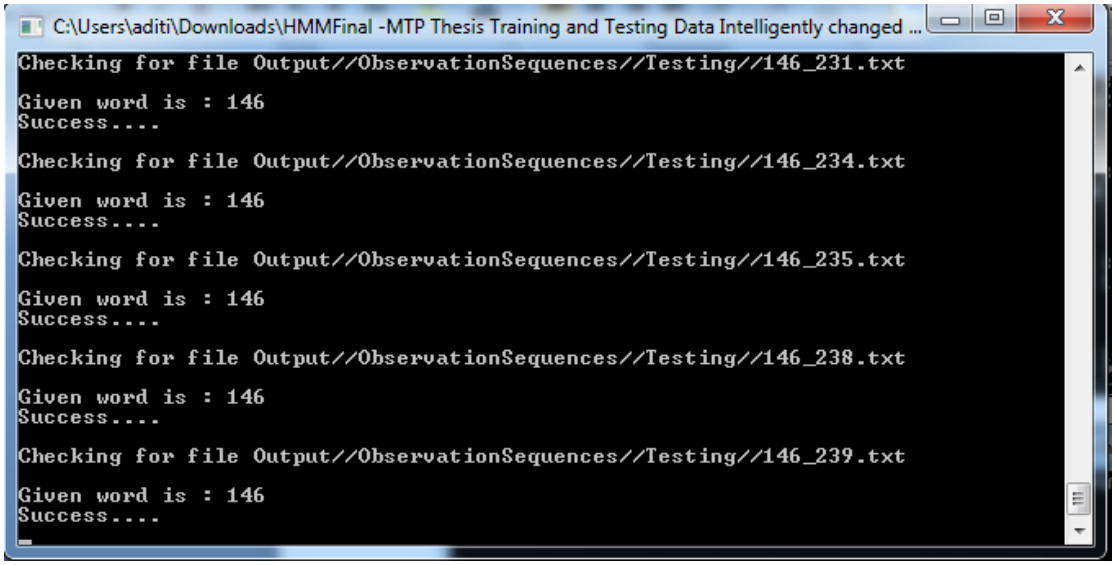


Figure 5.8: Testing happening on Stroke Data

Figure 5.9 shows final output in brief. It sums up the results total test cases we took were 3180. Successful test cases 2756 and failure test cases were 424 so this shows total success rate of 86.66%.

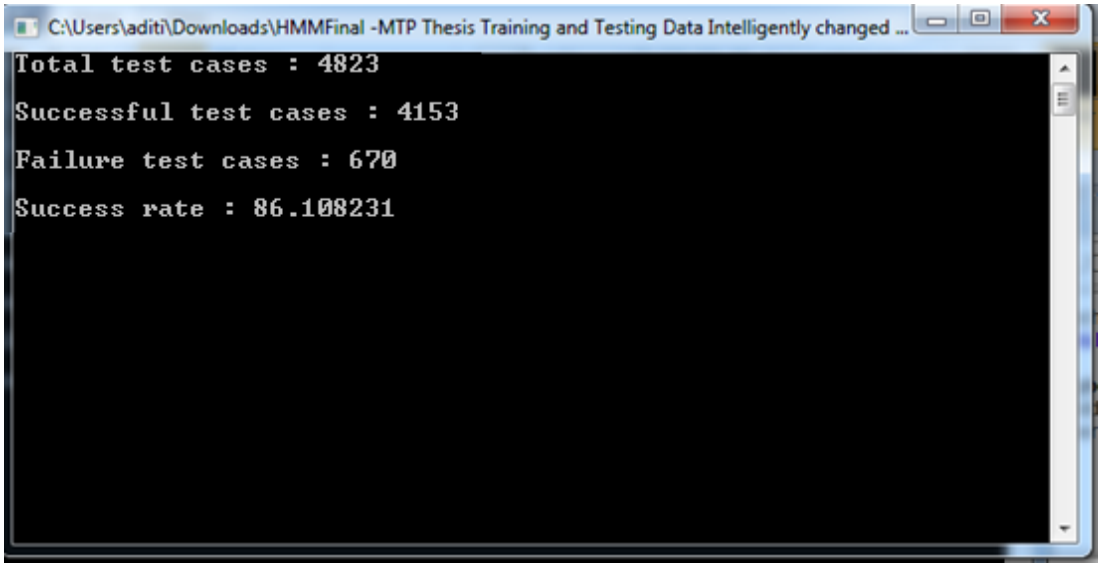


Figure 5.9: Final Output when there are 151 training instances per class

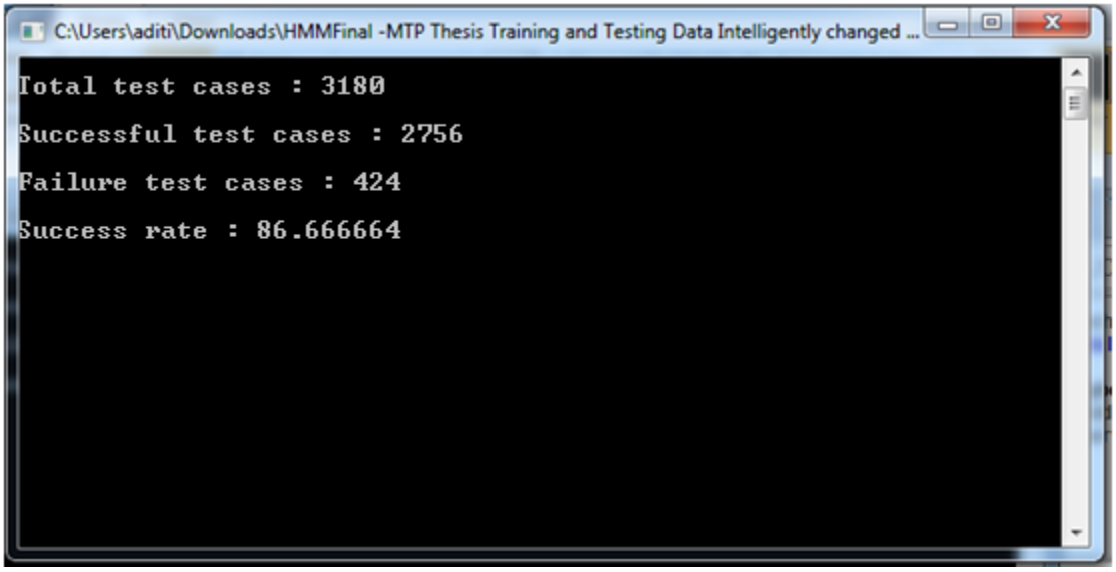


Figure 5.10: Final Output when there are 182 training instances per class

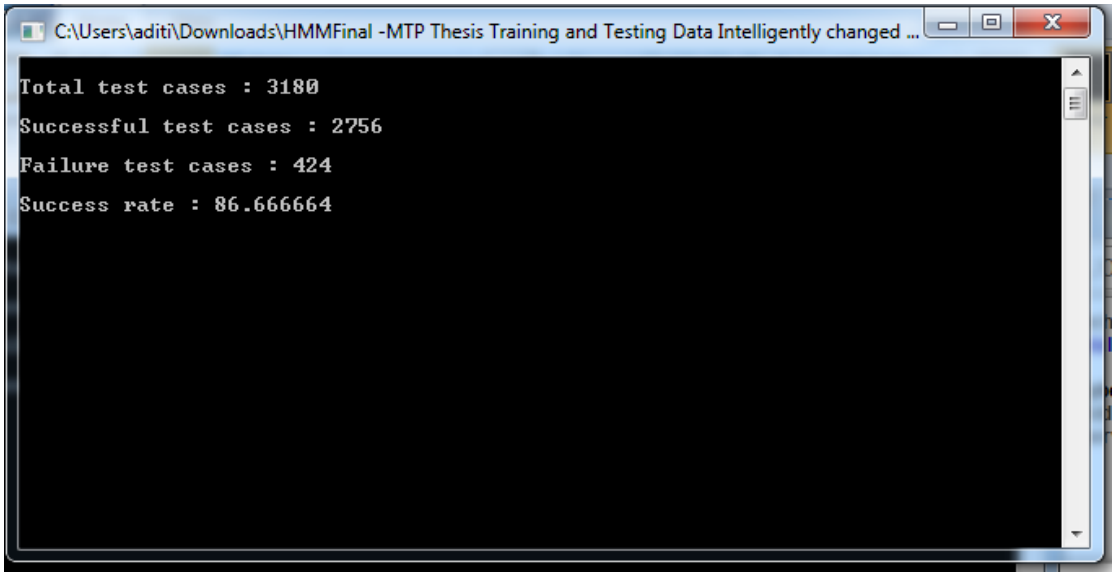


Figure 5.11: Final Output when there are 212 training instances per class

For our implementation, 242 instances for each class (or stroke) were considered i.e. at total of 12826 instances. Three different Training-Testing data combinations are used for experiments as shown below in Table 5.1

In first case 151 instances were used to train and 91 instances were used to test, in second case 182 instances were used to train and 60 instances were used to test and in third case 212 instances were used to train and 30 instances were used to test.

Table 5.1: Table depicting 3 different Testing Training Data combinations and their corresponding accuracies used for experiment

Training Instances (per Class)	Testing Instances (per Class)	Total Test Cases	Total Successful Test Cases	Total Failure Test Cases	Overall Accuracy %age
151	91	4823	4153	670	86.10%
182	60	3180	2756	424	86.66%
212	30	1590	1378	212	86.66%

The system is trained and tested by building HMM models for each class (Stroke). We observed that on increasing training data our efficiency also increases but gradually.

Thus, we can conclude that by increasing training data, the efficiency of the system will improve as it will be able to learn better.

Results are plotted in bubble graph where diagonal bubbles are clearly larger which means most of the strokes are recognized accurately and small dots away from diagonal depict incorrectly recognized strokes as shown in Figure 5.10.

Gaps between consecutive large bubbles on diagonal actually appear because of the invalid class numbers existing between them.

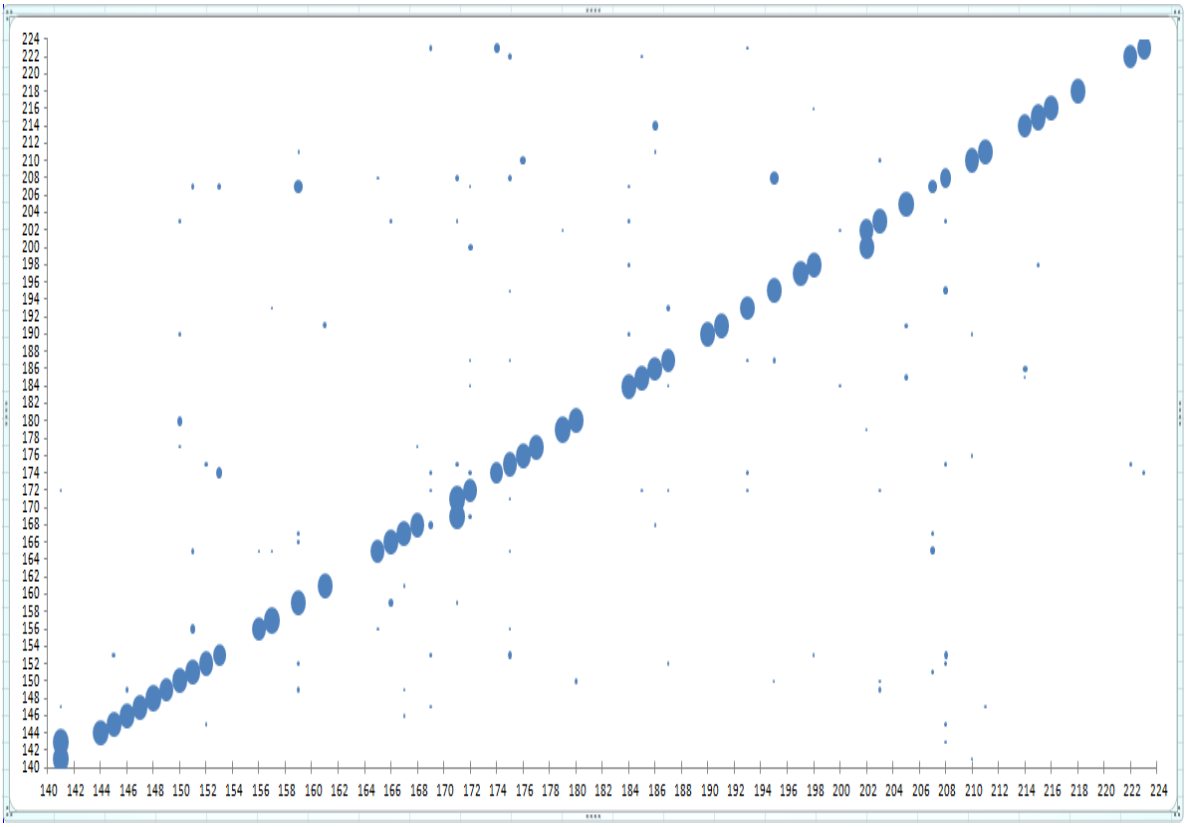


Figure 5.12: Results shown using Bubble Graph

Conclusion

In our work, we tried to recognize handwritten Punjabi strokes. Different users write differently and thus give rise to different shapes of stokes from which characters are formed. So we have trained on multiple instances of strokes, and those strokes are written in different styles by different users. The models are successfully trained on various instances of each class and then tested for other instances of these classes in three different variations. We also noticed that as more training data is considered, the accuracy of the system increased however slowly. Hidden Markov Model worked well for problems like this.

Future Work

In Future we can add more training data and can train models even more accurately which will thus improve accuracy of recognition system. This system can further be used for full Punjabi character recognition system as this makes the fundamental basic building block for that. We can also use other machine learning models and hence can strengthen Punjabi character recognition system.

References

[1] L.R. Rabiner, B.H. Juang, “An Introduction to Hidden Markov Models,” IEEE ASSP MAGAZINE, January 1986.

[2] Karun Verma, Rajendra Kumar Sharma, “Comparison of HMM- and SVM- based stroke classifiers for Gurumukhi script,” Springer, April 2016.

[3] Rabiner, L.R., Levinson, S.E., and Sondhi, M.M., “On the Application of Vector Quantization and Hidden Markov Models to Speaker-Independent, Isolated word recognition,” Bell System Tech. J,Vol. 62, No. 4, pp.1075-1105, April 1983.

[4] Forney, jr., G.D., “The Viterbi Algorithm”, Proc. IEEE, Vol. 61, pp.268-278, March 1978.

[5] Lawrence R. Rabiner, “A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition,” Fellow, IEEE,Vol. 77,No.2,February 1989.

[6] Lawrence Rabiner, Biing-Hwang Juang, “Fundamentals of Speech recognition,”

[7] Baum, L.E., Petrie, T., Soules, G., and Weiss, N., “A Maximization Technique Occuring in the statistical Analysis of Probabilistic Functions of Markov Chains,” Ann. Math. Statistic., 41(1970),pp. 164.

[8] Beatrice Lazzerini, Francesco Marcelloni, “Fuzzy classification of handwritten characters”, 18th International Conference of the North American, pp. 566 – 570, 1999

[9] M Hanmandlu, K R Murali Mohan, Sourav Chakraborty, “Fuzzy logic based handwritten character recognition”, Proceedings of International Conference on image Processing, vol.3, pp. 42 – 45, 2001

[10] Weipeng Zhang, Yuan Yan Tang, Yun Xue, “Handwritten Character Recognition Using Combined Gradient and Wavelet Feature”, International Conference on Computational Intelligence and Security, vol. 1, pp. 662 – 667, 2006

[11] Plamondon R, Srihari SN (2000), “Online and off-line handwriting recognition: a comprehensive survey”. IEEE Trans Pattern Anal Mach Intell 22(1):63–84

[12] Biswas C, Bhattacharya U, Parui SK (2012), “Hmm based online handwritten Bangla character recognition using dirichlet distributions”. In: IEEE 2012 international conference on frontiers in handwriting recognition (ICFHR). pp 600–605

[13] Gernot A. Fink, Szilard Vajda, “Online Bangla Word Recognition Using Sub-Stroke Level Features and Hidden Markov Models”, in 2010 12th International Conference on Frontiers in Handwriting Recognition.

[14] S.K. Parui , U bhattacharya, B. Shaw, “Neural Combination of ANN and HMM for Handwritten Devanagari Numeral Recognition”. Tenth International Workshop on Frontiers in Handwriting Recognition, Oct 2006, La Baule (France), Suvisoft, 2006

[15] Parui SK, Guin K, Bhattacharya U, Chaudhuri BB (2008),”Online handwritten Bangla character recognition using HMM” In: IEEE 19th international conference on pattern recognition, 2008, ICPR 2008. pp 1–4

[16] Jayaraman A, Chandra SC, Srinivasa CV (2007), “Modular approach to recognition of strokes in Telugu script”. In: IEEE ninth international conference on document analysis and recognition, 2007. ICDAR 2007, vol 1. pp 501–505

[17] Aparna K, Subramanian V, Kasirajan M, Prakash GV, Chakravarthy V, Madhvanath S (2004), “Online handwriting recognition for Tamil”. In: IEEE ninth international workshop on frontiers in handwriting recognition, 2004. IWFHR-9 2004. pp 438–443

[18] Bharath A, Madhvanath S (2007), “Hidden markov models for online handwritten Tamil word recognition”. In: IEEE ninth international conference on document analysis and recognition, 2007. ICDAR 2007, vol 1. pp 506–510

[19] Kumar R, Sharma RK, Sharma A (2015), “Recognition of multistroke based online handwritten Gurmukhi aksharas”. Proc Natl Acad Sci India Sect A Phys Sci 85(1):159–168

[20] Sharma A, Kumar R, Sharma R (2010), "Hmm-based online handwritten Gurmukhi character recognition". Mach Gr Vis Int J 19(4):439–449

[21] Pal, U. and Chaudhary, B.B. (2004), "Indian Script Character Recognition, A survey,

Pattern Recognition", Elsevier, pp. 1887-1899.

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

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

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

[25] Almuallim H, Yamaguchi S (1987), "A method of recognition of Arabic cursive handwriting", IEEE Trans Pattern Anal Mach Intell 9(5):715–722

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

[27] Lehal, G. S., Singh, C. (2000), "A Gurmukhi Script Recognition System, Proceeding

of International Conference on Pattern Recognition", Vol. 2, pp. 557-560.

[28] Lehal, G. S., Singh, C. (2002), "A post Processor for Gurmukhi OCR", Sadhana, Vol. 27, Part 1, pp. 99-111

[29] Lehal, G. S., Singh, C. (2006), "A Complete Machine printed Gurmukhi OCR".

[30] Anuj Sharma, Rajesh Kumar, R. K. Sharma, "Online Handwritten Gurmukhi Character Recognition Using Elastic Matching," Image and Signal Processing, 2008. CISP '08. Congress on vol.2, no., pp.391-396, 27-30 May 2008

[31] Anuj Sharma, R.K. Sharma, Rajesh Kumar, "Online Handwritten Gurmukhi Character Recognition", Ph.D. Thesis, Thapar University, 2009 [Online].

[32] D. Sharma, G. S. Lehal, Preety Kathuria, "Digit Extraction and Recognition from Machine Printed Gurmukhi Documents", MORC Spain, 2009

[33] Hu J, Brown MK, Turin W (1996), "Hmm based online handwriting recognition". IEEE Trans Pattern Anal Mach Intell 18(10):1039–1045

[34] Garcia-Salicetti S, Doizzi B, Gallinari P, Mellouk A, Fanchon D (1995), "A hidden markov model extension of a neural predictive system for online character recognition". In: IEEE proceedings of the third international conference on document analysis and recognition, 1995, vol 1. pp 50–53

[35] Bellegarda EJ, Bellegarda JR, Nahamoo D, Nathan KS (1993), "A continuous parameter hidden markov model approach to automatic handwriting recognition", 1993. EP Patent 0,550,865

[36] Liu C-L, Jaeger S, Nakagawa M (2004), "Online recognition of Chinese characters: the state-of-the-art", IEEE Trans Pattern Anal Mach Intell 26(2):198–213

[37] Jager S, Liu C-L, Nakagawa M (2003), "The state of the art in Japanese online handwriting recognition compared to techniques in western handwriting recognition", Doc Anal Recognition 6(2):75–88

[38] Kumar R, Sharma RK (2013), "An efficient post processing algorithm for online handwriting Gurmukhi character recognition using set theory", Int J Pattern Recognit Artif Intell 27(04):1–17

[39] Verma K, Sharma R (2015),” Performance analysis of zone based features for online handwritten Gurmukhi script recognition using support vector machine” in Selvaraj H, Zydek D, Chmaj G (eds) Progress in systems engineering. Volume 330 of advances in intelligent systems and computing. Springer International Publishing, Berlin, pp 747–753

List of Publications

A.Gupta and K.Verma, “Handwritten Recognition in Punjabi using Hidden Markov Model”, “Fourth International Symposium on Women in Computing and Informatics (WCI-2016)”. (Accepted)

Video Link:

https://www.youtube.com/channel/UCzUROhR9Cn2oI0yytiu37gg