On Segmentation of Words from Online Handwritten Gurmukhi Sentences

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

Master of Technology

in

Computer Science and Applications

Submitted by

Devesh Vasantrao Dahake

(Roll No. 601534003)

Under the supervision of:

Dr. R.K. Sharma

Professor

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

THAPAR UNIVERSITY

PATIALA - 147004

JUNE 2017

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled, "On Segmentation of Words from Online Handwritten Gurmukhi Sentences", in partial fulfillment of the requirements for the award of the degree of Master of Technology in Computer Science and Applications submitted in Department of 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 the award of any degree of this or any other University.

Dahake

(Devesh Dahake)

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

RK Sharma 12.7.17

(R. K. Sharma)

Professor,

CSED

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 not have been possible without the encouragement and able guidance of my supervisors **Dr. R. K. Sharma**. I thank my supervisor for their time, patience, discussions and valuable comments. Their enthusiasm and optimism made this experience both rewarding and enjoyable. Their discipline and sincerity towards work, teaches sincerity is more important than seriousness in life.

I am equally grateful to **Dr. Maninder Singh**, Associate Professor and Head, Computer Science & Engineering Department, a nice person, an excellent teacher and a well – credited researcher, who always advised me with his valuable suggestions. I am also grateful to **Dr. Sanmeet Bhatia**, P.G. Coordinator for giving motivation and inspiration to complete this thesis 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 equipped with net facilities, immensely useful for the learners to equip themselves with the latest in the field.

I am also thankful to the entire faculty and staff members of 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 least, I would like to thank my family whom I dearly miss and without whose blessings none of this would have been possible. To my parents, I own thanks for their wonderful love and encouragement. I would also like to thank my brother and sister, since they insisted that I should do so. I would also like to thank my close friends for their constant support.

Date: June, 2017

Place: Thapar University, Patiala

(Devesh Dahake)

ABSTRACT

Keyboard based devices face many problems relating to hardware failure or damages occurring due to multiple users handling a device and decaying of the device as hardware gets old. So there is a need to provide other ways of communication between machine and human beings, it is done through speech input and handwriting input. Handwriting is a natural way of communication between machine and human with pen-based technology emerging rapidly.

This thesis deals with word segmentation from online handwritten Gurmukhi sentences. For recognizing sentences, proper segmentation of sentences into words is very important. The method proposed in this thesis for word segmentation considers online data at stroke level in which white spaces between the words are not explicitly known. This work, as such, focuses on the recognition of online Gurmukhi sentences. Thresholding approach is used for segmenting words from a sentence. The basic handwriting unit of a sentence is a stroke. One or more strokes form the words and creation of words is nothing but a sentence. In the proposed approach, vertical gap between the strokes are first located and then based on the maximum threshold value the word is extracted from the sentence. Testing of proposed approach has been performed on 200 sentences. A segmentation accuracy of 91.0% has been achieved with the use of the proposed algorithm for segmenting sentence into words for the online Gurmukhi script.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

HMM	Hidden Markov Model
OHWR	Online Handwriting Recognition
PC	Personal Computer
PCA	Principal Component Analysis
PDA	Personal Digital Assistant
SVM	Support Vector Machine
FSA	Finite State Automaton
RBF	Radial Basis Function
SVC	Support Vector Classification
SVR	Support Vector Regression
DOM	Document Object Model
XML	Extensible Markup Language
CSV	Comma Separated Values

CHAPTER 1

INTRODUCTION

These days, a number of researchers are involved in handwritten script recognition. A very handy and useful work has been done in online handwritten Gurmukhi script recognition (Sharma et al., 2008). Segmentation is one of the important phases in online handwritten script recognition process to get meaningful recognized data as an output. We work in regional languages to communicate effectively with the machine. As we know, there are only two ways to communicate with the machine, through speech, and through writing. Research have shown that writing is a better and effective way to communicate with the computer.

Over a period of time, many techniques have been developed to segment offline data, but little work has been done on online data. Word segmentation is a difficult task in the field of natural language programming. So, the methodology presented in this paper focuses on online handwritten Gurmukhi script data. We first collect online handwritten stroke level Gurmukhi script data for sentence and perform different experiments on online data to segment words.

A number of techniques are studied to segment online stroke level data and character level data, but to segment words from sentences has not been studied. While writing on writing pad the pen tip position captures the data when the pen moves from one position to another and the movement of the pen when the writer put's pen on the writing pad till the pen lifts up is one stroke. Such a digital pen having sensors captures the data stroke wise and it is easy to recognize the online data at stroke level. The various methods are available for segmenting the words, but are restricted to offline handwriting recognition only. Techniques like vertical histogram projection on machine printed character, Otsu method, heuristic method and feature detection of word images are available for segmentation of offline handwritten script data.

Sentence segmentation is a difficult task in the field of natural language processing. Improving segmentation accuracy to the highest level has always been a motivation for researchers. The script, like Gurmukhi, has always been difficult to

handle because of the unpredicted writing style of different users. There are many languages like Gurmukhi for which inputting data to a computer is difficult. So there is a need to provide a communication between machine and human beings. As mentioned earlier, there are two ways to provide communication in between machine and human and that is through writing and speech. For those who are not able to write, speech recognition is the best way for communication with a machine, but it is not very efficient in a noisy environment. Here, we input data to the computer through handwriting. But there are many problems in online handwriting as there are many variations between different users of handwriting. So to attain high accuracy in segmentation is always a difficult task.

1.1 Foundations

Handwritten script data can be taken from images and from writing pads. Handwriting recognition is of two types i.e. offline handwriting recognition and online handwriting recognition. In offline handwriting recognition system, the scanned images are passed to the recognition systems. There are many tools available to segment offline data. In online handwriting recognition, automatic conversion of handwritten data to its corresponding pixel value has been carried out with the help of digital pen having a sensor to capture consecutive *x* and *y* coordinate of each pixel. Since touch screen devices are emerging rapidly, handwriting recognition of all languages also attracts researchers towards them. The present work focuses word segmentation from sentences of online handwritten Gurmukhi script data.

1.2 Gurmukhi Script

Gurmukhi is the script of Punjabi language which is widely spoken across the globe. The word Gurmukhi means "from the mouth of Guru". Gurmukhi script which was first used by second guru, Guru Angad, is being used by over 130 million people across the world. Salient features of this script (Sharma *et al.*, 2008) are:

i. The Gurmukhi script is written from left to right.
ii. The horizontal line at the upper part of a character called the headline.
iii. Gurmukhi word can be partitioned into three zones (Figure 1.1). The zone above the headline is called an upper zone where vowels and sometimes a portion of vowels reside. The second zone is called a

middle zone where consonant and subpart of vowels dwell. This is the busiest (most involved) zone. The third zone is called a lower zone and it is located just below the middle zone.

iv. Gurmukhi script has 12 *matras* and 41 characters shown in Figure 1.2 (Verma and Sharma, 2016).

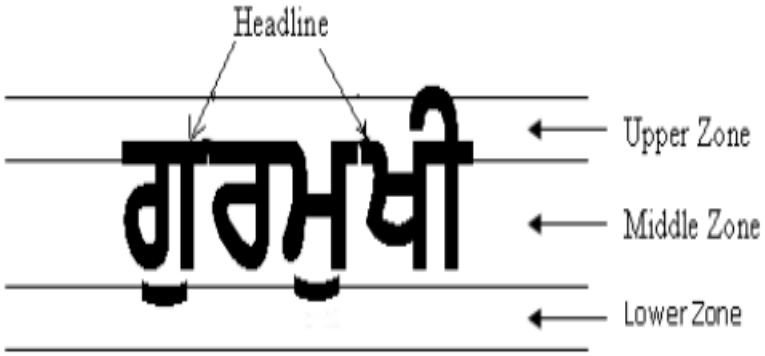

Figure 1.1: Three zones in Gurmukhi script.

ੳ	ਅ	ੲ	ਸ	ਹ	9 Matras	◌ਾ
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		◌ੈ
ਖ਼	ਗ਼	ਜ਼	ਫ਼	ਗ਼		◌ੋ
ਜ਼	41 Consonants					◌ੌ
◌ੰ	◌ਂ	◌ੱ	3 Nasal Symbols			

Figure 1.2: Gurmukhi characters and *matras*.

1.3 Online Handwriting Recognition System

Online handwriting recognition systems have six stages viz. data collection, segmentation, preprocessing, feature extraction, recognition, and post-processing Preprocessing involves further four stages viz. size normalization and centering, interpolation of missing points, smoothing and resampling shown in Figure 1.3.

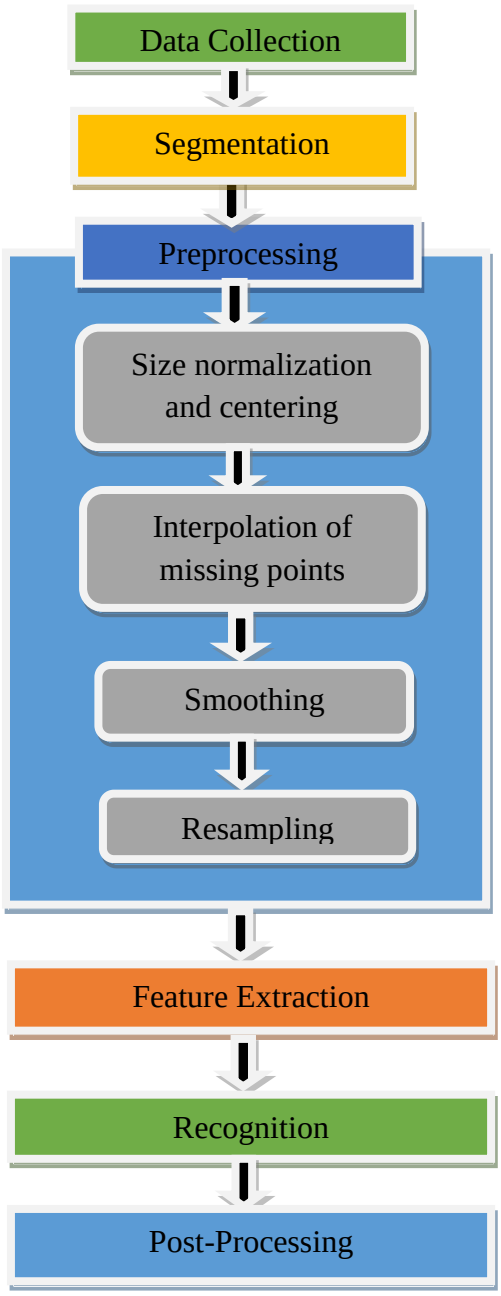

Figure 1.3: Phases of OHWR system.

1.3.1 Data Collection

In data collection phase, data are collected from a pen computing device or digitizer. Such devices have a sensor that uses a pen-tip position, pen-up and pen-down switching to capture *x* and *y* coordinates. It records the sequences of consecutive *x* and *y* points of the coordinate plane accurately.

For the present study, we used tablet PC to collect online handwritten data. Some commonly used pen computing devices are Cross Tech3+ Multifunction Pen, Wacom CTL 471/K0-Cx Tablet, Apple 12.9-inch IPad PR. Figure 1.4 shows devices which captures the movement of the pen.

Figure 1.4: Pen computing devices.

1.3.2 Segmentation

It is an important stage in online handwriting recognition and the accuracy of recognition depends on proper segmentation. Data is represented at a stroke or character level so that we can contemplate stroke and character separately. Segmentation is of two types; internal segmentation and external segmentation (Figure 1.5). Internal segmentation is performed during the recognition process and external segmentation is performed prior to recognition. External segmentation provides good interactivity and saves the computation time.

Many of the segmentation algorithms are available for offline data, but only limited work had been done on online data and that too only for segmentation performed on a word to separate strokes and characters. Line segmentation into words for online data has not been studied for Gurmukhi script.

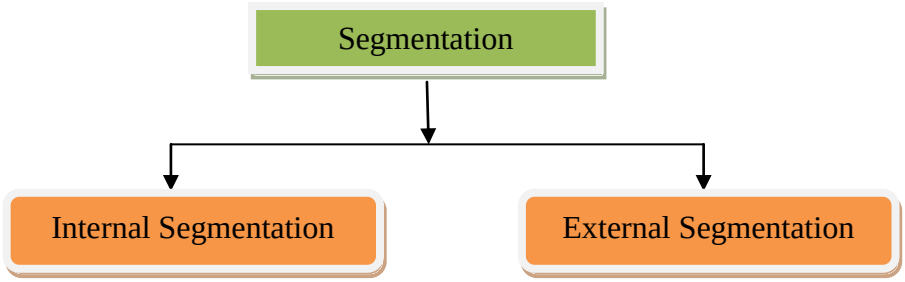

Figure 1.5: Types of Segmentation.

1.3.2.1 External Segmentation Technique

As discussed earlier, the external segmentation procedure is performed prior to recognition. Means, it is always carried out before recognition procedure. This technique, as such, saves computation time and provides good interactivity.

Sometimes, it is necessary to segment data before preprocessing and recognition stage, this strategy comes in the category of external segmentation. External segmentation can be performed on both online and offline data. External segmentation performed on online handwritten Gurmukhi data is discussed in Chapter 1.6. For segmenting online data using an external segmentation technique, we need the online captured data of the script. This online data is captured by pen computing device having sensors to capture consecutive *x* and *y* coordinate. External segmentation also performed on offline data. Offline data are of printed or handwritten documents. Many techniques are available for segmenting offline data and it is discussed in Chapter 2.1.

1.3.2.2 Internal Segmentation Technique

Internal segmentation can be used for offline and online data. The segmentation performed during the recognition procedure is called as internal segmentation. When a writer writes on pen computing device, the system recognizes online handwritten script data and output shows properly segmented recognize data. Then it is an internal segmentation technique. The internal segmentation algorithm to recognized online handwritten Gurmukhi data is discussed in Chapter 4.

Internal segmentation works during recognition. In this case, one cannot get a dataset to performed segmentation. Segmentation carried out internally with recognition procedure. This makes procedure too tedious to understand what exactly happens internally. So, it is good to use the advantages of external segmentation to perform segmentation using internal segmentation strategy. To segment offline data, there are many standard techniques available. One such technique is an Otsu method.

1.3.3 Preprocessing

Preprocessing in online handwriting recognition is applied to minimize noise and contortions in the input text, which happens due to software and hardware restrictions. This noise contains unequal distances of points from neighboring locations, the inconsistent size of text, left and right bend in handwriting, missing points and jitter in the text. A novel stride for rearrangement of recognized stroke for the recognition of online handwritten Gurmukhi characters is proposed by Sharma et al., (2009). The proposed technique consists of stroke's identification as dependent and major dependent strokes, the rearrangement of strokes, according to their positions, and the combination of strokes to recognize characters.

In online handwriting recognition process, it specifically includes size normalization and centering, interpolation of missing points, smoothing, slant correction and resampling of points.

1.3.3.1 Size Normalization and Centering

The handwriting style of every user is distinctive, some of the writers write in small size, while others write in large size. Therefore, it is needed to convert them to a uniform size. Some of the writers write at the border of writing pad, so the centering method is required. The following steps are used for normalization,

1.3.3.1.1 Translation

Translation is used to locate every stroke uniformly with respect to its origin to remove the translation variation and it is carried out by the addition of translation factor and its originating location. Translation basically moves a point starting with one position, then onto the next position on the screen. We would be able to move the point in two-dimensions by including translation coordinates or translation vectors

'*tx*' and '*ty'* from its original point *x* and *y* to get new points as *x′* and *y′* shown in Figure 1.6.

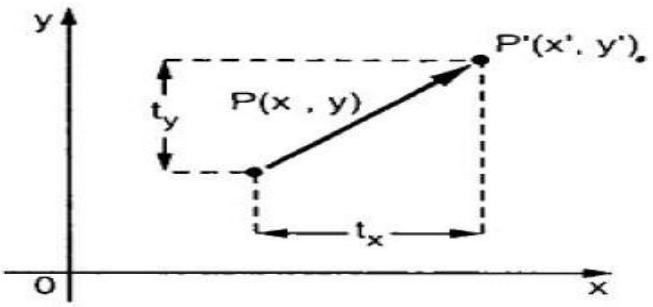

Figure 1.6: Translation of points in 2D.

From above figure, we can write that-

$$x' = x + tx$$

$$y' = y + ty$$

1.3.3.1.2 Rotation

Rotation is used to rotate an object with a specific angle from its inception. In the underneath Figure 1.7, we would be able to shift line OP to OP′ by rotating through an angle theta, so that P(*x*, *y*) goes to P′(*x′*, *y′*).

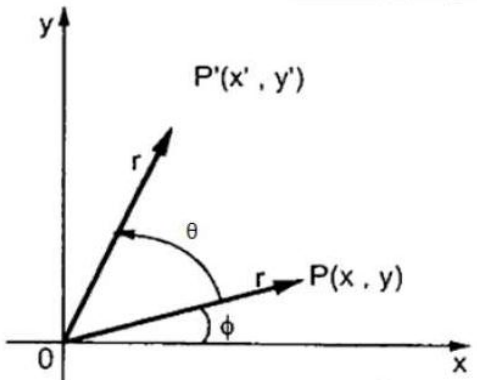

Figure 1.7: Rotation of points in 2D.

From above figure and using standard trigonometric, we represented origin point as

$$x = r\,cos\,\emptyset \text{.....(1)}$$

$$y = r \sin \emptyset \text{.....(2)}$$

In the same way, we can represent P′(x′, y') as-

$$x' = r\cos(\emptyset + \theta) = r\cos\emptyset\cos\theta - r\sin\emptyset\sin\theta \ldots\ldots (3)$$

$$y' = r\sin(\emptyset + \theta) = r\cos\emptyset\sin\theta + r\sin\emptyset\cos\theta \ldots\ldots\ldots (4)$$

Substituting equation (1) and (2) in (3) and (4) respectively,

$$x' = x\cos\theta - y\sin\theta$$

$$y' = x\sin\theta + y\cos\theta$$

1.3.3.1.3 Scaling

Scaling is utilized to change the measure of an object and is necessary in an online handwriting process, because every writer writes in different styles and size. Someone may write in small size and someone may use a particular stroke or character. And hence there is a need to normalize an object to a uniform size and it is carried through scaling factor. In scaling we like to compress or to expand the dimension of the object to a uniform size. Scaling can be achieved by multiplying the scaling factor with its original coordinates to get desired uniform coordinates.

Let us take *A*, *B*, *C* and *D* are the original coordinates and *Sa*, *Sb*, *Sc* and *Sd* are the scaling factor and hence by multiplying original coordinates with scaling factor to get new scaled coordinates as *A′*, *B′*, *C′* and *D′* respectively shown in underneath Figure 1.8.

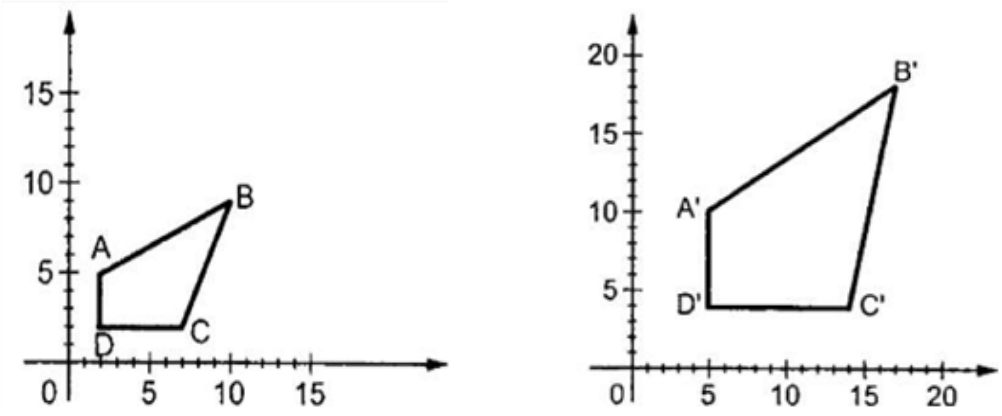

Figure 1.8: Scaling of the object.

As shown in above figure, we represented scaling as follows-

$$A' = A \times Sa$$

$$B' = B \times Sb$$

$$C' = C \times Sc$$

$$D' = D \times Sd$$

1.3.3.2 Interpolation of Missing Points

Writing speed of every user is different. Writing on writing pad at higher speed may miss some points. This missing point is interpolated using many techniques like Bezier curve.

1.3.3.3 Smoothing

Smoothing in an input text is required to remove jitter in handwriting. Smoothing is averaging a point with its neighbor.

1.3.3.4 Slant Correction

Different writer has a habit of writing at different slope. Some writings lean to the left while others lean to the right. So, slant correction is applied to normalize the slope. Therefore, it's an important process in online handwriting recognition.

1.3.3.5 Resampling

Resampling, means to make adjacent points equidistant from their neighboring points. To make them equidistant will also help to extract feature.

1.3.4 Feature Extraction

Feature extraction technique is an essential step in handwriting recognition. As best feature provides the best result, it is an important to extract efficient and essential feature for data representation and processing. Some of the features are more suitable for one script, while others are more suitable for another script. There are two sorts of the features; high-level feature and low-level feature. The high-level feature includes headlines, dots, straight line, crossing and loops and low-level feature includes slant, directions, area, slope, and positions.

Some techniques are available for extracting features. DFT (Discrete Fourier Transform) based feature extraction technique to recognize online handwritten Gurmukhi stroke is proposed by Aggarwal and Sharma, (2016). To attain high recognition accuracy, computation of good feature is very important. To extract good features, the authors employed Discrete Fourier Transform (DFT). In their work, they used 86 stroke classes with 75-100 variations of each stroke. A total of 8408 stroke samples were considered. They have utilized LibSVM with RBF kernel and 11-fold cross validation approach for recognition.

Baseline detection is also important for recognition of the word. A new methodology is proposed to detect the baseline in Arabic handwritten words (Pechwitz and Margner, 2002). Baseline estimation is an inception for word and precondition for preprocessing, segmentation and recognition process. In this paper, they introduce a technique that is applicable to polygonally approximated skeleton handling. They find the feature in skeleton using an algorithm and preprocessed using linear regression.

A new framework has been proposed for recognition of unconstrained Bangla vocabulary using HMM (Samanta et al., 2014). The word is first segmented into sub-strokes, but instead of recognizing sub-strokes, they focused on recognition of the whole word. In their work, they consider a linear and circular feature. The Gaussian distribution is utilized for the linear feature and Von Mises distribution is utilized for the circular feature. They implement smoothing of HMM constraint to evade potential over-fitting and poor generalization. They consolidate the recognition result of two HMMs: one in view of the input sample in expected sort and the other considering the sample in the invert sort.

Swethalakshmi et al. (2006) have proposed a framework for recognition of online handwritten Devanagari and Tamil characters using SVM. They also discussed the preprocessing steps along with feature extraction method proposed for better recognition accuracy. The result obtained from their work shows that the unfailing classification is achievable using SVM.

1.3.5 Recognition

In online handwriting recognition, recognition is an essential stage, where the classifier classifies input data and matches it with the associated stored class file. The input vector of stroke matches with the stored class of feature vectors, which

determines the recognition accuracy. There are many classifiers available, like naïve based, Support vector machine (SVM), Hidden Markov Model (HMM), neural network, etc. Normally for pattern recognition, text processing, etc., Support vector machine is highly recommended.

Support vector machine (SVM) is the supervised learning model which is utilized for classification and separating the classes in feature space. It is regularly and popularly used for image processing, text processing, and iris recognition, etc. The name itself hint to providing support to something, so let's take an example where a person wants to walk from the bridge where a bridge doesn't have a side rope for a person to get balance over there. So, in this situation, a normal person likes to walk from the center of the bridge instead of walking across the boundary of the bridge. Support vector machine works exactly same as it likes to separate different class of data in such a way that we can get large margin. To separate different class of dataset, SVM draws the hyperplane in such a way that it efficiently separate dataset with a large margin. A margin is a distance from hyperplane to the nearest data point of the dataset shown in underneath Figure 1.9. Larger the margin minimum will be the generalization error occurs.

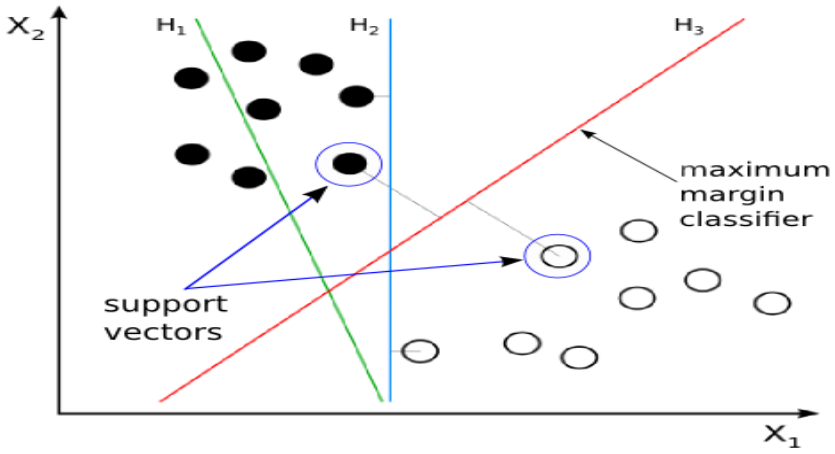

Figure 1.9: Support vector machine (SVM).

In addition, to performing classification on the linear dataset, it also performs classification on the non-linear dataset using a kernel trick. There are four different kernels on which SVM works and they are,

1) Linear kernel
2) Polynomial kernel

3) Radial basis function (RBF) kernel
4) Sigmoid kernel

As it is the supervised learning model, the dataset should be labeled. If dataset is not labeled then supervised learning is not required. For such dataset, we should use unsupervised learning methods. For this, we have to form a group of clustered data by using unsupervised learning algorithm, which is also called support vector clustering.

Support vector machine is used for classification and regression. There are five SVM listed below,

1) c SVC
2) nu-SVC
3) One-class SVM
4) Epsilon SVR
5) nu-SVR

The fourth and fifth types of SVM are used for regression and called as support vector regression (SVR).

1.3.5.1 Techniques for Recognition

The elastic matching technique is proposed for online handwritten Gurmukhi character recognition (Sharma et al., 2008). Recognition of character is done in two stages. In the first stage, recognition of strokes is done and in the second stage, recognition of character is done based on the recognized strokes.

A new technique is proposed for database creation and recognition of online handwritten isolated characters of Bangla script (Mondal et al., 2009). It depicts a plan for extraction of sub-strokes from the online examples written by hand Bangla characters, which are notably cursive in shape. They incorporate another feature vector to be processed for each sub-stroke. They performed recognition using two classifiers, viz. Hidden Markov Model (HMM) and the nearest neighbor classifier in view of Dynamic Time Warping (DTW). The second classifier outflanks the HMM-based classifier.

Belhe et al. (2012) have proposed an approach performs the recognition of online handwritten isolated Hindi words utilizing a blend of HMMs based on Devanagari

symbols. The encompassing assignment of Akshara segmentation, symbol detection, and resulting word recognition is the focus of their work. Utilizing online stroke information for proposing symbol candidates and getting HOG attribute set by comparing with their corresponding pictures will give us recognition which is free of stroke arrangement and stroke shape varieties. The proposed framework is appropriate to unconstressed handwriting. They extracted symbol from 60,000 words for training and tested the 140 symbol-HMM models. The framework is intended to yield at least one competitor words to the client, by following different tree ways under the condition that the symbol probability at each node is above the threshold. The tests are performed on 10,000 words yield an accuracy of 89.0%.

Aparna et al. (2004) proposed a methodology for recognition of online handwritten Tamil character. A structure or shape-based portrayal of a stroke is utilized, in which a stroke is represented to be a string of shape feature. Utilizing this string representation, an obscure stroke is recognized by contrasting against the database, using string matching technique. Character end is resolved by utilizing FSA. A full character is recognized by distinguishing all the part of strokes.

1.3.6 Postprocessing

Postprocessing phase is used to correct misclassified stroke using linguistic knowledge. The post-processor can be used to obtain the estimates of smaller as well as a larger linguistic unit such as words.

1.4 Problems in Online Handwriting Recognition System

The handwriting recognition system is basically of two types, offline handwriting recognition system and online handwriting recognition systems. The document is scanned first and then it is recognized by the computer in offline handwriting recognition systems. In online handwriting recognition system, the system recognizes handwriting while the text is being written. Online handwriting is an effective way where problems can be spotted by the user at the time of recognition. The system can be modified to correct such misclassified stroke and detect the correct character. Online handwriting recognition systems capture the set of strokes and its associated consecutive sequence of points for each stroke. The process by which computer

recognizes character written by hand is called a natural handwriting recognition system.

1.4.1 Changes in Handwriting Pattern

There are many changes in writing style and their pattern. Change occurs due to writing speed, character size, their style, and pattern of writing. Variation in individual writing also depends on the mood of the user, it is due to writing from long time or writing on dissimilar devices. The shape of the stroke and character also depend on the word and also on the order and the direction of the stroke. The variations in Gurmukhi character written by a different user and written by same user shown in underneath Figure 1.10 and Figure 1.11 respectively (Aggarwal and Sharma, 2016).

Figure 1.10: Variations in Gurmukhi characters written by five users.

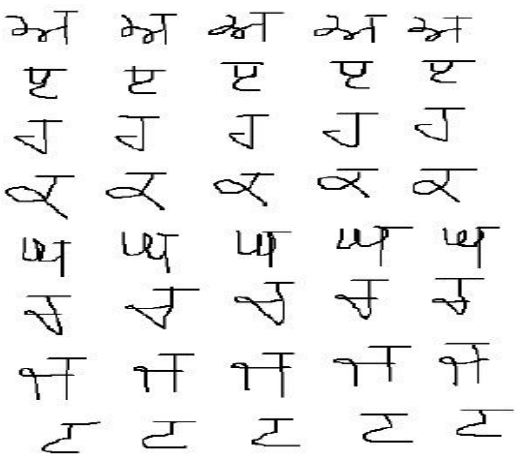

Figure 1.11: Variations in Gurmukhi characters written by the individual user.

1.4.2 Material and Personal Factors

Material factor depends on the hardware device, and the level of comfort provided to the writer. Personal factor depends on the individual user as a writer could be left handed or right handed and could result in different variations in their writing. Both left and right handed writers use different slant direction and position in handwriting.

1.4.3 Writer-Dependent vs. Writer-Independent Recognition System

Both writer-dependent and writer-independent system reflects their results in recognition accuracy. In writer dependent recognition system, the machine remembers the writing style of the writer and can use the data for better recognition of the characters written by that writer. Hence the recognition accuracy will be much better in writer dependent recognition system. But in case of writer-independent recognition system, the system does not give weight to the individual style of the writer. The writer-independent handwriting recognition system is more difficult to develop in comparison to the writer-dependent handwriting recognition system, because it checks all the aspects of handwriting and likely to store all variations of writing.

1.5 Gurmukhi Word Segmentation using External Segmentation

As mentioned earlier, internal segmentation is performed during recognition and external segmentation performed prior to recognition. External segmentation also saves computation time and provides good interactivity. The advantage of external segmentation can also be used while performing internal segmentation during recognition. It is, therefore, imperative that segmentation algorithm should be tested for its accuracy before running online handwriting recognition procedure.

The testing procedure involves using different writers and collecting their sample writings in XML file. The system does not recognize explicit values of white spaces and is a drawback of this XML file. Therefore, the input file is read in x and y coordinates of consecutive stroke. The sample data file is shown in Figure 1.12.

Figure 1.12: Input dataset of stroke wise samples.

As mentioned earlier, the captured data from the pen computing device are stored in the XML file. This XML file is parsed by the DOM parser to get stroke wise data stored in the data structure. The proposed segmentation algorithm described in Chapter 4 is applied on a given dataset. The segmented result containing word-wise data are then stored in CSV file. This CSV file is then given to handwriting recognition systems. The structure of CSV file containing segmented words is shown in Figure 1.13.

Figure 1.13: Output file contains word wise data.

The above CSV file contains word-wise segmented data. Each word is recognized using given id. As shown above, the id 1000 belongs to one sentence and under one id multiple words are present. Each word is recognized till recognition systems get "#" symbol. The recognition results with and without segmentation are shown in Figure 1.14 and 1.15 respectively.

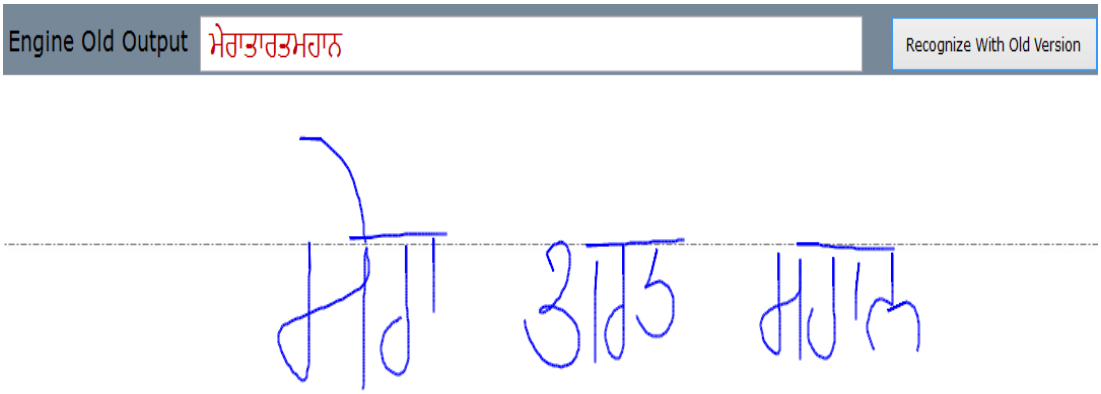

Figure 1.14: Recognition without segmentation.

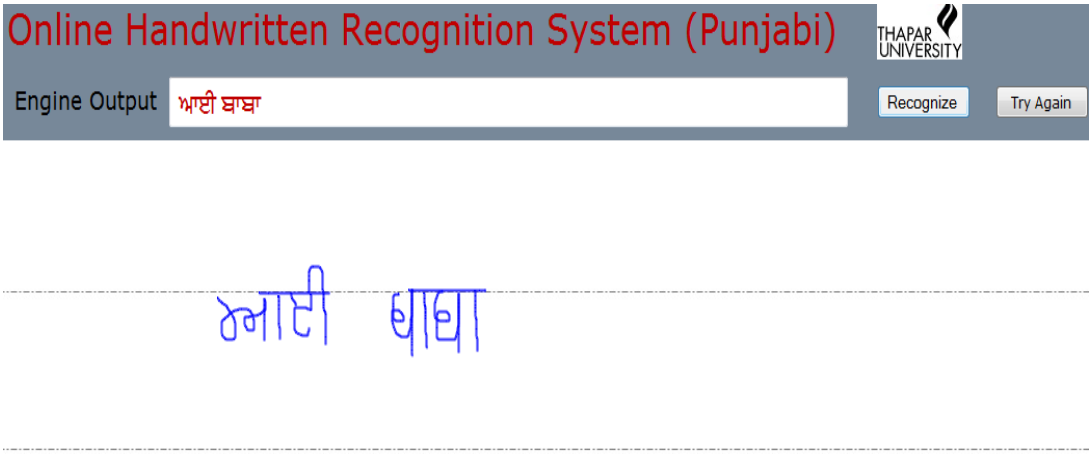

Figure 1.15: Recognition with segmentation.

There are three different phases in the word recognition process using external segmentation approach and they are as follows,

1) Data collection
2) Segmentation procedure
3) Recognition

The detailed work and algorithm are described in Chapter 4.

1.5.1 Problems Arising while Reading Dataset

The problem arises due to dataset description. The stroke wise data are stored in XML file shown in Figure 1.12. The white space values between the words are not given explicitly and it is difficult to judge where exactly the words are ended. The second problem is to create CSV file for more than one sentence, which contains word-wise segmented data. Both the problems are solved by observing given stroke wise dataset carefully. Using the DOM parser the XML file is parsed and values of stroke wise data are stored in the data structure. Now, it is easy to check variations in the value of *x*-axis and alternately decide how to apply segmentation algorithm on a given dataset.

1.6 Thesis Outline

The purpose of this work is to segment words from online handwritten Gurmukhi sentences.

In the present Chapter, an overview of Gurmukhi Script, phases of an Online handwriting recognition system, problems in handwriting recognition and Gurmukhi word segmentation using external segmentation is discussed. Chapter 2 describes the literature survey on segmentation of online and offline handwritten script data. Problem statement and significance of the proposed work described in chapter 3. Chapter 4 describes the segmentation approach and algorithm. The result obtained by applying the proposed algorithm is discussed in chapter 5. Chapter 6 describes the conclusion obtained from the present study and also discussed future work problems.

CHAPTER 2

LITERATURE SURVEY

The important and handy work has been done on segmentation. A lot of work has been done on the segmentation of offline data in the form of handwritten and printed images documents. To know the segmentation process in online handwritten data, the segmentation done on offline data is also important to know.

2.1 Work Related to Segmentation in Offline Recognition System

Segmentation in offline recognition systems is applied on handwritten and printed document images. Several techniques are available for segmenting offline data. Recognition of offline handwritten script data utilizes dataset as a printed or handwritten document.

Wang et al. (2011) has presented dynamic text line segmentation for real-time recognition of Chinese sentence. Utilizing SVM, they discover a connection between progressing stroke and past text-line. The progressing stroke decided to go with a past text line or to shape another line.

Yin et al. (2009) has presented a new approach for line segmentation using minimal spanning tree clustering with distance metric learning. The associated parts of images are gathered together in a tree and frame a cluster, segmentation accuracy improve by the distance metric method. They achieved correct text line detection rate as 92.0%.

Palakollu et al. (2011) have presented a technique for segmenting handwritten Hindi text. In this paper, their main motive is to detect a base line and header line correctly, so that, they can divide the line correctly. Before calculating base line and header line, the average height of the line is estimated. They performed their experiment on 500 lines and achieved 93.6% of segmentation accuracy.

Zheng et al. (2004) have presented another approach for machine printed Arabic character segmentation utilizing a vertical histogram algorithm, and segmentation

based on the upper line, base line and lower line values is performed from the horizontal histogram. In this work, author divides words into character and they achieved a segmentation rate as 94.0%.

Jesper Durebrandt (2015) has presented a segmentation approach for both word and line, to segment line, they apply the vertical projection profile to calculate base line, middle line, and upper line. As they performed their experiment on cursive word first they find connected component and base on that find the gaps between the connected component to segment text line. A similar approach applied to word segmentation is to find out gaps between connected components where each connected component provided with class id.

Sundaram et al. (2013) had proposed a methodology to segment a Tamil word based on attention feedback based method. In that work, they segmented online handwritten Tamil words into its integral symbols. In their segmentation method, the strategy comprised of two modules, in particular, dominant overlap criterion segmentation module and attention feedback segmentation module. In dominant overlap criterion segmentation module, based on boundary box overlap criterion, the word is first divided into stroke gathering. A stroke gathering may contain a valid symbol, some portion of the valid symbol and a merger of the valid symbol. In attention feedback segmentation module based on feature the over-segmented and under segmented stroke group is detected and after feedback from SVM classifier and feature from stroke group, the suspected group is converted to form a valid symbol.

Kumar et al. (2012) have implemented text -line segmentation of handwritten documents using clustering method based on thresholding approach. The technique comprises of three phases, in particular, drawing boundary box, a grouping of boundary box and text-line segmentation. Based on the grouping of boundary box text-line is segmented into words. Utilizing threshold value, the boundary box for each stroke is grouped into one group and based on threshold value text line is segmented into words.

Louloudis et al. (2009) had proposed text line and word segmentation of handwritten documents where segmentation of text lines is carried with the Hough transform. In preprocessing stage, the connected component is found with the help of average width and the average height of the character. Hough transforms applied on

connected component and in post processing phase merging technique applied on Hough transform result.

Kavallieratou et al. (2003) had proposed an incorporated framework for handwritten document image processing. In this paper, for segmentation, the page is vertically isolated into three sections and for each section, they applied horizontal histogram. The valley with minima less than the threshold is utilized to segment the text-line.

Papavassiliou et al. (2010) has proposed to extricate text-line and word from the manually written report. The segmentation algorithm is implemented based on the gap area within vertical zones and an ideal progression of text by applying the Viterbi algorithm. At that point, a text line separator drawing technique is applied and associated components are assigned to the text line. Word segmentation in view of crevice metric, which fulfilled the target function of a soft-margin linear SVM, that divides the successively associated components.

2.2 Work Related to Segmentation in Online Recognition System

Segmentation in online recognition systems applied to online handwritten script data. The online handwritten script data captured when the writer wrote on pen computing devices. This pen computing device having sensors to capture consecutive x and y values of the coordinate plane. As the present work focuses on online handwritten Gurmukhi data, it is necessary to discusss previous work of online data.

Bhattacharya et al. (2008) has presented an explanatory plan for online handwritten Bangla cursive word recognition. It concentrates on the technique for segmentation of line and words. This approach first identifies the headline which causes under-segmentation, but after removing headline the proper segmentation has been carried out. For segmentation, it then calculates a horizontal gap between adjacent words to get the proper segmented words as an output.

Ghosh (2013) has proposed a new approach for segmenting a word into characters. It first calculates the busy zone of the words and imagine the headline simply over the beginning stages of the assessed occupied zone. At that point, this exploits the pixels crossing the evaluated headline by checking their separation. This

approach was applied to the dataset of 5500 Bangla words and gives a segmentation accuracy of 94.9%.

Ibrayim et al. (2013) described the methodology for character segmentation in the online handwritten cursive Uyghur script. First, it removes delayed stroke from handwritten words and after that detects breakpoint from concavities ligatures by the shape and fleeting examination of the stroke direction, it also utilizes dynamic programming to discover the best segmentation point for each character.

Ghosh et al. (2009) has proposed the word segmentation methodology, where in the first stage, they divide the image into two zones. The uppermost zone is taken as 33% of the total image. Using a downside progression of stroke in the upper zone, the word is segmented into consolidation of strokes.They segment the word at that pixel, where the six successive strokes gratify certain angle.

Bhattacharya et al. (2002) has proposed a technique to segment the online handwritten Bangla words into characters. For segmentation, they investigate the joining arrangement of characters and modifiers of Bangla. Utilizing histogram, they discover the blank spaces in between the successive stroke.

CHAPTER 3

PROBLEM STATEMENT AND SIGNIFICANCE OF THE PROPOSED WORK

3.1 Problem Statement

The present work focuses on word segmentation of sentences in online handwritten Gurmukhi script data. For recognition of sentence, proper segmentation of words from the sentence is very important. This segmentation process is applied to input strokes of Gurmukhi script. An input stroke consists of *x* and *y* coordinates for each stroke of the sentence. It finds the vertical gap between consecutive strokes and based on threshold value the sentence is segmented into words.

The main problem here is to decide the ending of the words as the white spaces in between the words are not explicitly known. So, dynamically a threshold value is generated for each sentence, based on that threshold value and the vertical gap between the strokes, the sentence is segmented into words. The working flowchart is shown in underneath Figure 3.1 and next Chapter describes the segmentation procedure in detail.

3.2 Significance of the Proposed Work

As discussed earlier, the present work deals with segmentation problem. Many techniques and algorithms are also discussed in Chapter 2. The algorithms and techniques are mostly focused on offline data. Segmentation of online handwritten script data is not a difficult task, but very little work has been done on this field. Techniques are available to segment cursive words into character based on connected component. To segment strokes from different character has also been proposed. To segment strokes or to record strokewise data of particular script, the digital pen works well. Its movement like pen-up, pen-down and pen switching gives separated strokewise data. To segment words from sentences is not studied. No such work has been undertaken for Gurmukh script. Therefore, trial has been made to segment words from online handwritten Gurmukhi sentences. Though internal segmentation has been used in this work, both external and internal segmentation method has been discussed.

Also the advantage of external segmentation has been incorporated in the internal segmentation algorithm as applied to online handwritten Gurmukhi script. Segmentation is an important step for online handwriting recognition systems. It is expected that the current work on segmentation will be useful for further development of recognition systems of Gurmukhi script.

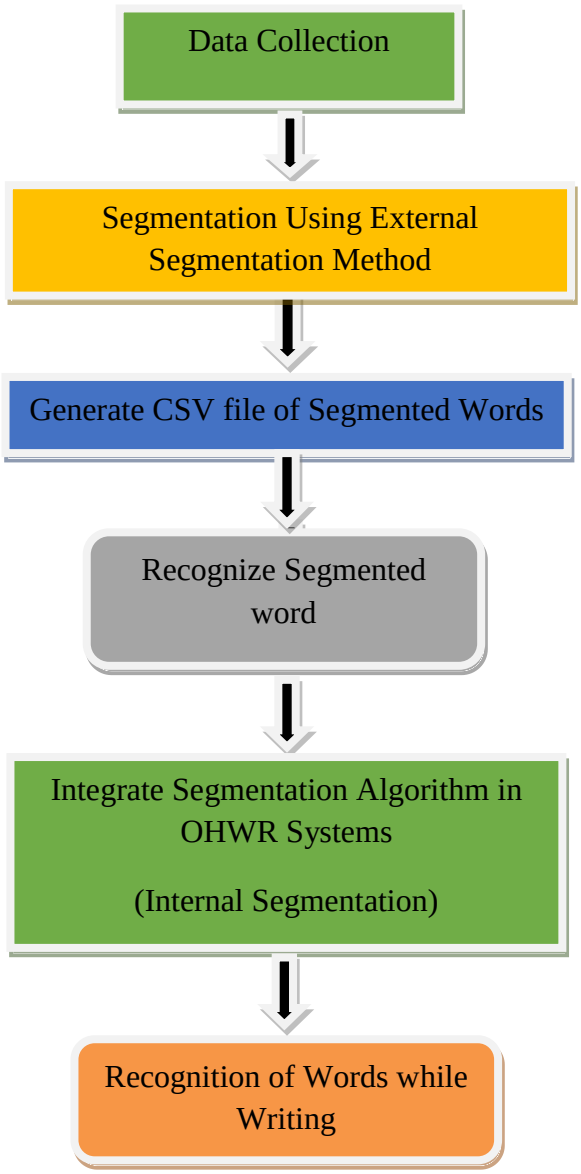

Figure 3.1: Working flowchart.

CHAPTER 4

SEGMENTATION OF WORDS FROM ONLINE HANDWRITTEN GURMUKHI SENTENCES

In the present work, we have proposed segmentation algorithms for online handwritten Gurmukhi data which has *x* and *y* coordinates of all the strokes present in a sentence. The proposed work falls in the category of internal segmentation. After segmentation, recognition of strokes is carried out using LibSVM with RBF kernel at the word level.

4.1 Segmentation Approach and Algorithm

4.1.1 Data Collection

In data collection phase, data are collected using a touch based device. Such devices have a sensor that uses a pen-tip position, pen-up and pen-down switching to capture *x* and *y* coordinates. It records the sequences of consecutive *x* and *y* points of the coordinate plane accurately.

For the present study, we used tablet PC to collect online handwritten data. Some other commonly used pen computing devices are Cross Tech3+ Multifunction Pen, Wacom CTL 471/K0-Cx Tablet, and Apple 12.9-inch iPad PR.

4.1.2 Segmentation Approach

As mentioned earlier, this study involves online handwritten Gurmukhi script stroke dataset. The dataset is prepared using a data collection system. Proposed algorithm is then applied for segmentation of this dataset. The value of white spaces between the words of collected dataset is not explicitly known. The white space values are calculated by observing the variation of *x* coordinates of each stroke. There are many ways to find variations on an individual basis, but the method (out of considered one) with the better result is considered for calculating the threshold values. So, in proposed method, the segmentation of words from the sentence is carried out with Thresholding approach and the technique comprises of two phases, in particular, drawing boundary box, and word segmentation using threshold value.

4.1.2.1 Drawing a Boundary Box

For drawing boundary box, the minimum and maximum values from *x* and *y* coordinates of each stroke are determined. This helps to calculate the vertical gaps between two strokes. Thus, the segmentation of words from a sentence is done on the basis of vertical gap between two consecutive strokes and the threshold value.

4.1.2.2 Word Segmentation

In this part, the mean value of vertical gaps of consecutive strokes is calculated. This mean value is considered as the threshold value for a segmentation algorithm. Now, the value of vertical gaps between two consecutive strokes is compared with this threshold value. The segmentation of the word is done at strokes where the vertical gap values are greater than the threshold value.

4.1.3 Proposed Algorithm

The proposed segmentation algorithm for segmenting words from sentences is as follows,

1) {Initialization}
    i. StrokeList ← List for 'n' number of strokes.
    ii. List of *x* and *y* coordinates for each stroke.
2) Find the minimum and maximum value for x and y coordinates of each stroke and store in the list XMin, XMax, YMin and YMax respectively.
    a) Set max to the first coordinate of list *x*.

For i ← 1 to *x*.size - 1

If *x*[i] > max Then

Set max = *x*[i]

Endif

Endfor

b) Set min to first coordinate of list *x*.

For i ← 1 to *x*.size - 1

If *x*[i] < min Then

Set min = *x*[i]

Endif

Endfor

3) Find horizontal gaps in between each stroke and store in list Gap.

a) For i ← 1 to XMin.size

Gap[i-1] = XMin[i] – XMax[i-1]

Endfor

b) Convert each negative value of list Gap to a and store in list positiveGap

For i ← 0 to Gap.size - 1

positiveGap[i] = Gap[i] * (-1)

Endfor

4) Find the average of the list positiveGap.

gapAvg ← $(\sum_{i}^{n-1} positiveGap[i])/(n - 1)$

5) Calculate threshold.

Threshold = gapAvg

6) For i ← 0 to Gap.size - 1

If (Gap[i] > threshold)

{

Perform segmentation

}

7) End

CHAPTER 5

EXPERIMENTAL RESULTS

Segmentation has been carried out on 10 different test sets; each test set consists of 20 sentences. A total of 200 sentences have thus been tested for segmentation. Based on individual test set results, the individual test set accuracy and overall segmentation accuracy have been calculated. The accuracy has been calculated and tabulated. To check accuracy, over segmentation, under segmentation and correct segmentation have been tabulated. Test set has been chosen with sentences consisting of one word, two words, three words, four words, five words, words with and without headline, sentences with uneven gaps between words, sentences with variations in size of words, sentences with variable and uniform length of words. Next section contain the experiments and corresponding results on these sentences.

5.1 Sentences Consisting of One Word

Experiment 1 has been performed on the sentences that contain only one word. 20 different sentences are tested for one word. A segmentation accuracy of 100.0% is achieved on these sentences. The sample output image of such a sentence is shown in Figure 5.1 and segmentation results with accuracy is shown in Table 5.1, respectively.

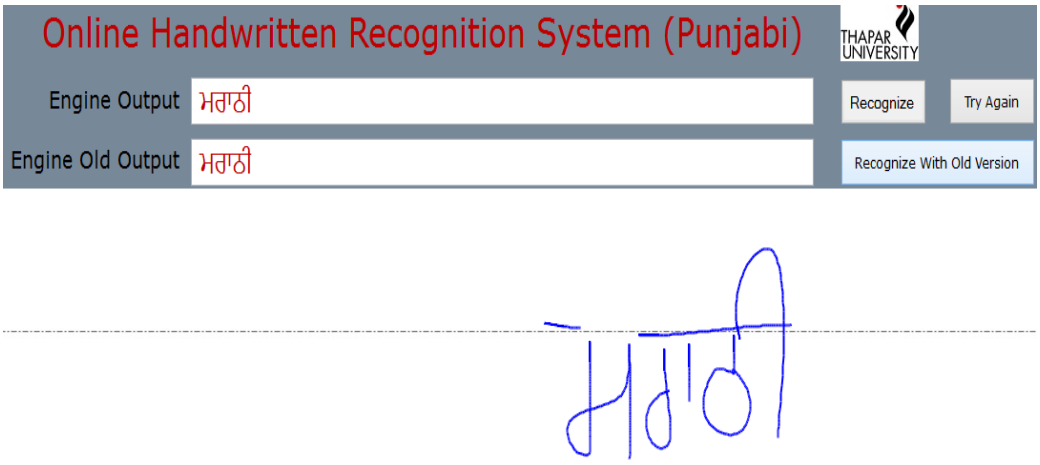

Figure 5.1: Sentence consisting of one word.

Table 5.1: Segmentation result of a sentences consisting of one word.

Test set	No. of Sentences	Under Segmentation	Over Segmentation	Correct Segmentation	Accuracy (in %)
1	20	0	0	20	100.0

5.2 Sentences Consisting of Two Words

Experiment 2 has been performed on the sentences that contain two words. Here, in this experiment, 20 different sentences consisting of two words have been tested and segmentation accuracy of 95.0% is achieved. The sample output image is shown in Figure 5.2 and a segmentation results with accuracy is shown in Table 5.2, respectively.

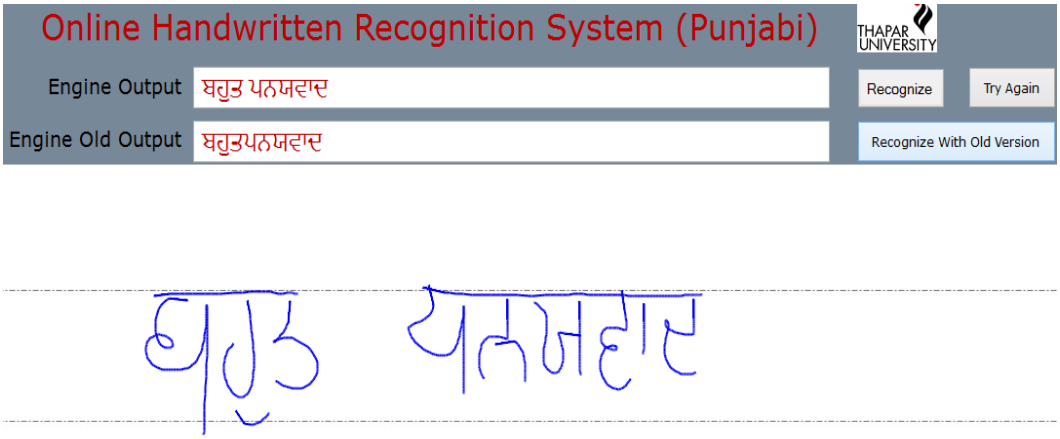

Figure 5.2: Sentence consisting of two words.

Table 5.2: Segmentation result of sentences consisting of two words.

Test set	No. of Sentences	Under Segmentation	Over Segmentation	Correct Segmentation	Accuracy (in %)
2	20	1	0	19	95.0

5.3 Sentences Consisting of Three Words

Experiment 3 has been performed on the sentences that contain three words. Here in this experiment, 20 different sentences have been tested and segmentation accuracy of 95.0% has been achieved. The sample output image is shown in Figure 5.3 and segmentation results with accuracy is shown in Table 5.3, respectively.

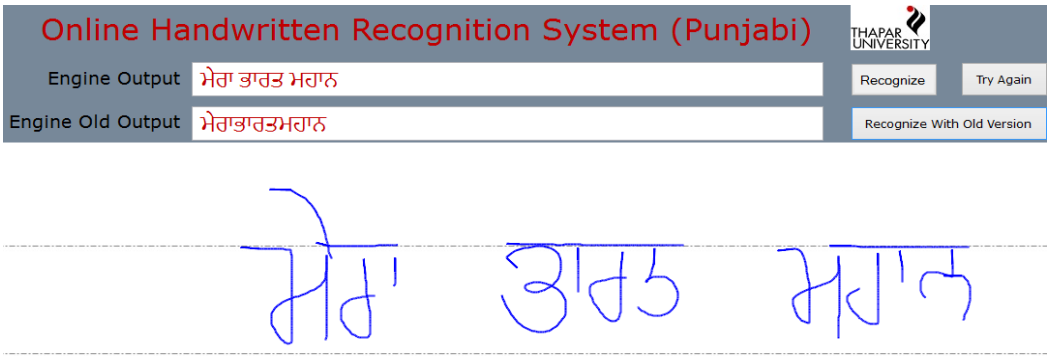

Figure 5.3: Sentence consisting of three words.

Table 5.3: Segmentation result of a sentences consisting of three words.

Test set	No. of Sentences	Under Segmentation	Over Segmentation	Correct Segmentation	Accuracy (in %)
3	20	1	0	19	95.0

5.4 Sentences Consisting of Four Words

Experiment 4 has been performed on the sentences that contain four words. Here in this experiment, 20 different sentences have been tested. A segmentation accuracy of 90.0% is achieved. The sample output image is shown in Figure 5.4 and segmentation result with accuracy is shown in Table 5.4, respectively.

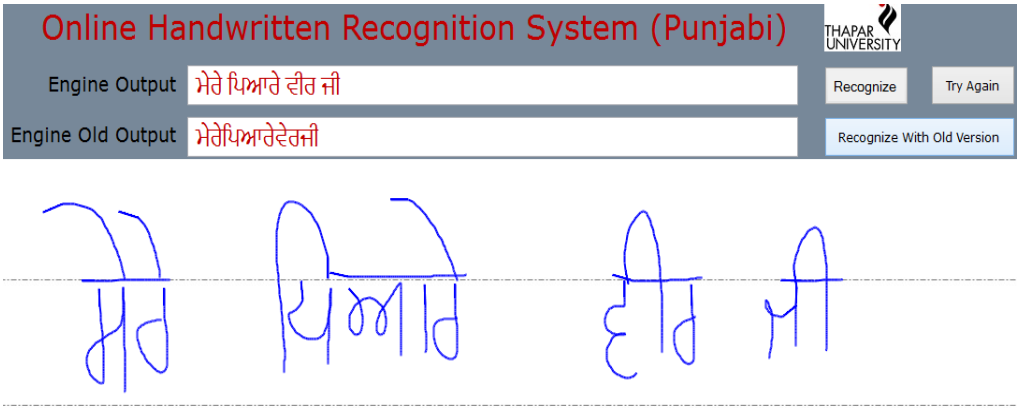

Figure 5.4: Sentence consisting of four words.

Table 5.4: Segmentation result of a sentences consisting four words.

Test set	No. of Sentences	Under Segmentation	Over Segmentation	Correct Segmentation	Accuracy (in %)
4	20	2	0	18	90.0

5.5 Sentences Consisting of Five Words

Experiment 5 has been performed on the sentences that contain five words. Here in this experiment, 20 different sentences have been tested and segmentation accuracy of 85.0% is achieved. The sample output image is shown in Figure 5.5 and segmentation results with accuracy is shown in Table 5.5, respectively.

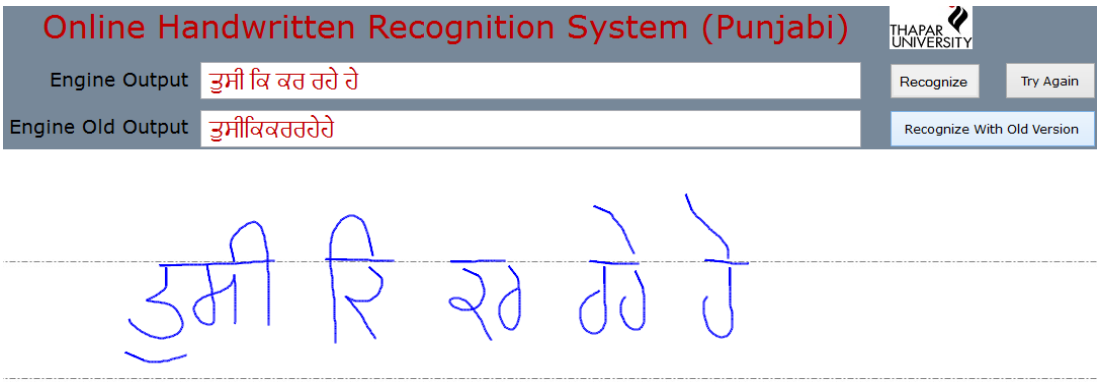

Figure 5.5: Sentence consisting of five words.

Table 5.5: Segmentation result of Sentences consisting of five words.

Test set	No. of Sentences	Under Segmentation	Over Segmentation	Correct Segmentation	Accuracy (in %)
5	20	3	0	17	85.0

5.6 Sentences Consisting of Words (2 or more) with/without Headline

Experiment 6 has been performed on the sentences that contain words with the headline and without a headline. Here, in this experiment headline roles a big part, as it takes a big part in drawing a boundary box of the stroke. Here in this experiment, 20 different sentences have been tested and segmentation accuracy of 100.0% is achieved. The sample output image is shown in Figure 5.6 and segmentation results with accuracy is shown in Table 5.6, respectively.

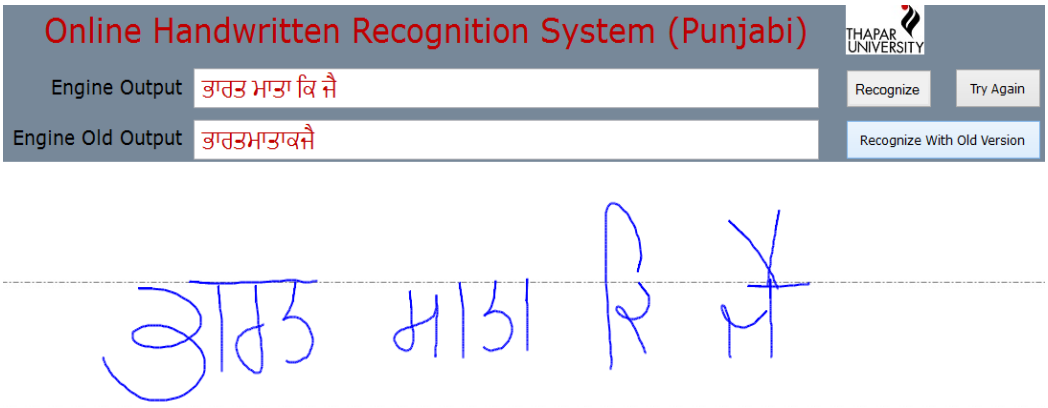

Figure 5.6: Sentence consisting of words (2 or more) with/without a headline.

Table 5.6: Segmentation result of Sentences consisting of words (2 or more) with/without a headline.

Test set	No. of Sentences	Under Segmentation	Over Segmentation	Correct Segmentation	Accuracy (in %)
6	20	0	0	20	100.0

5.7 Sentences having Uneven Gap between Words

Sometimes people wrote with irregular gaps between the words and it is important to check the algorithm for its suitability. The experiment 7 has therefore been performed on sentences that contain words with uneven gaps in between them. Here in this experiment, 20 different sentences have been tested and segmentation accuracy of 75.0% is achieved. The sample output image is shown in Figure 5.7 and segmentation results with accuracy is shown in Table 5.7, respectively.

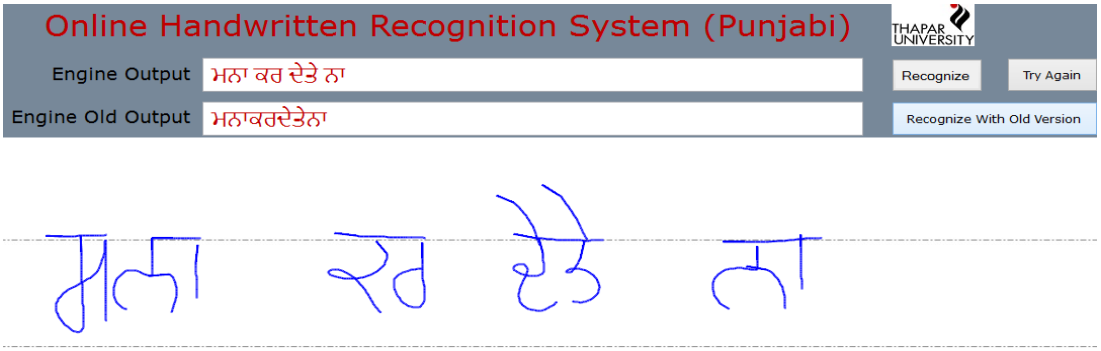

Figure 5.7: Sentence having uneven gap between words.

Table 5.7: Segmentation result of Sentences having uneven gap between words.

Test set	No. of Sentences	Under Segmentation	Over Segmentation	Correct Segmentation	Accuracy (in %)
7	20	5	0	15	75.0

5.8 Sentences with Variations in Size of Words

Experiment 8 has been performed on sentences that contain words with small and big size. Here in this experiment, 20 different sentences have been tested, and segmentation accuracy of 85.0% is achieved. The sample output image is shown in Figure 5.8 and a segmentation results with accuracy is shown in Table 5.8, respectively.

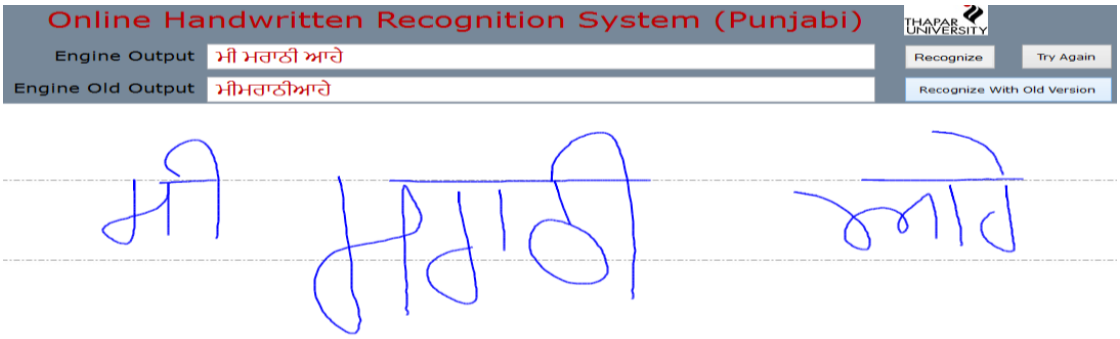

Figure 5.8: Sentence with large variation in the size of words.

Table 5.8: Segmentation result of sentences with large variation in the size of words.

Test set	No. of Sentences	Under Segmentation	Over Segmentation	Correct Segmentation	Accuracy (in %)
8	20	3	0	17	85.0

5.9 Sentences Consisting of Words with Varied Length

Experiment 9 has been performed on the sentences that contain words with variable length. 20 different sentences are tested for words with variable length. A segmentation accuracy of 85.0% is achieved on these sentences. The sample output image of such a sentence is shown in Figure 5.9 and a segmentation results with accuracy is shown in Table 5.9, respectively.

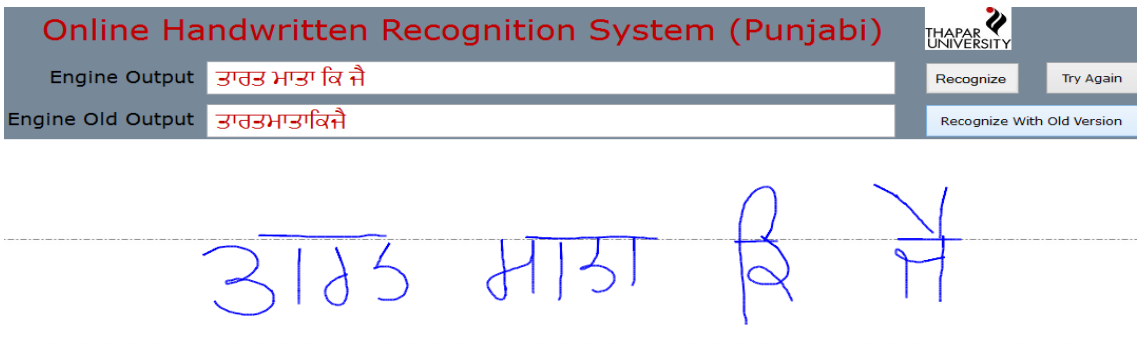

Figure 5.9: Sentence consisting of words with varied length.

Table 5.9: Segmentation result of sentences consisting of words of varied length.

Test set	No. of Sentence	Under Segmentation	Over Segmentation	Correct Segmentation	Accuracy (in %)
9	20	3	0	17	85.0

5.10 Sentences Consisting of Words with Same Length

Experiment 10 has been performed on sentences that contain words having a similar length. 20 sentences have been tested and segmentation accuracy of 100.0% is achieved. The sample output image is shown in Figure 5.10 and segmentation results with accuracy is shown in Table 5.10, respectively.

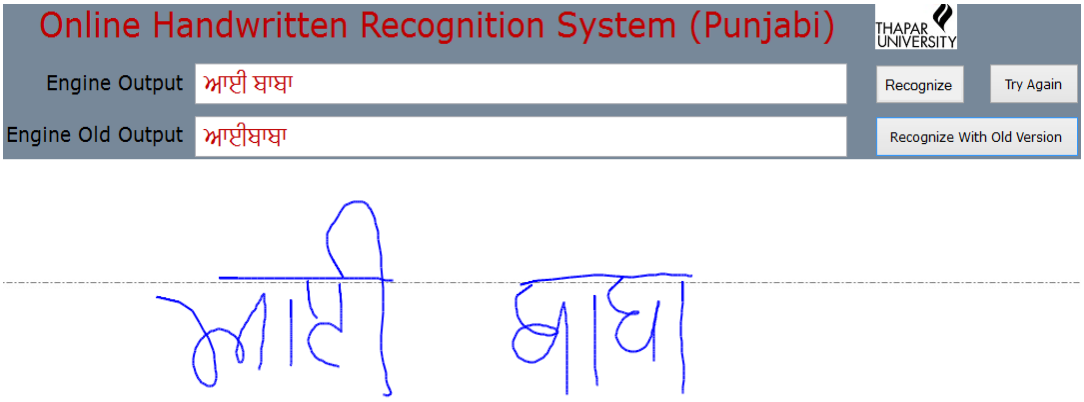

Figure 5.10: Sentence consisting of words with approximately same length.

Table 5.10: Segmentation result of sentences consisting of words with approximately same length.

Test set	No. of Sentences	Under Segmentation	Over Segmentation	Correct Segmentation	Accuracy (in %)
10	20	0	0	20	100.0

5.11 Overall Results

Table 5.11: Overall segmentation results.

Type of Sentences	No. of Sentences	Segmentation			Accuracy (in %)
		Under Segmentation	Over Segmentation	Correct Segmentation	
Sentences consisting of one word	20	0	0	20	100.0
Sentences consisting of two words	20	1	0	19	95.0
Sentences consisting of three words	20	1	0	19	95.0
Sentences consisting of four words	20	2	0	18	90.0
Sentences consisting of five words	20	3	0	17	85.0
Sentences consisting of words (2 or more) with/without headline	20	0	0	20	100.0
Sentences having uneven gap between words	20	5	0	15	75.0
Sentences with large variations in size of words	20	3	0	17	85.0
Sentences consisting of words with varied	20	3	0	17	85.0

length					
Sentences consisting of words with approximately same length	20	0	0	20	100.0
Total	200	18	0	182	91.0

5.12 Sentences where Improvements are Needed

In this section, we have presented a few cases where this segmentation approach does not perform well. This can be considered as a future work by other researchers in the field. Figures 5.11 – 5.15 contain such sample sentences.

5.12.1 Words in a Sentence having Small Gaps

The proposed algorithm is based on vertical gaps between words and threshold value. The underneath Figure 5.11 shows a sentence consisting of two words. This example doesn't have gap more than the required threshold value between words. Therefore, segmentation algorithm fails to segment these two words but reads these as one word.

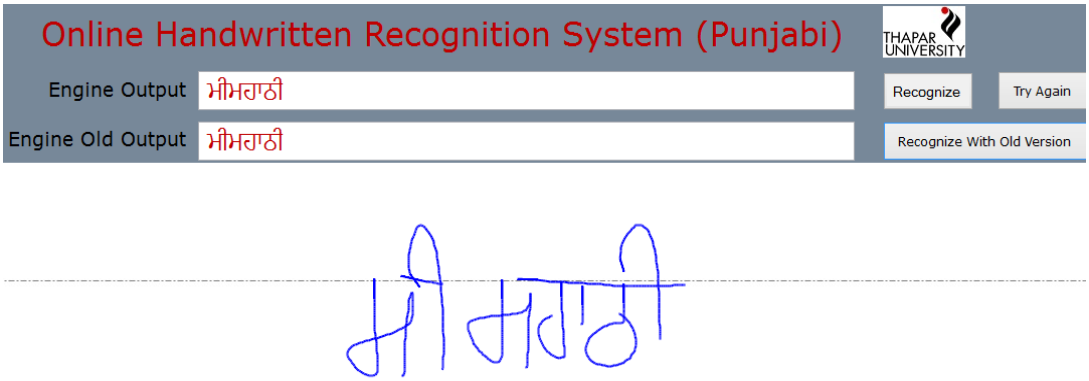

Figure 5.11: Sentence consists of words with small gap.

5.12.2 Sentence having Irregular Gaps between Words

The sentence shown in Figure 5.12 has uneven gaps between words. While determining the threshold, the algorithm may find some of the gaps in such sentences lower than the threshold value. In such a case problem of under segmentation would occur and segmentation algorithm is likely to fail in segmenting the words.

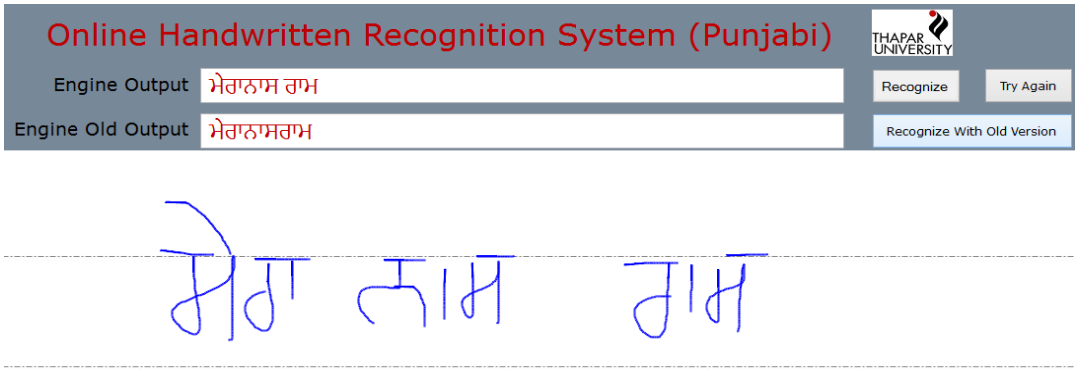

Figure 5.12: Sentence having irregular gaps between words.

5.12.3 Sentence Consists of Words in Slanting Style

As mentioned earlier, the proposed segmentation algorithm calculates vertical gaps between consecutive strokes. The cases shown in Figure 5.13 and Figure 5.14 have the sentences written in an inclined style. The vertical gap calculations are erroneous because of the incline of the sentence. The program, therefore, fails to segment words. The same sentence if written on *x*-axis or rotated to horizontal could be processed successfully.

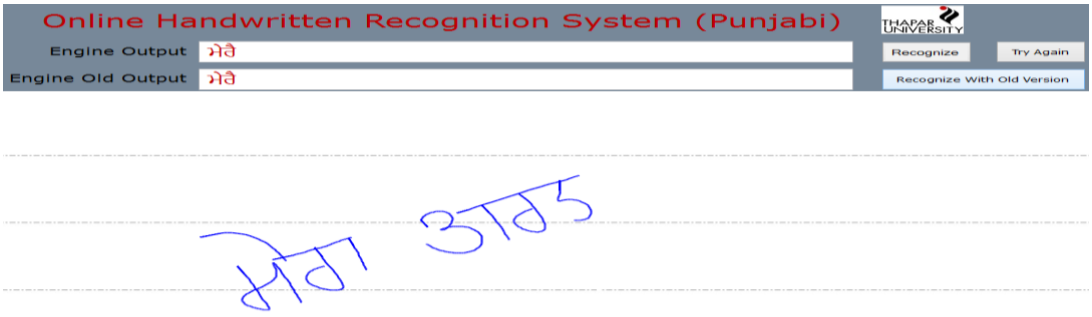

Figure 5.13: Sentence written in a slanted manner.

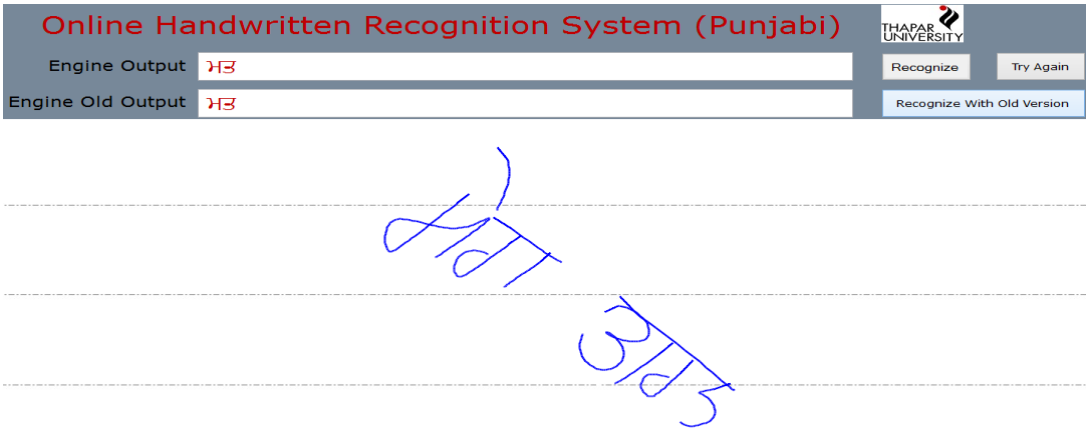

Figure 5.14: Sentence written in a slanted style.

5.12.4 Segmentation Performs on Words having Delayed Stroke

The proposed algorithm works with gaps between consecutive values along *x*-coordinate. The algorithm fails to segment words in a sentence, when the strokes of the letters in the sentence are not in continuous order. The delayed stroke is responsible for segmentation failure. In the example, the headline of the first word is written after writing all strokes. The Figure 5.15 shows one of such cases.

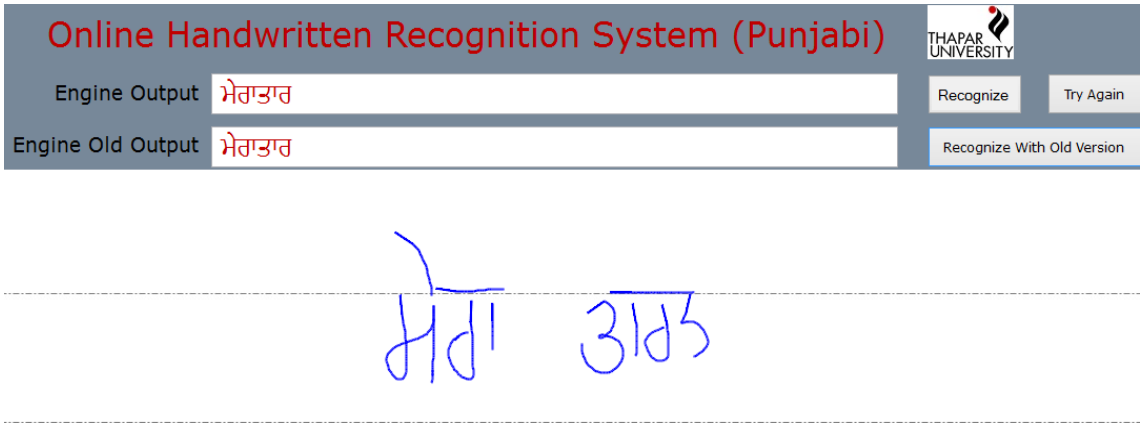

Figure 5.15: Sentence having delayed stroke.

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

6.1 Conclusion

The proposed methodology uses online handwritten Gurmukhi script data at stroke level when white spaces between the words are not explicitly known. This work is mainly focused on segmentation of words from different types of sentences as given in Table 5.11. This is evident from this table that the proposed algorithm works very well with sentences containing only one word or many words evenly spaced, giving an accuracy of 100.0%. The algorithm yields an overall segmentation accuracy of 91.0%.

6.2 Future Scope

This work can be extended in the future for segmentation of even more complex Gurmukhi sentences. This can also be extended in the direction of segmenting the sentences from a paragraph. One can also explore the possibility of extending this work for predicting next word while writing the words online. Future work can also be done to reduce limitations of this algorithm and to increase segmentation accuracy. Testing the algorithm on a larger dataset is an obvious extension of this work.

REFERENCES

[1] Aggarwal, Keerti, and R. K. Sharma. "DFT based feature extraction technique for recognition of online handwritten Gurmukhi strokes." *Inventive Computation Technologies (ICICT), International Conference on*. Vol. 3. IEEE, 2016.

[2] Aparna, K. H., et al. "Online handwriting recognition for Tamil." *Frontiers in Handwriting Recognition, 2004. IWFHR-9 2004. Ninth International Workshop on*. IEEE, 2004.

[3] Belhe, Swapnil, et al. "Hindi handwritten word recognition using HMM and symbol tree." *Proceeding of the workshop on Document Analysis and Recognition*. ACM, 2012.

[4] Bhattacharya, U., et al. "An analytic scheme for online handwritten Bangla cursive word recognition." *Proc. of the 11th ICFHR* (2008): 320-325.

[5] Bhattacharya, Nilanjana, Umapada Pal, and Kaushik Roy. "Individual Character Segmentation From Single Stroke Of Bangla Online Handwritten Text." *International Journal of Machine Intelligence ISSN* (2011): 0975-2927.

[6] Dürebrandt, Jesper. "Segmentation and Beautification of Handwriting using Mobile Devices." (2015).

[7] Ghosh, Rajib, Debnath Bhattacharyya, and Samir Kumar Bandyopadhyay. "Segmentation of Online Bangla Handwritten Word." *Advance Computing Conference, 2009. IACC 2009. IEEE International*. IEEE, 2009.

[8] Ghosh, Rajib. "Stroke segmentation of online handwritten word using the busy zone concept." *Soft Computing and Pattern Recognition (SoCPaR), 2013 International Conference of*. IEEE, 2013.

[9] Ibrayim, Mayire, Askar Hamdulla, and Dilmurat Tursun. "A Dynamic Programming Method for Segmentation of Online Cursive Uyghur Handwritten Words into Basic Recognizable Units." *JSW* 8.10 (2013): 2535-2540.

[10] Kavallieratou, E., Dromazou, N., Fakotakis, N. and Kokkinakis, G., 2003. An integrated system for handwritten document image processing. *International Journal of Pattern Recognition and Artificial Intelligence*, *17*(04), pp.617-636.

[11] Kumar, M. Ravi, Nayana N. Shetty, and B. P. Pragathi. "Text line segmentation of handwritten documents using clustering method based on thresholding approach." *International Journal of Computer Applications (0975–8878) on National Conference on Advanced Computing and Communications-NCACC*. 2012.

[12] Louloudis, G., Gatos, B., Pratikakis, I. and Halatsis, C., 2009. Text line and word segmentation of handwritten documents. *Pattern Recognition*, *42*(12), pp.3169-3183.

[13] "Machine Learning with scikit-learn." *Scikit-learn : Support Vector Machines (SVM) - 2017*. N.p., n.d. Web. 19 June 2017.

[14] Mondal, T., et al. "Database generation and recognition of online handwritten Bangla characters." *Proceedings of the international workshop on multilingual OCR*. ACM, 2009.

[15] Palakollu, Saiprakash, Renu Dhir, and Rajneesh Rani. "A New Technique for Line Segmentation of Handwritten Hindi Text." *Special Issue of International Journal of Computer Applications* (2011): 0975-8887.

[16] Papavassiliou, Vassilis, Themos Stafylakis, Vassilis Katsouros, and George Carayannis. "Handwritten document image segmentation into text lines and words." *Pattern Recognition* 43, no. 1 (2010): 369-377.

[17] Pechwitz, Mario, and V. Margner. "Baseline estimation for Arabic handwritten words." *Frontiers in Handwriting Recognition, 2002. Proceedings. Eighth International Workshop on*. IEEE, 2002.

[18] Samanta, Oendrila, Ujjwal Bhattacharya, and Swapan K. Parui. "Smoothing of HMM parameters for efficient recognition of online handwriting." *Pattern Recognition* 47.11 (2014): 3614-3629.

[19] Sharma, Anuj, Rajesh Kumar, and R. K. Sharma. "Online handwritten Gurmukhi character recognition using elastic matching." *Image and Signal Processing, 2008. CISP'08. Congress on*. Vol. 2. IEEE, 2008.

[20] Sharma, Anuj, R. K. Sharma, and R. Kumar. "Online handwritten gurmukhi strokes preprocessing." *Int. J. Mach. Graph. Vis* 18 (2009): 105-120.

[21] Sundaram, Suresh, and A. G. Ramakrishnan. "Attention-feedback based robust segmentation of online handwritten isolated Tamil words." *ACM Transactions on Asian Language Information Processing (TALIP)* 12.1 (2013): 4.

[22] Swethalakshmi, Hariharan, et al. "Online handwritten character recognition of Devanagari and Telugu Characters using support vector machines." *Tenth International workshop on Frontiers in handwriting recognition*. Suvisoft, 2006.

[23] Tutorialspoint.com. "Computer Graphics Tutorial." *Www.tutorialspoint.com*. N.p., n.d. Web. 19 June 2017.

[24] Verma, Karun, and Rajendra Kumar Sharma. "Comparison of HMM-and SVM-based stroke classifiers for Gurmukhi script." *Neural Computing and Applications* (2016): 1-13.

[25] Wang, Da-Han, and Cheng-Lin Liu. "Dynamic text line segmentation for real-time recognition of Chinese handwritten sentences." *Document Analysis and Recognition (ICDAR), 2011 International Conference on*. IEEE 2011.

[26] Yin, Fei, and Cheng-Lin Liu. "Handwritten Chinese text line segmentation by clustering with distance metric learning." *Pattern Recognition* 42.12 (2009): 3146-3157.

[27] Zheng, Liying, Abbas H. Hassin, and Xianglong Tang. "A new algorithm for machine printed Arabic character segmentation." *Pattern Recognition Letters* 25.15 (2004): 1723-1729.

APPENDIX A

PUBLICATION

Devesh Dahake, R.K. Sharma and Harjeet Singh, "On Segmentation of Words from Online Handwritten Gurmukhi Sentences,"*2nd International Conference on Man and Machine Interfacing, Bhubneswar,* 2017 (**communicated**).

APPENDIX B

VIDEO PRESENTATION LINK

https://youtu.be/hmLQ-2xFOAA

APPENDIX C

PLAGIARISM REPORT

ORIGINALITY REPORT

%8	%3	%7	%1
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Keerti Aggarwal, R.K. Sharma. "DFT based feature extraction technique for recognition of online handwritten Gurmukhi strokes", 2016 International Conference on Inventive Computation Technologies (ICICT), 2016 Publication	%1
2	Belhe, Swapnil, Chetan Paulzagade, Akash Deshmukh, Saumya Jetley, and Kapil Mehrotra. "Hindi handwritten word recognition using HMM and symbol tree", Proceeding of the workshop on Document Analysis and Recognition - DAR 12 DAR 12, 2012. Publication	%1
3	Mondal, T., U. Bhattacharya, S. K. Parui, K. Das, and V. Roy. "Database generation and recognition of online handwritten Bangla characters", Proceedings of the International Workshop on Multilingual OCR - MOCR 09 MOCR 09, 2009. Publication	%1