International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8, Issue-1, May 2019

A Bilingual (Gurmukhi-Roman) Online Handwriting Identification and Recognition System

Gurpreet Singh, Manoj Kumar Sachan

Abstract: ***Advances in recent technologies and the power of Artificial Intelligence (AI) always motivate the researchers to work for Human-Computer Interaction (HCI). So that, HCI behaves similar to HHI (Human-Human Interaction). Various applications which come under HCI are automatic emotion detection from facial images and videos, sentiment analysis from textual data or abbreviation slangs, handwriting recognition etc. Automatic recognition of handwriting is one of the most complex task for computer systems. The factors like the existence of variety of languages with their unique scripts, difference among the writing styles of different writers or even same writer over time, delayed strokes, noise, requirement of huge amount of training etc. make this task most difficult. In this paper, the problem of Online Handwriting Recognition (OHR) is addressed for the text containing bilingual data. The input data is written by using English and Punjabi languages, so both Roman and Gurmukhi scripts are used for the purpose of writing. During feature extraction process, the stress is given to the local features by considering direction of strokes, position or order of strokes, slope and area covered by one complete stroke. Global features include the information about Headline stroke, straight lines and dots. Statistical features consider Zone identification and Direction code histogram. The final recognition is done by using Multi Layered-Perceptron (MLP) neural network as a classifier in the presence of two simultaneous recognition engines for Roman and Gurmukhi scripts respectively. Experimental observations show good results for the bilingual text.***

Keywords: ***Bilingual Data; HCI (Human-Computer Interaction); HHI (Human-Human Interaction); Online Handwriting Recognition (OHR); Multi-Layered Perceptron (MLP) Neural Network***

I INTRODUCTION

Automation of different tasks performed in day to day life with the help of computing techniques reflects number of challenges. Human computer Interaction (HCI) is one of the challenges of this automation process. The main part of HCI is to feed the computing information in the automated system, such that, the system can accept the input in the form as it is given in the case of Human-Human Interaction (HHI) process. The field of Artificial Intelligence (AI) is the backbone of computing technologies. The different steps in AI applications such as input, training, learning, identification and recognition helps to build automated system by capturing human intelligence. Handwriting recognition by computer systems is one of the applications of AI. The target of these systems is to convert the handwritten data into digital form or computer understandable form [1,2]. These handwriting recognition systems are mainly used to provide fast input to some processes by replacing the use of hardware devices like keyboard.

There are two different modes of giving handwritten input to computer systems. One by giving scanned handwritten documents as images and second, with the help of digitizers & stylus (Digital pen) devices [1]. The system which accepts the input of handwritten scanned images is considered under the category of Offline handwriting recognition systems. When the handwriting input is given with the help of digitizers & stylus combination then the systems are considered as Online handwriting recognition systems [3,4]. For writing something in his/her own handwriting, the writers have to use some script representing the desired language of communication. Across the world, more than 7000 languages are used for communication purposes [5]. In India, we have 22 official languages to speak. According to Census, total number of mother tongues spoken in India are 1635 [6]. Every language has its own rules and regulations or script for writing. Means different writing styles are used according to the selected language as per its character set and rules.The complex nature of these scripts makes the handwriting recognition task more challenging. Also there are number of other challenges for the field of Online handwriting recognition like variations in the handwriting styles of writers, quality of digitizer, familiarity of writers with digitizer environment etc[20]. Digitizers based computing technologycame into the existence in 1968 [1]. The advent of this technology gave power to the users of computers to input data in computer systems with a rapid pace using handwriting samples. After that, number of machine vision or computer vision researchers worked on the idea to convert handwritten text to computer understandable form. A lot of work has been done for English language as this language is widely used for the purpose of communication. Almost 100% recognition rate is achieved by the researchers for converting handwritten English text into digital data [7–9]. The script used to write English text is Roman script. Along with English language, now researchers are working on the Online handwriting recognition systems for local languages because of their popularity among large group of people. Researchers are getting good results from Online handwriting recognition system build for languages used in Japan, China, India etc.[10–12]. Recently some work is also published for Online handwriting recognition system based on bilingual scripts [13].

Revised Manuscript Received on 30 May 2019.
***** Correspondence Author
Gurpreet Singh, CSE, CU, Gharuan
Manoj Kumar, Sachan CSE, SLIET Longowal

The authors worked on different language recognition engines for all the languages and scripts they considered. It means before entering the input text written using a single script they have to select the recognition engine meant for that script. So, still the automatic Online handwriting recognition of bilingual text is an open area. Very few researchers worked on bilingual text [13]. The combination of Gurmukhi and Roman scripts under bilingual identification and recognition is not covered till date.

In this paper, a technique is presented for Online handwriting recognition of bilingual scripts (Gurmukhi and Roman). The input text in large amount is written using Gurmukhi script. Some part of the input text is written using Roman script, when there is a need to clarify the meaning of some difficult Punjabi words by giving their English meaning in parenthesis. Punjabi is one of the 22 official languages of India. Worldwide this language is famous because its native speakers are 1.44% of the total population of the world i.e. about 102 million people. Punjabi is in the list of top 10 most spoken languages across the world [14]. Gurmukhi script is used to write Punjabi language. On the other hand English language is written with the help of Roman script. Here, a system is developed which accepts the text written by using two scripts (Gurmukhi and Roman). The main challenges of the system are: to input required data using digitizer, to identify the script of the word after its segmentation, development of recognition engines for Gurmukhi as well as for Roman scripts.

The next part of the paper will highlight the work done related to handwriting recognition systems for different languages and their scripts, After that, the system architecture for Online bilingual Gurmukhi-Roman handwriting recognition will be presented. Then a section explains the Gurmukhi words dataset prepared for training and testing of the developed system. After the dataset, every phase of language identification and recognition is presented in different section according to the sequential order of their execution. In the end, the result and discussion section will prove the strength of the system developed by explaining different tables and graphs. In the last section of the paper, conclusion is drawn by highlighting the key points and by proposing the future challenges about the field under consideration.

II RELATED WORK

This section of the paper will highlight the work done by different authors in the field of Script recognition. The work presented in this section will cover both Online or Offline handwriting recognition processes along with bilingual nature of handwritten samples given as input to the system. The idea about different techniques used at each stage of automatic recognition of handwriting systems is also presented in this section. Following is the brief introduction of some papers considered for literature review:

Keysers et al. [7] worked on Google's OHR system for the recognition of 22 different scripts and 97 languages. Their focus was on the reuse of components in different scripts and to speed up the recognition process. They considered time based and position based information of strokes to deal with the complexity factors such as: overlapping of strokes and the presence of delayed strokes. They used the concept of deep learning to handle the issue of automatic handwriting recognition.

Indhu et al. [13] developed an android application for bilingual (Malayalam & Telugu) OHR system based on character level recognition. They formed a language independent engine by working on structural and directional information of strokes. The features under consideration were normalized coordinate points, directions, curvature, loops, curliness and stroke length. They handled the problem of delayed strokes by the use of probability based stroke rules. SFAM (Simplified Fuzzy ARTMAP) artificial neural network was used for the purpose of classification. They achieved the recognition accuracy of 95.78% and 97.12% for Malayalam and Telugu languages respectively.

Aggarwal et al. [8] presented an approach for English language OHR system by using depth sensors. They used Kinect and Leap motion controllers as depth sensors to capture hand movements of writers. They worked on finger print tracking approach and collected data from 20 different users. Authors claimed the recognition accuracy of 97.59% for English language characters. For the purpose of segmentation, the concept of K-means clustering was used.

Samanta et al. [15] presented a script independent approach for OHR system. They checked the performance of their system for UNIPEN KROW dataset containing 211 English language words, ADBA dataset for Arabic language containing 200 words and Bangla online handwritten words dataset containing 110 classes. They focused on three segmentation schemes, first was based on Discrete Curve evolution algorithm. This algorithm decomposed an object into smaller meaningful visual parts. The second segmentation scheme was Minseg, the scheme was used to set local minima points. The last segmentation strategy was ANGseg. It worked on pen trajectory of individual strokes. They achieved high recognition rates with the use of local minima segmentation scheme as compared to other two schemes. Sadrnezhad et al. [9] worked on OHR system for Farsi language. They considered only the isolated characters and numbers for their study. The main focus was on directional components, which firstly assumed to have eight directional features. On the basis of directional features, hand movements were captured by the authors for further recognition process. They implemented HMM (Hidden Markov Model) with help of Baum-Welch algorithm for identification of handwritten samples. For numeric data input, they got 99.22% recognition rate and for Farsi characters 95.91% accuracy was achieved. Prasad et al. [16] compared the performance of one dimensional LDA(Linear Discriminant Analysis) and 2D-LDA for online handwritten character recognition for Kannada language. They processed the sample strokes data by forming Column vector in case of 1D-LDA but in case of 2D-LDA sample data processed directly. Tablet PC was used to collect data samples. Total 3750 training samples and 1550 test samples were considered for the purpose of evaluation. Authors used nearest neighbor classifier in their work along with second derivative feature vector. The recognition accuracies for 1D-LDA and 2D-LDA were observed as 87.4% and 87% respectively. Dutta et al. [17] worked on an android application for OHR system. The Bangla language was under consideration. Based on the look-up table, authors recognized the characters from stroke information.

To identify and recognize strokes they used nearest neighbor classifier. Learning Vector Quantization (LVQ) was used to store the information of strokes which were not been able to identify simply with the help of look-up table. The system proposed by them was writer dependent system. The maximum accuracy achieved by them for Bangla character classification was 90.20%. They gave the flexibility to the system by providing suitable entries to the look-up table. They claimed that, their system can be used for other languages too.

Teja et al. [18] worked on the segmentation aspect in OHR for "Malayalam" language. They focused on the concept of Ballistic strokes. The main purpose was to deal with the noise, which was generated due to the writing styles of the writers or the type of digitizer used to collect handwritten strokes. Firstly, they removed the noise from the input data. After that boundaries were detected by using curvature maxima approach. Cubic spline method was used to model handwritten strokes. Genius G-Note 7000 digital pad provided the stroke data in their experimentation and They achieved 94.55% recognition rate by using SVM (Support Vector Machine) classifier.

Chowdhury et al. [19] presented an approach using Levenshtein distance metric for OHR for Indian scripts, which included Devanagari, Bangla, Telugu and Tamil. They considered shape information and position information of strokes for identification. The work done by them was on isolated character recognition. They compared their work with MLP and HMM classifiers and achieved recognition rate as 98.43% for Bangla numerals, 86.24% for Bangla characters, 83.95% for Devanagari characters, 87.10% and 85% for Telugu and Tamil characters respectively.

Indu et al. [21] presented a simplified fuzzy system based on Adaptive Resonance Theory (ART) for OHR for Malayalam Language. They concentrated on directional and structural aspects of input data. The output formed by their system was font independent and based on Unicode model. They considered the dataset containing 8135 samples of Malayalam Language's words, with the contribution of 29 writers. During feature extraction process coordinate values, start & end quadrants, horizontal & vertical point density, loops, cusps and aspect ratio were considered. The average accuracy of recognition process was observed as 98.26%. Further for the purpose of post processing strokes concatenations and linguistic rules were used.

Zhao et al. [10] worked on a system for OHR for "Uyghur" (Chinese language). They performed complete word level recognition rather than only for isolated characters. "Uyghur" consists 32 characters in its character set and the writing direction for this language is usually form right to left. The complexity factor associated with the language is variations in shape of same character as per its appearance at different positions in a word. The problem of delayed strokes is also there in this language. Authors used HMM technique to deal with the problem of automatic handwriting recognition. During feature extraction process the main focus was on features like local angle, super segment, loops identification and location of strokes. They used the test set containing 1058 words.

Kunwar et al. [22] proposed a system for OHR for Tamil language. They considered only isolated characters of Tamil. The main objective was to enhance the speed of encoding and decoding process during recognition. Authors worked on the concept of natural objects rather than idealized one. So, they considered fractal coding for the purpose of recognition. The dataset under consideration was 1WFHR 2006 database. They worked on small segments and achieved 12% improvement in recognition rate as compared to previous existing work.

Hosny et al. [11] worked on OHR system for Arabic script. Their main focus was to deal with the delayed strokes like stocks/dots which mainly appeared above or below other characters. They handled the main difficulty level of the script as, cursive nature and unconstrained environment. For the recognition purpose, authors used advanced HMM as a classifier. They worked on International test set of Arabic language (ADAB dataset). For the purpose of extraction of meaningful features, chain code method was used to keep the record of directions in stroke, curvature, base line and vertical position vector. Normalization of selected features was done with the help of Z-score transformation. During testing phase, their observed accuracy was 96%.

Kaur et al. [23] presented a comparison of three different classifiers for online handwritten Gurmukhi script character recognition system. Authors considered KNN, MLP neural network and SVM classifiers for their study. They worked on dataset of Gurmukhi handwritten characters containing 2019 samples. 32 different stroke classes were considered for comparison purpose. The features under consideration were position of strokes along with temporal information and Fast Fourier Transformation (FFT) feature vector. To enhance the quality of input data various pre-processing operations like normalization & centering of strokes, Interpolation & smoothing, slant correction & sampling were performed by the authors. Their comparative study claimed the better performance of MLP classifier over KNN and SVM.

Baral et al. [24] proposed a novel method to detect the region between baseline and headline for Online Bangla handwritten words. This desired region is also known as Core region. The features used were value of maxima and first derivative peak value. MLP classifier was considered for the purpose of identification of strokes. Instead of complete strokes, authors worked on sub stroke level. The dataset used for observations contained 9424 Bangla words. The dataset was prepared with the help of 16 native writers. The handwriting samples of 10 writers were used to test the system. The accuracy for the identification and recognition of headline and baseline for complete words was 91.60% and 90.20% respectively.

Mondal et al. [25] worked on Online Handwriting dataset of Indian languages. They considered four existing datasets of Bangla, Devanagari, Tamil and Telugu languages. Two different approaches such as Float Point and directional code histogram were used for the purpose of feature extraction. The classifiers considered were KNN, MLP and HMM. Authors concluded that the performance of chain code feature was observed to be better as compared to Float Point method. The recognition accuracy level of KNN was observed to be on higher side as compared to other two classifiers. Baiju et al. [26] implemented an online handwritten character recognition system for Malayalam language by considering three different classifiers MLP, KNN and SVM.

They used Hi-Tech e-write mate digitizer for collecting dataset for testing and training purpose. Total 20 different writers contributed in dataset generation, which was the collection of 44 different characters of Malayalam language. Aspect ratio, intersections, accurate dominant points were the features under their observations. According to the selected feature vectors, they concluded the better performance of SVM classifier over MLP and KNN by claiming the recognition accuracy of 95.12%, 93.17% and 90.30% respectively for SVM, MLP and KNN classifiers.

Aggarwal et al. [27] proposed a composite technique by combining the power of two features together to produce an offline handwriting recognition system for Gurmukhi script characters. The selected features were Gradient and Curvature of an image. They formed the composite feature by two different ways. First, they considered the simple concatenation of the above said features. Then they formed the composite vector by taking the cross-product of Gradient and Curvature. Authors observed the better performance in case of cross-product feature vector.

Sharma et al. [28] proposed a post processing approach to increase the recognition rate of optical character recognition system, which they developed for printed and handwritten offline character set of Punjabi language. To improve the performance of their earlier system, they found some classes of similarity. Then, they used shape encoded method along with AVL search tree approach to identify the meaningful words of Punjabi, which were not addressed fully previously. Authors created the dictionary of these words and achieved 4.65% improvement in case of machine printed words & names and 7.45% improvement over handwritten words.

Sharma et al. [29] presented a method for the conversion of handwritten characters of Gurmukhi script to digital form. The method used by them was Elastic Matching. This method used a distance function to compare an unknown stroke with the already existing known stroke present in database. XML technology was used by them to create the database of strokes by providing unique ID's to each stroke. Authors claimed 90.08% recognition accuracy for 41 different characters of Gurmukhi script.

Gaur et al. [30] presented a new method for the generation of dataset for offline handwriting recognition system. They considered the character set of Indian languages (Hindi, Bengali and Telugu). The main purpose was to develop a dataset for handwriting recognition applications. The lack of such kind of datasets motivated the authors to work on this aspect of research. During feature selection process, the considered features were local gradient and histogram. HMM was used in classification phase to check the sequential dependencies of various classes.

kunwar et al. [31] presented a method for handwritten character recognition for Kannada language. Along with the character set of Kannada, authors considered all other special symbols like $,&,# etc. They proposed and implemented a writer independent system by taking dataset records from 69 different writers, belongs to four different locations. The technique used by them was statistical dynamic time wrapping. They claimed 46 times increased speed of recognition with their technique along with the accuracy of 88% in reference to writer independent and open vocabulary case.

The concluding remarks of above discussed literature reflects the fact that the number of researchers are currently working on Online handwriting recognition systems. They consider the different scripts for their work or even some are working on more than one scripts (Bilingual systems) at the same time. The different languages which were considered by the authors include Chinese, Malayalam, Arabic, Tamil, Bangla, Telugu, Punjabi, English, Kannada, Farsi etc. The different classifiers and techniques used by the authors were HMM, SVM, MLP, LDA, KNN, AVL Trees, Elastic matching, Fuzzy systems, ANN, Statistical dynamics time wrapping etc. The work done by the author either falls in the category of character recognition or complete word recognition. It is also observed from the literature that, the bilingual online handwriting identification and recognition part for the combination of Gurmukhi and Roman scripts, till date is not addressed by researchers.

III SYSTEM ARCHITECTURE

In this paper, the problem of Online handwriting recognition for bilingual scripts has been addressed. The scripts under consideration are Gurmukhi and Roman. Figure1. shows the architecture of the developed system. The basic idea behind the overall process is to take handwritten strokes of bilingual (Punjabi and English) text as input and convert this text into digital form. During the input process, bulk of the data is written in Punjabi language using Gurmukhi script. But for the purpose of clarification, meanings of some difficult Punjabi words are also written within "(" and ")" by using English language. To give handwriting as input to the system Tablet PC is used with active stylus. The implementation environment of C#.Net has been used for developing the bilingual system and digital ink is captured by using two different Ink classes i.e. "InkOverlay" and "InkCollector" [32].

After providing input to the system, next job is to pre-process the input data to avoid noise and other flickering effects. For the purpose of pre-processing, the steps like removal of repetitive points, consideration & identification of missing points and normalization of data are used. The pre-processed strokes further provide input to the segmentation phase. The process of segmentation is performed on the basis of individual strokes, stroke representing headline and identification of strokes in different writing zones. The number of individual strokes and the sequence of these strokes under consideration helps in finding the character formed by the strokes combination. The stroke named as "Headline" provides the information about the formation of a complete words of Gurmukhi script. Also the stroke "Headline" is used to distinguish between character which are similar in appearance with each other as per their shape is concerned. Segmentation of handwriting sample into three different zones as Lower zone, Middle zone and Upper zone helps to distinguish between consonants, vowels and half characters.The position of "Headline" stroke helps to identify upper zone and Middle zone region. Middle zone is also known to be as a core region of handwritten samples. The other factors like position and height of strokes also helps in the identification of lower zone.

This lower zone mostly containing half characters and some information about vowels. The next part after segmentation process is to find feature of interest from the segmented data. These features will help in the identification of the structural information about strokesIn this paper, the features under consideration are local features (low level), global features (high level), statistical features and structural features.Low level features focuses on the direction of pen movement, position of stroke in stroke sequence (delayed strokes), loops and area covered by an individual stroke. High level features consists of Headline, Straight line and Dots. Zoning and Directional code histogram has been considered under Statistical features and shape of individual stroke being considered as Structural features. Strokes identification dataset contains the information about the shape of the stroke and the unique ID assigned to each stroke. The classification process is implemented by using Multi-Layered Perceptron (MLP) neural network. In this paper three layered architecture of MLP is presented for classification step. MLP helps to generate the probability distribution of different stroke classes under consideration. On the basis of these probabilities, the shape of the stroke under consideration has been identified and compared. This information further helps in the identification of the script of the complete word under consideration. In case, the script is identified as Roman script, then these handwritten strokes are passed to Roman script recognizer for further processing. In this bilingual handwriting recognition system, an inbuilt Roman script recognizer is used. This Roman script recognizer is implemented within InkOverlay class of C#.Net [32]. This class has its automatic recognizer for English language.

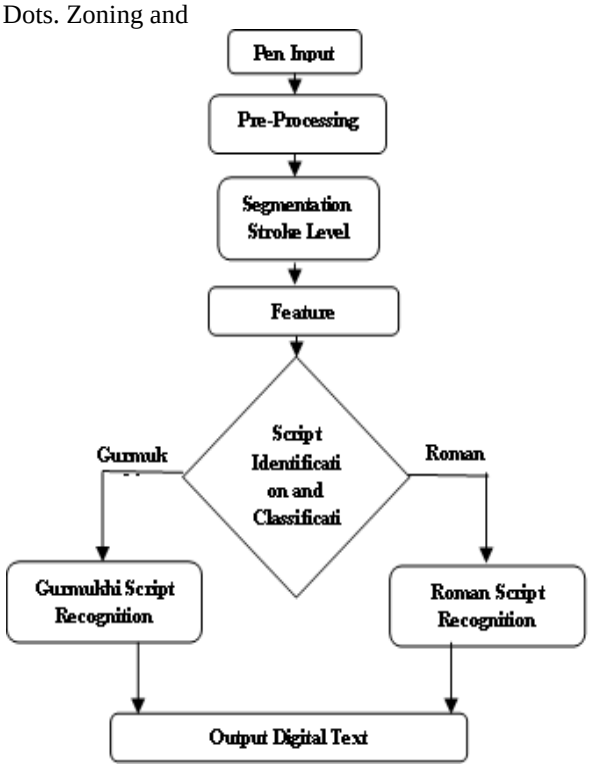

Figure1: Architecture for bilingual (Punjabi-English) Online handwriting recognition system

Table1. Mapping of strokes with Gurmukhi character symbols

C_Id	Character	Stroke Mapping	C_Id	Character	Stroke Mapping
1	T	(1)	30	w	(37)
2	n	(2), (3,4)	31	:	(4,5,38), (4,5,39)
3	J	(5,6,7)	32	o	(5,14)
4	;	(4,5,8), (5,9)	33	b	(5,6,41), (5,26,40,41)
5	j	(5,10)	34	t	(5,20)
6	e	(11), (5,11)	35	V	(4,42), (42,44)
7	y	(5,13)	36	P	(4,5,8,55), (5,9,55), (37,5,55)
8	r	(4,5,14)	37	I	(4,5,19,55)
9	x	(15), (15,5)	38	\	(5,13,55)
10	C	(5,16)	39	|	(5,33,55)
11	u	(5,17)	40	}	(4,5,14,55)
12	S	(5,18)	41	+	(5,6,41,55), (5,26,40,41,55)
13	i	(4,5,19)	42	'	(43)
14	M	(5,6,11)	43	"	(43,44), (4,43)
15	R	(5,20,21)	44	/	(45)
16	N	(5,22)	45	?	(45,45)
17	m	(5,23)	46	f	(5,46)
18	v	(5,24)	47	h	(5,47)

19	Y	(5,25)	48	k	(4,5), (5,44)
20	D	(5,7,26)	49	[	(48), (5)
21	s	(27), (5,27)	50	{	(5,5), (48,5), (48,48)
22	E	(5,5,29)	51	Z	(49)
23	d	(5,30), (5,31)	52	Z	(50)
24	X	(5,13), (5,29)	53	H	(55)
25	B	(4,5,7,32)	54	Q	(51)
26	g	(13), (29)	55	Q	(53)
27	c	(5,33)	56	_	(52)
28	p	(4,5,34)	57	U	(54)
29	G	(35),(5,36)			

The recognition accuracy of this Roman script recognizer is observed to be above 98% for both capital and lower case letters and their combinations. On the other hand, if the script is recognized as Gurmukhi for the word under consideration then the handwritten word is passed to Gurmukhi script engine. The implementation steps of Gurmukhi script recognizer are further discussed in this paper. In this way, two different script recognition engines are working together to identify these mixed handwritten strokes.

IV GURMUKHI DATASET

For taking handwriting samples of Gurmukhi script characters from various writers, a minimal set of Punjabi words is selected. During this selection process, 28 different Punjabi words have been observed, which further used to collect the handwriting samples of each Gurmukhi symbol at least once from every writer. Table5. shows all these selected words. All the symbols such as consonants, vowels and half characters of Gurmukhi script have been covered As discussed in the previous section, Tablet-PC (Acer-R7 model) is used as Digitizer and active stylus is used for writing on digitizer's surface. By using the hardware technology like Digitizer sample collection process generates number of strokes. Where a stroke is the collection of all the points, which can be touched by the stylus on the surface of digitizer between the events of Pen_Down() and Pen_Up(). The Pen_Down() event is generated when the pen touches the surface of digitizer. Pen_Up() event is generated when after giving handwriting sample the tip of digital pen leaves the surface of digitizer. During dataset generation, 55 different stroke classes have been identified. All these strokes are given unique stroke id's (S_ID) and represented by Table1. A single stroke class or the specific combination of these classes represents different symbols of Gurmukhi script. Table2. shows the 57 different symbols of Gurmukhi script. Each symbol itself represents a single character. The information about the mapping of different stroke classes together to form Gurmukhi characters is also represented by Table2. For example, the character represented by character id C_ID=3 in Table2. is formed with the help of stroke classes having stroke id's 5, 6 and 7 in Table1. This mapping is represented by stroke mapping field in Table2. Some symbols have been observed to have more than one mappings of stroke classes. For example, character Id's 2,4,6 etc. in Table2 shows more than one mappings. Figure2. represents a snapshot the database for a single complete word of Punjabi language " fNeN ". The database information reflect the identity of the writer, by this dataset. Fifty different writers contributed in the dataset generation by providing more than 8400 handwriting samples while capturing approximately 89000 strokes for training and test sets. These writer belongs to different regions and age groups. The list of writers consists of Govt. employees working in Punjab circle as their official communicating mode is Punjabi language, school students, good and average Punjabi writers as per their writing skills. The main motive behind the selection of these words is to cover all the symbols of Gurmukhi script at least once.

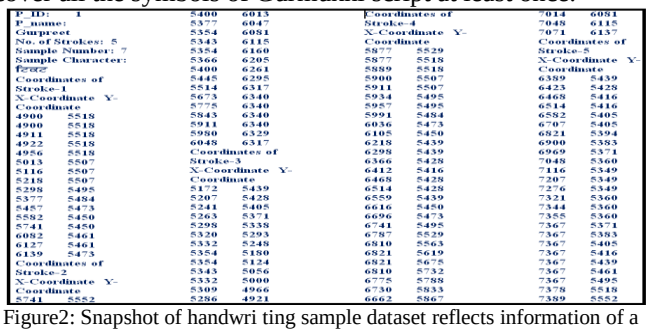

Figure2: Snapshot of handwri ting sample dataset reflects information of a complete Gurmukhi word

number of strokes used to write the complete word, the sample number, name of the word under consideration and co-ordinate points information for each stroke. In Figure2.the name of the writer is represented as "Gurpreet", total number of strokes are 5 and it reflects that, it is the 7th sample provided by the writer in the database. This information maintained by the dataset further helps in classification phase of handwriting recognition process.

V PRE-PROCESSING PHASE

This phase of handwriting recognition process is very important as it deals with the preparation of meaningful input data for the system. During data acquisition process, the main focus is to get information about all co-ordinate points from the surface of digitizer, which are used in the formation of a single stroke [1,2]. This stroke data can be captured with the help of ink objects like "InkOverlay" and "InkCollector" using C#.Net implementation environment [32]. While capturing strokes information some kind of noise can be observed because of the factors like: the quality of digitizer used to capture strokes, familiarity of writers with digitizer, speed of writing etc. To deal with these issues of noise, following pre-processing algorithms are implemented:

a. Removal of duplicate or repetitive points from stroke data.
b. Locating missing points from stroke information.
c. Normalization of stroke data.

a) Removal of duplicate points from stroke data: When stylus touches the surface of digitizer to generate handwritten samples, then the overall process produce some set of co-ordinate points for each stroke, sometimes few co-ordinate points may repeat in these sets. There are number of reasons for these repeated co-ordinate entries like familiarity level of writer with digitizer and stylus, writer's speed, the way of writing etc. Algorithm 5.1 describes a method used in this paper to deal with the issue of removal of repetitive points from captured stroke information.

Algorithm 5.1: Removal of Repetitive points:
Step 1: Consider a stroke 'S' and two empty arrays Xval[] and Yval[].
Step 2: Set i=0
Step 3: Point [] points = S.getpoints() // store the coordinates information of each point of stroke S in array "points".
Step 4: For each point in "points" array repeat Step 5 to Step 11
Step 5: if i=0 then
Set Xval[0]=point.x
Set Yval[0]=point.y
Step 6: Set j=0
Step 7: while j<i repeat steps 8 to 9
Step 8: if point.x=Xval[j] &&point.y=Yval[j] then
Set count=1
Step 9: Set j=j+1
Step 10: if count=0 then
Set Xval[i]=point.x
Set Yval[i]=point.y
Step 11: Set i=i+1
Step12: Stop

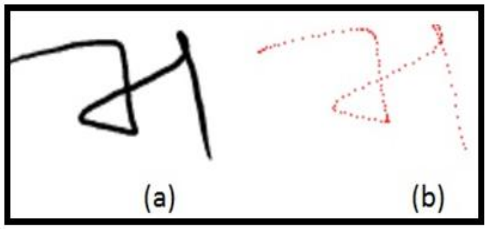

Figure3. (a) Handwritten stroke of Gurmukhi script character "Mumma"; (b) Pre-processed output of character "Mumma"

(3368,3266),(3379,3254),(3425,3243),(3470,3243),(3527,3232),(3595,3221)(3664,3198),(3743,3187)
(3812,3164),(3891,3153),(3994,3142),(4073,3131)(4119,3131),(4164,3142),(4198,3164),(4221,3187)
(4255,3221),(4289,3254)(4312,3311),(4335,3356),(4358,3423),(4369,3502),(4392,3592),(4392,3671)
(4403,3761),(4403,3840),(4403,3908),(4403,3975),(4403,4043),(4392,4088)(4392,4133),(4392,4167)
(4380,4178),(4358,4178),(4324,4167),(4289,4155)(4244,4121),(4210,4076),(4164,4020),(4119,3975)
(4073,3930),(4016,3885)(3971,3862),(3914,3851),(3868,3840),(3857,3840),(3846,3840),(3857,3817)
(3903,3806),(3971,3784),(4062,3761),(4153,3739),(4233,3716),(4324,3682)(4392,3649),(4506,3626)
(4597,3592),(4676,3547),(4756,3502),(4813,3457)(4858,3423),(4881,3378),(4892,3333),(4892,3299)
(4881,3266),(4858,3243)(4836,3221),(4801,3209),(4790,3198),(4779,3198),(4779,3198),(4779,3209)
(4779,3232),(4790,3254),(4801,3288),(4824,3322),(4836,3367),(4847,3435)(4870,3513),(4892,3626)
(4915,3750),(4938,3862),(4961,3998)

Figure4: Sample Stroke data of Character "Mumma" of Gurmukhi script

The above mentioned algorithm considers two single dimensional arrays as Xval[] and Yval[] to store the corresponding values of X and Y co-ordinates of each stroke respectively. "S" represents a captured stroke and holds the value of each point that is touched by the writer on the surface of digitizer while giving handwriting sample for that stroke. "points[]" is an array of type "Point" (A class in C#.Net) and is used to access co-ordinate values of stroke "S". In this algorithm, one by one sequentially the values of stroke "S" are added to arrays Xval[] and Yval[] for their respective co-ordinate points. But before adding these values to the arrays, the values are compared with the existing contents of arrays Xval[] and Yval[]. If no such previous similar entry is found then the values are added to the arrays otherwise discarded. Figure3(a) shows a handwritten stroke of character "Mumma" of Gurmukhi script and Figure3(b) shows its pre-processed output. Figure4 shows the stroke data of above said handwriting sample. Initially, there are total 83 co-ordinate points in the formation of stroke. The co-ordinate point (4779, 3198) repeated two times in the stroke information. So, after the execution of algorithm 5.1 these kind of duplicate entries are removed from the initial stroke information.

Algorithm 5.2: Identification of missing points of strokes

Step 1: Set i=0
Step2: Repeat steps 3 to 4 for n-1 values of Xval[] and Yval[]
Step3:
$SetDistance[i] = \sqrt{(Xval[i+1] - Xval[i])^2 + (Yval[i+1] - Yval[i])^2}$
Step4: Set i=i+1

Step5: Set D_{Avg}= Average(Distance[])
Step6: Set j=0
Step7: Scan for all values of Distance[] and repeat steps 8 to 9
Step8: if $(Distance[j] \geq 2 * D_{Avg})$ *then*
Set X_1=Xval[j], X_2=Xval[j+1], Y_1=Yval[j], Y_2=Yval[j+1]
Shift_val(Distance[], j+1)
Shift_val(Xval[], j+1)
Shift_val(Yval[], j+1)

$$SetXval[J+1] = \frac{X_1 + X_2}{2}$$

$$SetYval[J+1] = \frac{Y_1 + Y_2}{2}$$

$$SetDistance[j+1] = \sqrt{(Xval[j+1] - Xval[j])^2 + (Yval[j+1] - Yval}$$

Step9: Set j=j+1
Step10: Stop

b) Locating missing points from stroke information: Sometimes while giving handwriting samples or during strokes formation, writers may miss to touch some points on digitizer's surface. All this happens because of the factors like the quality of digitizer or stylus, handwriting styles, handwriting speed, familiarity level of writers with hardware etc. These points may play an important role in the identification of stroke's features like shape, structure etc. To add these new points in the existing stroke information, algorithm5.2 describes a method. So, after the execution of Algorithm5.1 the finalized arrays Xval[] and Yval[] are passed as input to Algorithm5.2. The above algorithm creates a new single dimensional array Distance[]. A single element of array Distance[i] represents the graphical distance between i^{th} and $(i+1)^{th}$ elements of arrays Xval[] and Yval[] by considering their values as X-coordinates and Y-coordinates respectively.

After the generation of Distance[] array, the average value of its elements is calculated as D_{AVG}. The next steps of Algorithm5.2 checks the possibility of generation of new points between two consecutive points of stroke 'S'. These new points can be considered if, the distance between two consecutive points of stroke 'S' must be greater than or equal to the double of the value of D_{AVG}. These new points can be identified by using mid-section formulae. For the stroke data represented by Figure5, the value of D_{AVG}= 59. In the given example, there exists two sets of points, where the distance between the points is greater than the double of D_{AVG}. These sets of points are [(4892,3626) , (4915,3750), Distance=126] and [(4938,3862), (4961,3998), Distance=138]. So these sets of points introduce two new points as (4904,3688) and (4950,3930) respectively by using mid section formula. Now update arrays Xval[] and Yval[] by adding the values of these calculated points as the output of Algorithm 5.2.

c) *Normalization of Stroke data:* The applications like Online handwriting recognition system have to define a paint area where the writers can generate handwriting strokes by touching the surface of digitizer with stylus. The co-ordinate system of paint area is actually sub-part of the overall screen co-ordinate system of computers. The co-ordinate system of screen pixels always consider origin as (1,1). But in case of paint area, its origin can be judged from the location of appearance of paint area on computer screen. So to normalize the origin of captured stroke to (1,1), Algorithm5.3. is used.

Algorithm 5.3: Normalization & Centering of Strokes:
Step1: Set X_{max}=Max(Xval[]), Y_{max}=Max(Yval[]), X_{min}=Min(Xval[]), Y_{min}=Min(Yval[]).
*Step2: Create an initial matrix 'I' and final matrix 'F" of order X*Y where: X=(X_{max}-X_{min}) and Y=(Y_{max}-Y_{min}).*
Step3: Set Origin as (X_{min}, Y_{min}) for 'I' and (1,1) for 'F'.
Step4: Set I(X, Y)=1 as per the corresponding values of X and Y in Stroke 'S' and Set other values of I(X,Y)=0.
Step5: Assign the values of I(X, Y) to F(M, N) for corresponding positions. Now 'F' is the final stroke matrix in normalized form.
Step6:Resize the final matrix F to (20 X 10) order.

In this algorithms, maximum values as well as minimum values of x and y co-ordinates are calculated. From these calculated values the actual order of matrix containing that single stroke is observed. After the calculation of matrix order, next part is to set (X,Y) position in matrix as "1" where (X,Y) is one of the entry of stroke data, otherwise set (X,Y)=0 in final matrix 'F' having origin as (1,1). At the end, the final matrix 'F' is resized to order (20 X 10). The entries of this resized matrix are used to give initial values to the MLP neural network's input layer for the purpose of stroke classification in the next phases.

VI SEGMENTATION OF STROKES

Segmentation is one of the most important step of handwriting recognition applications. This process provides a meaningful information to further steps i.e. feature extraction and classification [33]. Following three different levels of segmentation are performed in this paper:

a) Segmentation at Stroke level
b) Segmentation of Headline stroke
c) Segmentation at Zone level

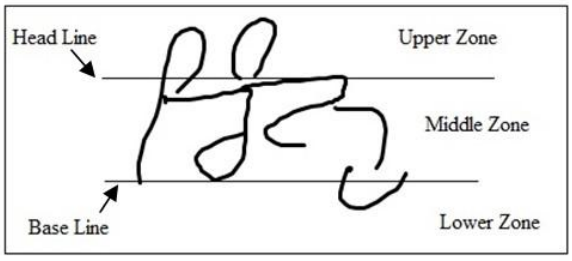

Figure5. Representation of different zones in a complete Gurmukhi word.

First and the most important step in case of Online handwriting recognition systems is the identification of handwritten data as ink strokes. Here the strokes are identified on the basis of their structures. The information about the structure or the shape of the stroke is captured by giving the emphasis on the directional movement of digital pen during stroke formation. Second step of the segmentation process is to find or extract the information about the stroke, which can be considered as Headline in case of Gurmukhi script. Here, the "Headline" is a special stroke.

It is used to join different characters of Gurmukhi script together to form a complete words. It is a horizontal line which appears on the top of all the characters forming a complete word. The identification of headline stroke further helps in the segmentation of complete word of Gurmukhi script into character. The separation of headline stroke from a complete word, provides more information about different characters, used in the formation of that word.

Third and the last level of segmentation process involves in the identification of Upper, Middle and Lower Zones. Upper zone of a complete word contains the information about the part of different strokes which appears above headline. Usually, some complete vowels or small portions of some vowels appears in upper zone. The character id's 42,43,44,45,46,47,51,52 and 53 in Table1 represents these vowels, which appears in upper zone. The character id's 1 and 57 of Table1 represents two special consonants of Gurmukhi script, some part of these consonants also appears in upper zone. Middle zone or Core zone is the area existing between baseline and headline strokes. This part of handwriting sample mainly consist strokes meant for consonants of Gurmukhi script and some portion of vowels (Character Id's 46, 47 and 48 of Table1). The last one is Lower zone. This zone contains the information about some vowels like Character Id's 49 & 50 and half characters of Gurmukhi script like Character Id's 54,55& 56 of Table1. All the three zones are represented clearly in Figure5, which shows a complete word of Gurmukhi script.

VII FEATURE EXTRACTION PROCESS

After performing the segmentation process, raw information about the handwriting samples in the form of stroke data is captured. This information itself contain many hidden facts. These hidden facts further helps in recognition process. These facts are considered to be different features on the basis of which, strokes can be categorized.

The identification of different strokes is done according to the different shapes represented in Table2. In this paper, low level or local features like directional aspect of pen movement during stroke formation, location of strokes according to co-ordinates system of digitizer, formation of loops and area covered by a single stroke are considered for feature extraction process. For high level features, strokes representing headline, straight lines and dots are considered. Zoning is also considered as a statistical feature. Following algorithms presents a way, to identify stroke representing Headline for Gurmukhi scripts words.

Algorithm7.1: Identification of Headline Stroke:

Step1: For each Segmented Word W_i in input data repeat steps 2 to 11.
Step2: Refresh array Headline_S[] // Remove all the entries of array Headline_S[]
Step3: Set k=0 and j=0
Step4: For each Stroke S_j in W_i repeat steps 5 to 8
Step5: S_ID = MLP_Classifier (S_j)
Step5: Set Height= Max_Y(S_j) - Min_Y(S_j)
Step6: if Height <= 3 and (S_ID=5 or S_ID=48) then
Set Headline_S[k]=j
Step7: Set k=k+1
Step8: Set j=j+1
Step9: Set C= Count (Headline_S[])
Step10: if C>1 then
Merge_Strokes(Headline_S[]) // Merge number of stokes to a single stroke by forming a straight line
Step11: Stop

In case of Gurmukhi script words, Headline stroke always appear on the top of Gurmukhi characters forming these complete words. This is a special stroke, which join the different characters together to make complete words. Its shape is like a horizontal straight line. The algorithm7.1 detects this stroke from the segmented word information. In the beginning, the algorithm takes a segmented complete word of Gurmukhi script as input. An empty array Headline_S[] is generated to capture the stroke number which represents headline stroke. By using the MLP classifier, which will be discussed in detail in the next section, a stroke is identified which in shape is similar to either S_ID=5 or 48 of Table2. Then the height of the identified stroke is checked. If the height is less than or equal to 3 then the stroke id is added to array Headline_S[]. In the end, different strokes represented by the array are merged together to form a single stroke. This output stroke is considered to be the headline stroke.

Algorithm7.2: Identification of different zones:

Step1: For each Segmented Word W_i in input data repeat steps 2 to 11.
Step2: Set J=0
Step3: For all Stokes S_j in W_i repeat steps 4 to 6
Step4: Set Height= Max_Y(S_j) - Min_Y(S_j)
Step5: if Height>=10 then
set Base_Stroke=j and go to step 7
Step6: Set j=j+1
Step 7: Set Baseline_Y = Min_Y(S_j) - 1
Step8: Refresh array Zone_S[] // Remove all the entries of array Zone_S[]
Step9: Set i=0 and j=0
Step10: For each Stroke S_j in W_i repeat steps 11 to 14
Step11: Set Stroke_Y_Mid= (Max_Y(S_j) - Min_Y(S_j)) / 2
Step12: Set Headline_Y_Mid= (Max_Y(Headline) - Min_Y(Headline)) / 2
Step13: if Stroke_Y_Mid<Baseline_Y then
Set Zone_S[j]="LZ"
else if Stroke_Y_Mid>Headline_Y_Mid then
Set Zone_S[j]="UZ"
else
Set Zone_S[j]="MZ"
Step14: j=j+1
Step15: Stop

After the identification of headline stroke, the next part is to find the appearance of each captured stroke in different zones as Upper, Middle and Lower, as discussed earlier in Figure5.

Algorithm7.2 is used to find these zones for all the captured strokes. The first part of the algorithm find the baseline stroke of the segmented word of Gurmukhi script. Then the second part of the algorithm updated the array Zone_S[]. This array is used to store the zone information of all strokes. The zones are identified according to the criteria that, if the mid value of y-coordinate of the stroke under consideration is above the location of headline stroke then the zone is identified as Upper zone and array is updated with the value "UZ". If that mid value is below the baseline stroke then the array is updated as "LZ" as the identified zone is lower zone. Other than these cases the zone of stroke is considered to be as middle zone and the value of array will be updated with the value "MZ".

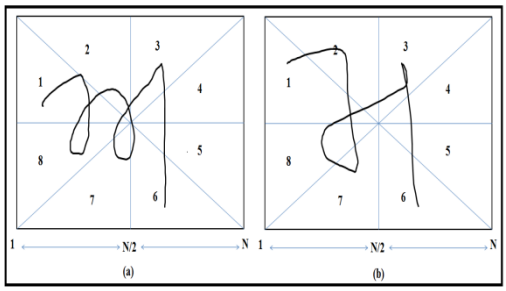

Figure6: N x N space matrix used for directional code generation for strokes

Algorithm7.3: Generation of directional code:

Step1: Consider a Stroke "S"
Step2: Set $X_{max} = Max\_X(S)$
$X_{min} = Min\_X(S)$
$Y_{max} = Max\_Y(S)$
$Y_{min} = Min\_Y(S)$
Step3: Set $X = X_{max} - X_{min} and Y = Y_{max} - Y_{min}$
Step4: Set $N = Max(X, Y)$
Step5: if $N\%2 == 0\ then set N = N + 1$
Step6: Create a matrix M or order N x N and map the values of stroke "S" in "M"

Step7: Set Z=0 and U_{id}=" "
Step8: Repeat step 9 for all the values of stroke "S" in matrix "M" in the sequential order of stroke's points
Step9: $If\ \left(S_x \geq 1\ and S_x \leq \frac{N}{2}\right) and\ \left(S_y \geq \frac{N}{2} + 1\ and S_y \leq N then\right.$

{
$if(S_x > S_y)$
{
$if\ (Z! = 1)$
{
Set $U_{id} = U_{id} + "1"$
Set Z=1
}
}
else
{
$if\ (Z! = 2)$
{
Set $U_{id} = U_{id} + "2"$
Set Z=2
}
}
}
else $If\ \left(S_x \geq \frac{N}{2} + 1\ and S_x \leq N\right) and\ \left(S_y \geq \frac{N}{2} + 1\ and S_y \leq N then\right.$

{
$if(S_x > S_y)$
{
$if\ (Z! = 4)$
{
Set $U_{id} = U_{id} + "4"$
Set Z=4
}
}
else
{
$if\ (Z! = 3)$
{
Set $U_{id} = U_{id} + "3"$
Set Z=3
}
}
}
else $If\ \left(S_x \geq \frac{N}{2} + 1\ and S_x \leq N\right) and\ \left(S_y \geq \frac{N}{2} + 1\ and S_y \leq N 2 then\right.$

{
$if(S_x > S_y)$
{
$if\ (Z! = 5)$
{
Set $U_{id} = U_{id} + "5"$
Set Z=5
}
}
else
{
$if\ (Z! = 6)$
{
Set $U_{id} = U_{id} + "6"$
Set Z=6
}
}
}
else
{
$if(S_x > S_y)$
{
$if\ (Z! = 7)$
{
Set $U_{id} = U_{id} + "7"$
Set Z=7
}
}
else
{
$if\ (Z! = 8)$
{
Set $U_{id} = U_{id} + "8"$
Set Z=8
}
}

}

Step10: Stop and check the value of U_{id}

Algorithm7.3 is used to generate a unique code (U_{id}) for every stroke given as handwriting sample by the writers. In actual, these codes are directional codes. The main purpose of these codes is to trace the directional movement of stylus while generating handwriting samples. Figure6(a) represents the stroke id 2 of Table2. The directional code generated for this stroke is "12181235678123456". Where, Figure6(b) represents the sample for stroke id 37 of Table2 and the directional code calculated by the algorithm for this stoke is "121878123456". The flow of the algorithm at first shows that, The horizontal and vertical lengths of the stroke under consideration are calculated. Then the maximum value from these two decides the order of the directional matrix M. The order is represented as NxN. After that, the stroke data is normalized within the boundaries of this NxN matrix. The area covered by this matrix is divided into 8 equal parts. All these parts represents eight different directions 1,2,3.....,8 as specified in Figure7. Then the stroke points are considered one by one in the sequential order as per their appearance at the time of stroke capturing. According to the position of a each point in its respective directional zone, a string is generated as U_{id} by concatenating the identified directional symbol of stroke at the end of the string U_{id}. If the last character of U_{id} represents the same symbol as identified for that particular point of the stroke then no concatenation will be performed and the U_{id} remains as it is. After repeating the same process for all stroke points, the final value of U_{id} represents the unique directional code for the stroke under consideration.

VIII CLASSIFICATION PHASE

After the handwriting recognition phases such as Pre-processing, Segmentation and Feature extraction, the next step is classification of strokes and classification of different characters present in a sequential order in the segmented word.

The input to this phase is the collection of earlier identified segmented words and the strokes used in the formation of these words. In this paper Multi-Layered Perceptron (MLP) neural network is used for the purpose of classification. The power of Artificial Neural Network (ANN) to deal with the complex problems where unstructured or large data is involved is the reason behind the selection of MLP neural network for this handwriting recognition process.

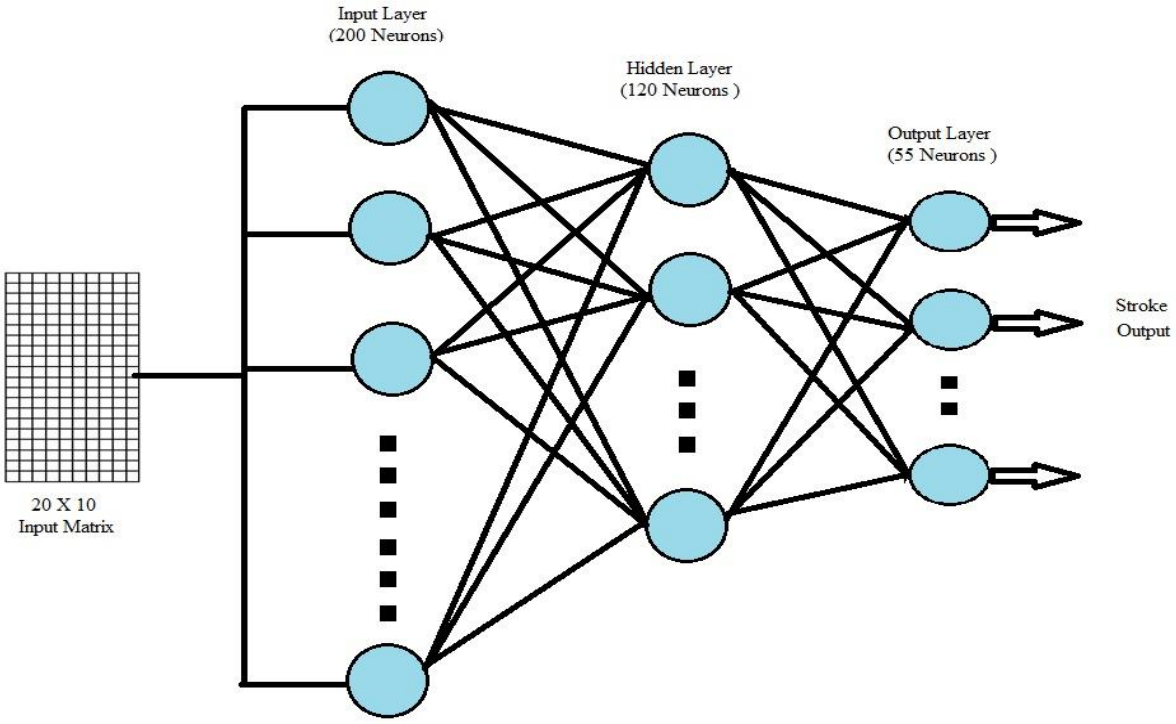

Figure7: Architecture of MLP Neural Network for Strokes identification

Three layered architecture of MLP neural network is used for classification. The architecture is represented by Figure7. The first layer is Input layer. This layer contain 200 computing units as neurons. The input is fed to this layer with the elements of (20 X 10) matrix. This matrix is the resultant of the pre-processing step as discussed earlier in Algorithm5.3. Structural information of every single stroke is also represented by this matrix.

Algorithm8.1: MLP classifier for stoke identification

Step1: Initiate Input layer of MLP neural network with 201 neurons by assigning the corresponding entries of matrix M. The initial values are consider to be X_0, X_1, X_2, ..., X_{200}, where X_0 represents input as Bias.

Step2: Initialize the weights of network by random values as W_0, W_1, W_2, ..., W_{200}.

Step3: Take 120 neurons in Hidden layer. Now iterate and compute:

$$a(x) = b + \sum_{i=1}^{200} W_i X_i$$

for each neuron by considering b as bias. a(x) is the

input at hidden layer.

Step4: Consider 55 neurons at Output layer for each stroke type. Output can be computed by using normalized exponential function represented by Equation 1.

Step5: Compare the computed output with the target output. In case of a perfect match, show the output and exit.

Step6: Else update weights as:

$$W_i = W_i - \alpha X_i where i = 1,2,3, \dots \dots \dots ,200$$

α is the learning rate and act as a constant.

Step7: Go to step 3.

Algorithm8.1 is used for the implementation of MLP classifier. Every neuron or computing unit present in the input layer takes the input from the corresponding 200 elements of the pre-processed matrix. Each matrix represented a single stroke used in the formation of segmented word. The second layer of neural network is the hidden layer and it contains total 120 neurons for the purpose of training. Third and the last layer is the output layer. The output layer have 55 computing units. These 55 units are in actual the different classes representing the stroke categories. So the output of MLP neural network classifies or identifies the different segmented strokes according to their S_ID's present in Table2. Random weights are assigned to neurons and the activation function of Softmax is used to find the probability distribution of different classes for stroke identification at output layer.

$$\sigma(Z)_j = \frac{e^{Z_j}}{\sum_1^k e^{Z_k}} for J = 1 \dots \dots \dots \dots . . k \quad (1)$$

Equation1. represents the normalized exponential function. This is used to find the probability distribution of all the output units. After finding probability distribution, stroke identification is performed by checking the largest probability of node present in output layer.

Here 'j' represents a single class under consideration at output layer, 'k' represents the total number of stroke classes, 'Z' represents the unknown stroke under consideration at input layer and $\sigma(Z)_j$ represents the value of probability of 'Z' for j^{th} unit at output layer . After the identification of stroke id's for all the strokes belongs to segmented word W_i. The mapping of these strokes is checked according to Tab1e1. In this way different characters are identified. These characters together represents the identification of a complete word under consideration.

IX RESULTS & DISCUSSION

In this paper, Online handwriting recognition problem for bilingual scripts has been addressed. The input to the developed system is given in the form of handwritten text containing the words of Punjabi language and English language. The major part of the text considered to be written by using Gurmukhi script as Punjabi language words. English words have been written by using Roman script to provide clarification about some difficult Punjabi words used in the text. So, in the input text almost 90% words belongs to Punjabi language and rest 10% belongs to English language. Figure8. reflects a snapshot of the implemented system. Figure8(a) shows the input given to the system with the help of Tablet PC and active stylus as a handwriting sample. This sample of handwriting contain two words written by using Gurmukhi script and other two words i.e. "SLIET" and "Longowal" written by using Roman script. Figure8(b) shows the output generated by the system after the execution of Pre-processing phase. Figure8(c) represents the final output generated by the system as the digital text.

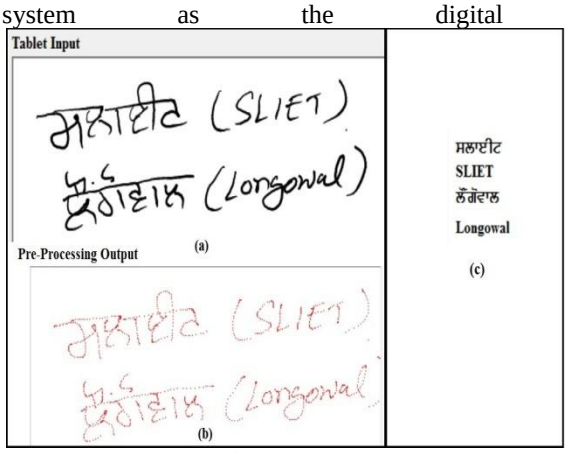

Figure8: Snapshot of bilingual handwriting recognition system

The Algorithm7.1 used for the identification of Headline stroke during feature extraction process provides 100% success rate in all training and testing cases. The overall results for Algorithm7.2 are observed to have about 93% of accuracy. The reason behind this decrease in the zoning accuracy is the appearance of some character in more than one zones. C_ID 1,46,47 and 57 of Table3 have their appearance in more than one zones. These zones are Middle zone and Upper zone. Table2. show the different strokes used for the purpose of classification of characters and words of Gurmukhi script along with their stroke ID's. Table2. also represents the recognition rate of these strokes by the MLP neural network. The recognition accuracy of strokes used for representing "(" and ")" is observed as 100%. Which reflects the 100% results of the identification process that, the segmented word under consideration belongs to Gurmukhi script or Roman script.

Table2: Recognition %age of strokes for Gurmukhi script

S_ID	Stroke	%age	S_ID	Stroke	%age	S_ID	Stroke	%age	S_ID	Stroke	%age
1		100	15		95	29		87	43		98
2		97	16		86	30		92	44		100

3		98	17		88	31		93	45		94
4		100	18		97	32		97	46		100
5		100	19		92	33		97	47		100
6		94	20		93	34		90	48		98
7		95	21		96	35		96	49		93
8		96	22		89	36		93	50		100
9		95	23		90	37		96	51		97
10		97	24		92	38		98	52		95
11		100	25		97	39		97	53		98
12		90	26		94	40		100	54		87
13		91	27		93	41		100	55	.	100
14		100	28		100	42		100			

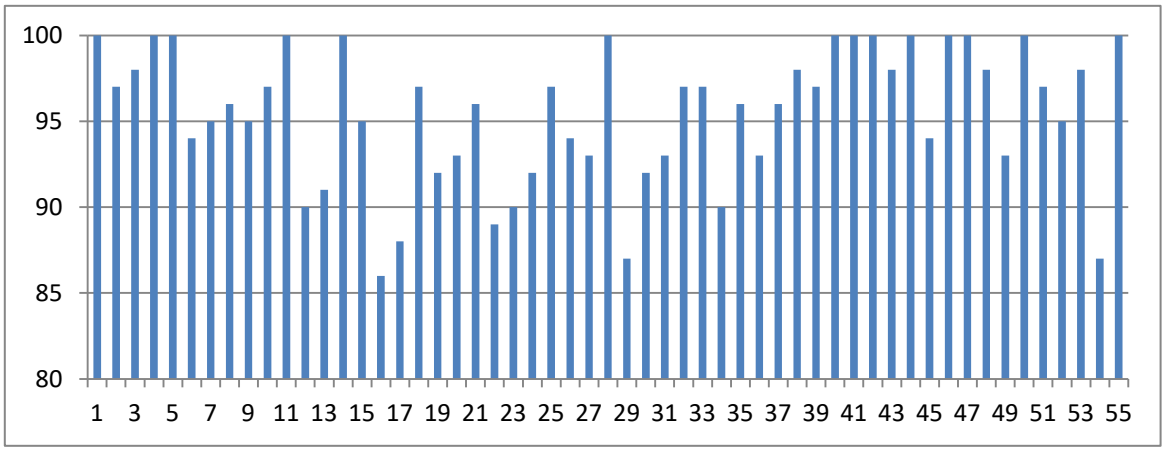

Figure9: Recognition %age of strokes for Gurmukhi script

The classification of Roman characters and words have been performed with the help of existing script classification function in C#.Net. This classification function is present in "InkOverlay" class of .Net library. The recognition percentage of all characters and complete words belongs to Roman script has been observed above 97%. The overall recognition percentage of Table2 has been recorded as 95.47%, which represents different strokes of Gurmukhi script's character set. The similar structure of some stroke classes affect the overall recognition percentage. These sets of stroke classes are {1 & 54}, {2 & 4}, {6 & 9}, {12 & 19} and {17, 25, 30 & 31}. The problem of these confusing strokes has further been addressed at the time when the sequence of strokes have been combined to form characters or words as per the mapping represented in Table1. In this paper, Figure9.represents the graphical view of Table2. for the purpose of clarity.

Table3: Recognition %age of Gurmukhi Characters

C_Id	Character	Recognition %age	C_Id	Character	Recognition %age
1	T	100	30	w	96
2	n	98	31	:	98
3	J	95	32	o	100
4	;	96	33	b	100
5	j	97	34	t	94

6	e	100	35	V	100
7	y	95	36	P	97
8	r	98	37	I	98
9	x	95	38	\	96
10	C	86	39	|	99
11	u	88	40	}	96
12	S	98	41	+	100
13	i	96	42	'	98
14	M	96	43	"	96
15	R	94	44	/	94
16	N	95	45	?	97
17	m	97	46	f	96
18	v	93	47	h	100
19	Y	98	48	K	93
20	D	94	49	[	100
21	s	98	50	{	98
22	E	96	51	Z	94
23	d	95	52	Z	91
24	X	97	53	H	100
25	B	96	54	Q	91
26	g	92	55	Q	93
27	c	97	56	_	100
28	p	95	57	U	100
29	G	96			

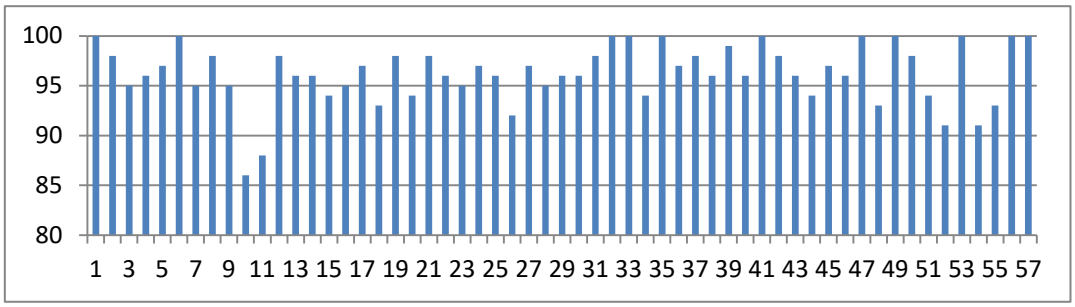

Figure10: Recognition %age a isolated characters of Gurmukhi script

Table4: Accuracy of OHR systems for Gurumukhi Script at Character Level

S. No.	Authors	Method	Recognition %age
1	A. Sharma et al.[29]	Elastic String Matching	90.08
2	A. Sharma et al. [35]	HMM	91.95
3	A. Sharma et al.[34]	Smallest Line Segment	94.69
4	R. Kaur et. Al. [23]	KNN/SVM	89.64
5	**Proposed Method**	**MLP**	**96.24**

Table3. represents the character set of Gurmukhi script. These characters are used to write text in Punjabi language in Indian Regions. Total 57 characters are there in Gurmukhi script. These are the combination of Consonants, Vowels and Half characters. Table3. also reflects the recognition accuracy achieved for these characters by the developed system. The overall recognition rate for Gurmukhi characters has been observed as 96.24%. Table4. represents the work done by other researchers in the field of online handwritten character recognition for Gurmukhi script. Previously, Sharma et al. [34] achieved maximum 94.69% recognition rate at character level for Gurmukhi script. They used the small line segment approach in their research. Figure10. represents the results of Table3 in a graphical form.

The system discussed in this paper shows the improvement of 1.55% for isolated character recognition process.

Table5: Recognition %age of Complete words of Gurmukhi script

S.No.	Word	Recognition %age	S.No.	Word	Recognition %age
1	;ogzu	94	15	|;b	98.33
2	\op{ik	96.33	16	tZSk	94
3	Peb	99	17	I?pok	96.6
4	nwo{d	97.4	18	gqqPB	94.5
5	UeV	100	19	n"PXh	97.6
6	Fsozrk	96.5	20	go:'r	97.2
7	u[}bh	96.4	21	fJeZm	95.8
8	jZE	94.66	22	Gkd"A	94.75
9	T[;skd	97	23	MoBk	96.25
10	Y'b	98.66	24	c[ZcV	97.6
11	fJziD	95	25	fCnkB	93.8
12	y/vDk	93.8	26	;zGk+	95.2
13	fozBQ	95.2	27	fJzR	94
14	xZN	93.66	28	;_o	98.66

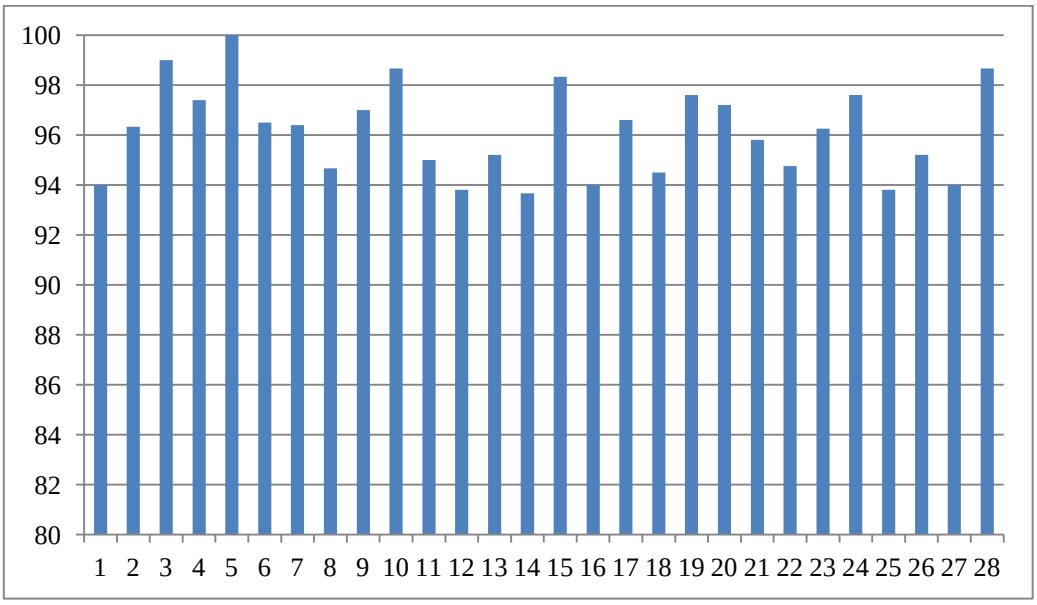

Figure11: Recognition %age of Gurmukhi Dataset words

Table6: Accuracy of OHR systems for Gurmukhi script at Word level

S.No	Authors	Method	Recognition %age
1	A. Sharma et al. [36]	Rearrangement of Strokes	81.02
2	M. Sachan et al. [12]	Nearest Neighbour /SVM	86.90
3	S.Singh et al. [37]	KNN/HMM	85.00
4	Proposed Method	MLP	89.14

A dataset based on 28 Gurmukhi script words is presented in this research work. These selected words cover all the characters of Gurmukhi script. Total 50 writers of different age groups and professions contributed in the formation of this dataset. Table5. represents the recognition results of these closed vocabulary Gurmukhi words. The overall word level recognition for this dataset is observed as 96.13%. The experiment is also performed on some open vocabulary words. The word level recognition rate for those open vocabulary Gurmukhi script words is observed as 89.14%. Table6. reflects the work done by other researchers in the field of Online handwritten Gurmukhi script recognition. The authors of Table6.worked on complete word level handwriting recognition systems. Previously, the highest recognition rates for complete word level recognition of Online Gurmukhi script has been observed as 86.90% by Sachan et al. [12] and 85% by Singh et al. [37]. Figure11. presents the graphical view of Table4. Online handwriting recognition work on bilingual English and Punjabi languages till date is not addressed by any author. The implemented system experienced the recognition accuracy of 93.07% for the bilingual system, when both English and Punjabi words are considered together.

X CONCLUSION & FUTURE SCOPE

Automatic handwriting recognition is always considered as one of the most complex computing problems. As already discussed in this paper, the applications of handwriting recognition are either considered under Online handwriting recognition category or Offline handwriting recognition category. Both the process differ from each other on the basis of input provided. In this paper, an Online handwriting recognition system is presented, which consider two different scripts, Roman and Gurmukhi for English and Punjabi languages respectively. The input is given with the help of Tablet pc and active stylus. The handwritten text input consist both Gurmukhi script and Roman script words. To improve the quality of the input supplied, pre-processing steps like removal of repetitive stroke points, identification of missing points and normalization are performed. Segmentation step is performed at the level of stroke identification, headline extraction and identification of different writing zones. The recognition rate of 100% is achieved for script identification as Roman or Gurmukhi. In case, the extracted word is recognized as English word then Roman script identification engine takes that word and convert it to its digital form by considering the character set of Roman script. Almost 97% recognition accuracy has been observed from Roman script engine. If the word under consideration is identified as Punjabi word, then every stroke represented that word is passed to Gurmukhi script recognition engine, where MLP neural network is used as a classifier. At the level of segmentation, headline stroke identification process achieved the recognition accuracy of 100%, Zone identification process achieved the recognition accuracy of 93%. Classification process observed recognition rates of 94.47% and 96.24% for strokes classification and isolated character identification respectively. A compact dataset of 28 Gurmukhi words is selected to take handwritten samples of writers. More than 8400 handwriting samples are collected. For the closed vocabulary of this dataset, the recognition rate is observed as 96.13% and the overall recognition of complete Gurmukhi open vocabulary words is observed as 89.14%.
As a future aspect, more than two scripts can also be consider to strengthen the challenge. Multi-layered neural network can further be compared with other classifiers such as SVM, HMM, etc. to check the performance of these classifiers in bilingual script recognition environment.

REFERENCES

1. R. Plamondon, S.N. Srihari, Online and off-line handwriting recognition: a comprehensive survey, Pattern Anal. Mach. Intell. IEEE Trans. 22 (2000) 63–84.
2. C. Tappert, C. Suen, T. Wakahara, The state of the art in online handwriting recognition, IEEE Trans. Pattern. (1990).http://dl.acm.org/citation.cfm?id=83137.
3. G. Singh, M. Sachan, Multi-layer perceptron (MLP) neural network technique for offline handwritten Gurmukhi character recognition, in: 2014 IEEE Int. Conf. Comput. Intell. Comput. Res. IEEE ICCIC 2014, 2015: pp. 221–225. doi:10.1109/ICCIC.2014.7238334.
4. G. Singh, M. Sachan, A Framework of Online Handwritten Gurmukhi Script Recognition, 8491 (2015) 52–56.
5. Languages of World, (n.d.). https://www.ethnologue.com/guides/how-many-languages (accessed May 11, 2017).
6. Mother Tongue, (n.d.). http://www.censusindia.gov.in/Census_Data_2001/Census_Data_Online/Language/gen_note.html (accessed May 11, 2017).
7. D. Keysers, T. Deselaers, H.A. Rowley, L.-L. Wang, V. Carbune, Multi-Language Online Handwriting Recognition, IEEE Trans. Pattern Anal. Mach. Intell. (2016) 1–14. doi:10.1109/TPAMI.2016.2572693.
8. R. Aggarwal, S. Swetha, A.M. Namboodiri, J. Sivaswamy, C. V Jawahar, Online handwriting recognition using depth sensors., in: ICDAR, 2015: pp. 1061–1065. http://dblp.uni-rier.de/db/conf/icdar/icdar2015.html#AggarwalSNSJ15.
9. Z. Sadrnezhad, A. Nekouie, M.V. Jahan, Online handwriting Farsi character and number recognition based on hand movement direction using Hidden Markov Models, in: 2014 Int. Congr. Technol. Commun. Knowledge, ICTCK 2014, 2015: pp. 1–6. doi:10.1109/ICTCK.2014.7033518.
10. H. Zhao, W. Ge, A HMM- based approach for online Uyghur word handwriting recognition, in: Cross Strait Quad-Regional Radio Sci. Wirel. Technol. Conf. (CSQRWC), 2012, 2012: pp. 204–207. doi:10.1109/CSQRWC.2012.6294993.
11. I. Hosny, S. Abdou, A. Fahmy, Using Advanced Hidden Markov Models for Online Arabic handwriting recognition, (2011)

565–569.
12. M.K. Sachan, G.S. Lehal, V.K. Jain, A System for Online Gurmukhi Script Recognition, (n.d.) 299–300.
13. T.R. Indhu, V. Vidya, V.K. Bhadran, Multilingual Online Handwriting Recognition System: An Android App, in: Proc. - 2015 5th Int. Conf. Adv. Comput. Commun. ICACC 2015, 2016: pp. 33–36. doi:10.1109/ICACC.2015.11.
14. Nationalencyklopedin, (n.d.). https://en.wikipedia.org/wiki/List_of_languages_by_number_of_native_speakers (accessed May 11, 2017).
15. O. Samanta, A. Roy, U. Bhattacharya, S.K. Parui, Script independent online handwriting recognition, in: Proc. Int. Conf. Doc. Anal. Recognition, ICDAR, 2015: pp. 1251–1255. doi:10.1109/ICDAR.2015.7333964.
16. M.M. Prasad, 1D-LDA verses 2D-LDA in online handwriting recognition, in: Proc. Int. Conf. Circuits, Commun. Control Comput. I4C 2014, 2014: pp. 431–433. doi:10.1109/CIMCA.2014.7057838.
17. D. Dutta, A.R. Chowdhury, U. Bhattacharya, S.K. Parui, Stroke Level User-Adaptation for Stroke Order Free Online Handwriting Recognition, in: Proc. Int. Conf. Front. Handwrit. Recognition, ICFHR, 2014: pp. 250–255. doi:10.1109/ICFHR.2014.50.
18. S.P. Teja, A.M. Namboodiri, A ballistic stroke representation of online handwriting for recognition, in: Proc. Int. Conf. Doc. Anal. Recognition, ICDAR, 2013: pp. 857–861. doi:10.1109/ICDAR.2013.175.
19. S.D. Chowdhury, U. Bhattacharya, S.K. Parui, Online Handwriting Recognition Using Levenshtein Distance Metric, in: 12th Int. Conf. Doc. Anal. Recognit. Online, 2013: pp. 79–83. doi:10.1109/ICDAR.2013.24.
20. M.A. Abuzaraida, A.M. Zeki, A.M. Zeki, Problems of writing on digital surfaces in online handwriting recognition systems, in: 2013 5th Int. Conf. Inf. Commun. Technol. Muslim World, ICT4M 2013, 2013: pp. 1–5. doi:10.1109/ICT4M.2013.6518879.
21. T.R. Indhu, V.K. Bhadran, Malayalam online handwriting recognition system: A simplified fuzzy ARTMAP approach, in: 2012 Annu. IEEE India Conf. INDICON 2012, 2012: pp. 613–618. doi:10.1109/INDCON.2012.6420691.
22. R. Kunwar, A.G. Ramakrishnan, Online Handwriting Recognition of Tamil Script Using Fractal Geometry., in: ICDAR, 2011: pp. 1389–1393. http://dblp.uni-trier.de/db/conf/icdar/icdar2011.html#KunwarR11.
23. R. Kaur, M. Singh, Stroke based online handwritten Gurmukhi character recognition, in: 2016 Int. Conf. Adv. Comput. Commun. Informatics, ICACCI 2016, 2016: pp. 598–601. doi:10.1109/ICACCI.2016.7732111.
24. S. Baral, S. Bhattacharya, A. Chakraborty, U. Bhattacharya, S.K. Parui, A Machine Learning Approach to Detection of Core Region of Online Handwritten Bangla Word Samples, in: Proc. Int. Conf. Front. Handwrit. Recognition, ICFHR, 2014: pp. 458–463. doi:10.1109/ICFHR.2014.83.
25. T. Mondal, U. Bhattacharya, S.K. Parui, K. Das, D. Mandalapu, On-line handwriting recognition of Indian scripts - The first benchmark, in: Proc. - 12th Int. Conf. Front. Handwrit. Recognition, ICFHR 2010, 2010: pp. 200–205. doi:10.1109/ICFHR.2010.39.
26. K.B. Baiju, K. Sabeerath, Online recognition of Malayalam handwritten scripts — A comparison using KNN, MLP and SVM, in: Int. Conf. Adv. Comput. Commun. Informatics, 2016. doi:10.1109/ICACCI.2016.7732357.
27. A. Aggarwal, K. Singh, Handwritten Gurmukhi character recognition, in: Int. Conf. Comput. Commun. Control, 2015. doi:10.1109/IC4.2015.7375678.
28. D.V. Sharma, G.S. Lehal, S. Mehta, Shape encoded post processing of Gurmukhi OCR, in: 10th Int. Conf. Doc. Anal. Recognition, ICDAR '09., 2009. doi:10.1109/ICDAR.2009.180.
29. A. Sharma, R. Kumar, R.K. Sharma, Online handwritten Gurmukhi character recognition using elastic matching, in: Proc. - 1st Int. Congr. Image Signal Process. CISP 2008, 2008: pp. 391–396. doi:10.1109/CISP.2008.297.
30. S. Gaur, S. Sonkar, P.P. Roy, Generation of synthetic training data for handwritten Indic script recognition, in: Proc. Int. Conf. Doc. Anal. Recognition, ICDAR, 2015: pp. 491–495. doi:10.1109/ICDAR.2015.7333810.
31. R. Kunwar, P. Mohan, K. Shashikiran, A.G. Ramakrishnan, Unrestricted Kannada online handwritten akshara recognition using SDTW, in: 2010 Int. Conf. Signal Process. Commun. SPCOM 2010, 2010. doi:10.1109/SPCOM.2010.5560515.
32. G. Singh, M.K. Sachan, Data capturing process for online Gurmukhi script recognition system, in: IEEE Int. Conf. Comput. Intell. Comput. Res., 2015: pp. 518–521. doi:10.1109/ICCIC.2015.7435778.
33. M.K. Sachan, G.S. Lehal, V.K. Jain, A novel method to segment online Gurmukhi script, in: Commun. Comput. Inf. Sci., 2011: pp. 1–8. doi:10.1007/978-3-642-19403-0_1.
34. A. Sharma, K. Dahiya, Online Handwriting Recognition of Gurmukhi and Devanagari Characters in Mobile Phone Devices, Int. J. Comput. Appl. (2012) 37–41.
35. A. Sharma, HMM based Online Handwritten Gurmukhi Character Recognition, Mach. Graph. Vis. Int. J. 19 (2010) 439–449.
36. A. Sharma, R. Kumar, R.K. Sharma, Rearrangement of recognized strokes in online handwritten Gurmukhi words recognition, in: Proc. Int. Conf. Doc. Anal. Recognition, ICDAR, 2009: pp. 1241–1245. doi:10.1109/ICDAR.2009.36.
37. S. Singh, A. Sharma, I. Chhabra, Online Handwritten Gurmukhi Strokes Dataset Based on Minimal Set of Words, ACM Trans. Asian Low-Resource Lang. Inf. Process. 16 (2016) 1–20. doi:10.1145/2896318.
38. D.L. Rudolf Eignemann, Von Neumann Computers, in: 1998: pp. 1–30. doi:10.1002/pssb.201300062.
39. V. Connelly, D. Gee, E. Walsh, A comparison of keyboarded and handwritten compositions and the relationship with transcription speed, Br. J. Educ. Psychol. 77 (2007) 479–492. doi:10.1348/000709906X116768.
40. W.Y. Leng, S.M. Shamsuddin, Writer identification for Chinese handwriting, Int. J. Adv. Soft Comput. Its Appl. 2 (2010) 142–173.
41. Y. Elarian, I. Ahmad, S. Awaida, W.G. Al-Khatib, A. Zidouri, An Arabic handwriting synthesis system, Pattern Recognit. 48 (2015) 849–861. doi:10.1016/j.patcog.2014.09.013.
42. M. Tanvir Parvez, S.A. Mahmoud, Arabic handwriting recognition using structural and syntactic pattern attributes, Pattern Recognit. 46 (2013) 141–154. doi:10.1016/j.patcog.2012.07.012.
43. H. Lee, B. Verma, Binary segmentation algorithm for English cursive handwriting recognition, Pattern Recognit. 45 (2012) 1306–1317. doi:10.1016/j.patcog.2011.09.015.
44. G. Singh, M. Sachan, Multi-layer perceptron (MLP) neural network technique for offline handwritten Gurmukhi character recognition, in: 2014 IEEE Int. Conf. Comput. Intell. Comput. Res., 2014: pp. 1–5. doi:10.1109/ICCIC.2014.7238334.
45. A.K. Kinjarapu, W. Godavari, A. Pradesh, A. Pradesh, Online Recognition of Handwritten Telugu Script characters, (2016) 426–432.
46. K.B. Urala, A.G. Ramakrishnan, S. Mohamed, Recognition of open vocabulary, online handwritten pages in Tamil script, 2014 Int. Conf. Signal Process. Commun. SPCOM 2014. (2014). doi:10.1109/SPCOM.2014.6984002.
47. M.A. Abuzaraida, A.M. Zeki, A.M. Zeki, Online recognition system for handwritten hindi digits based on matching alignment algorithm, Proc. - 3rd Int. Conf. Adv. Comput. Sci. Appl. Technol. ACSAT 2014. (2014) 168–171. doi:10.1109/ACSAT.2014.36.
48. O. Samanta, U. Bhattacharya, S.K. Parui, Smoothing of HMM parameters for efficient recognition of online handwriting, Pattern Recognit. 47 (2014) 3614–3629. doi:10.1016/j.patcog.2014.04.019.
49. Accredited Language Services, (n.d.). https://www.accreditedlanguage.com/2016/09/09/the-10-most-common-languages/ (accessed October 3, 2017).
50. D. Keysers, T. Deselaers, H.A. Rowley, L.-L. Wang, V. Carbune, Multi-Language Online Handwriting Recognition, IEEE Trans. Pattern Anal. Mach. Intell. (2016) 1–14. doi:10.1109/TPAMI.2016.2572693.
51. H. Suwanwiwat, V. Nguyen, M. Blumenstein, U. Pal, Off-line handwritten Thai name recognition for student identification in an automated assessment system, in: Proc. Int. Jt. Conf. Neural Networks, 2014: pp. 2347–2353. doi:10.1109/IJCNN.2014.6889657.
52. R.S. Zinjore, R.J. Ramteke, Recognition of Handwritten Bilingual Characters- Numerals using Shape Context, in: IEEE Int. WIE Conf. Electr. Comput. Eng., 2016: pp. 19–21.
53. S. Haboubi, S.S. Maddouri, H. Amiri, Discrimination between Arabic and Latin from bilingual documents, in: 2011 Int. Conf. Commun. Comput. Control Appl. CCCA 2011, 2011. doi:10.1109/CCCA.2011.6031496.
54. B.A. Vighnesh, B.M. Yadav, A. Kumar, M. V Raghunadh, Autonomous Bilingual Character Recognition and Writer Identification, 2 (2013) 219–224.
55. B. V Dhandra, Gururaj Mukarambi, M. Hangarge, a Script Independent Approach for Handwritten Bilingual Kannada and Telugu Digits Recognition, 3 (2011) 155–159.
56. G.S. Lehal, A bilingual Gurmukhi-English OCR based on multiple script identifiers and language models, 4th Int. Work. Multiling. OCR (MOCR '13). (2013) 3.1-3.5. doi:10.1145/2505377.2505381.
57. D.S.B. Patil, Neural Network based Bilingual OCR System: Experiment with English and Kannada Bilingual Documents, Int. J. Comput. Appl. 13 (2011) 6–14. doi:10.5120/1803-2279.

58. R.K. Singh, A. Pandey, A Classification of Handwritten Multilingual Documents, 3 (2014) 1229–1233.
59. R. Kumar, R.K. Sharma, A. Sharma, Recognition of Multi-Stroke Based Online Handwritten Gurmukhi Aksharas, Proc. Natl. Acad. Sci. India Sect. A - Phys. Sci. 85 (2015) 159–168. doi:10.1007/s40010-014-0183-z.
60. A. Sharma, R. Kumar, R.K. Sharma, HMM based Online Handwritten Gurmukhi Character Recognition, Mach. Graph. Vis. Int. J. 19 (2010) 439–449.
61. M.K. Sachan, G.S. Lehal, V.K. Jain, A System for Online Gurmukhi Script Recognition, in: Commun. Comput. Inf. Sci., 2011: pp. 299–300.
62. C. Tappert, C. Suen, T. Wakahara, The State of the Art in Online Handwriting Recognition, IEEE Trans. Pattern Anal. Mach. Intell. 12 (1990) 787–808.

AUTHOR PROFILE

Gurpreet Singh received B.Tech. Degree in Computer Science & Engineering from Punjab Technical University, Jalandhar in 2007 and M.Tech Degree in Computer Science & Engineering from BBSBEC in 2011. He worked as Lecturer in the Department of Computer Science & Engineering, IET, Bhaddal, Punjab from 2007 to 2010. He worked as Assistant Professor in the Department of Computer Science & Engineering, IET, Bhaddal from 2010 to 2014. He was a Research Scholar in CSE Department at SLIET, Longowal from 2014 to 2018. Currently, he is working as an Assistant Professor in Chandigarh University, Punjab. His research areas include Digital Image Processing, Handwriting recognition and Natural Language Processing. He is a lifetime member of Indian Society for Technical Education.

Manoj Sachan received B.Tech. degree in Computer Science from TIET, Patiala; ME degree from Punjabi University Patiala and Ph.D. degree in Computer Science & Engineering from Punjab Technical University, Jalandhar. He worked as Lecturer in the Department of Computer Science & Engineering, TIET, Patiala from 1992 to 1999. Then he worked as Assistant Professor in the Department of Computer Science & Engineering, SLIET, Longowal from 1999 to 2006. Presently he is working as Associate Professor in Computer Science & Engineering department, SLIET, Longowal since 2006. His research areas include Pattern recognition, Neural Networks and Online Gurmukhi Script Recognition. He is a lifetime member of Indian Society for Technical Education (ISTE).