Devanagari Character Segmentation and Recognition

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

Master of Engineering

in

Computer Science and Engineering

Submitted By

Gaurav Pagare

Roll No. 801332008

Under the supervision of

Mr. Karun Verma

Assistant Professor

(CSED)

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

THAPAR UNIVERSITY

PATIALA – 147004

July 2015

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled,**"***Devanagari Character Recognition and Segmentation* **"**,in partial fulfillment of the requirements for the award of degree of Master of Engineering in *Computer Science and Engineering* submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of *Mr. Karun Verma* and refers other researcher's work which are duly listed in the reference section.

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

(Gaurav Pagare)

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

Mr. Karun Verma

Assistant Professor

CSED, Department

Countersigned by

(Dr. Deepak Garg)

Head

Computer Science and Engineering Department

Thapar University

Patiala

(Dr. S. S. Bhatia)

Dean (Academic Affairs)

Thapar University

Patiala

Acknowledgement

First of all, I am extremely thankful to my respective guide ***Mr. Karun Verma*** of Thapar University for his valuable guidance, advice, motivation, encouragement, moral support, sincere effort and positive attitude with which he solved my queries and provide delightful ambiance for learning, exploring and making this thesis possible. It has been a great pleaser and experience to work under his sanctuary.

Dr. Deepak Garg is one of the best teachers that I have had in my life. He sets high standards for his students and he encourages and guides them to meet those standards.

I would like to thank my family members who are dearest and precious to me for their love, encouragement, blessings and support in all respects. Most importantly, none of this would have been possible without the love and patience of my family. My immediate family, to whom this dissertation is dedicated to, has been a constant source of love, concern, support and strength all these years.

IamalsothankfultotheentirefacultyandstaffmembersofComputerScienceandEngineering Departmentfortheirdirect-indirecthelp,cooperation,loveandaffection,whichmademystayatThaparUniversitymemorable.

Date: July, 2015
Place: Thapar University, Patiala (Gaurav Pagare)

Abstract

Machine and human interaction is very essential in today's scenario. This interaction would make search engines, social media, artificial intelligence, cognitive computing more interactive and user friendly. Handwriting recognition is the systematic process of identifying the characters, numbers and symbols present in the handwritten document. In the current work, a recognition model for digitizing handwritten Devanagari characters is proposed. Auto associative recognition technique for Devanagari characters and numerals proposed in the current work by using classifier. To solve recognition problem a dynamic model based on the Hopfield neural network is deployed. The model performs operation in parallel and making it, faster and optimal in solving recognition problem.

Table of Content

List of Figures

Chapter 1

Introduction

1.1 Handwriting Character Recognition

Handwriting recognition is the systematic processes of identifying the handwritten character, numbers and symbol present in the written document. In the recognition focus on, digitize the data with the help of recognition technique. We can apply input from, either by documenting form or by PDAs, Smart phones, tablets. In character recognition, we transform language representation to its symbolic form.

Internet usage is growing very rapidly day-by-day, number of websites are also growing at a very fast pace. In addition, many of the websites now days are in Devanagari script, so there is a need for an application which can recognize Devanagari content and search some keyword within that content [1]. In today's computing environment, there is a need to recognize humans in order process documents. That is why we have an idea to develop the application, which take online handwritten input and perform proper recognition of Devanagari character. This can be done, by either reading in the form of the written form of documents, in a form of speech or by command. The need to communicate with the machine naturally, introduced the term called "Natural Language Processing". Reason to choose the Devanagari is the third most used scripting language. Devanagari is used in India's many regions. Hindi is used by 1.4 billion people and maximum number of users is from India. The research on recognition of Devanagari Character was started in 1977 [2]. Handwriting recognition is a need of human machine interaction, it is the capability of the system to understand the character intelligently and make useful for us. HCR input raw data set is given and we apply a different model to identify the correct character recognition with accuracy. Every day technologies are evolving and need to communicate with the devices that are known by people is the major concern.

In today's computing environment, there is need for a machine to understand the human action either by in the written form of documents, in a form of speech or by command. We want to communicate between the machine and the human and convert the

handwritten data in the form of digital. Every one has the computer, but still many works in the offices, banks, universities still done on the paper and have to convert this into digital by manual process. Handwriting recognition is one of the applications of the "Natural language processing" in which human and machine can act together with each other.

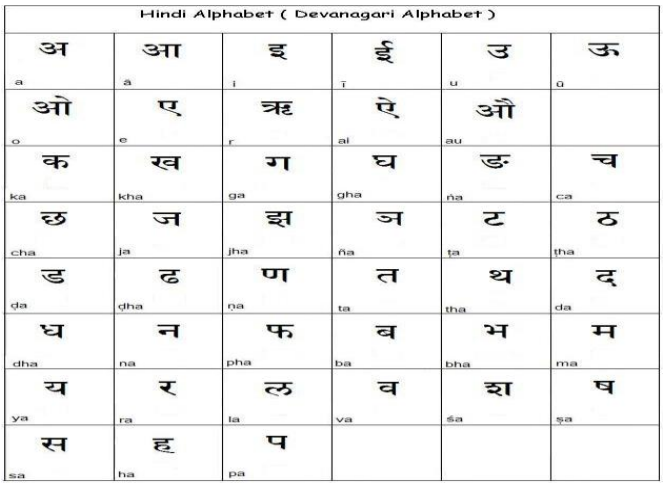

Figure 1.1 Devanagari character script [2].

Handwriting recognition is need of human machine interaction, it is the capability of the system to understand the character intelligently and make useful for us. HCR input raw data set is given and we apply a different model to identify the correct character recognition with accuracy. We need a strong classifier which can recognize the language easily and more precisely, but the problem is the single generic model is not able to fulfill our task because the dependencies of different languages like Gurumukhi, Hindi, Telugu, Marathi, Bangla and many more having the different alphabet set and unlike the shape of the character. So language specific model is needed, for example, in Hindi, English and in many other we start reading from left to right but in Urdu we start reading from the right.

Figure 1.2 Devanagari numerical characters [2].

The intention of this thesis to make an application for recognizing handwritten Devanagari characters. Machine and human interaction are very essential in today's scenario. We need this interaction in search engine, social media, AI, cognitive computing everywhere. Machine and computer together make tasks easy; it is a very emerging research area.

1.2 Handwritten Character Segmentation

Character segmentation is a very important process for recognition; correct segmentation will give optimum recognition results. Segmentation is the processes where we divide the character into the strokes, and these strokes are given as an input to the recognition engine. There is a chicken egg type of problem in recognition and segmentation, which will perform first segmentation or recognition.

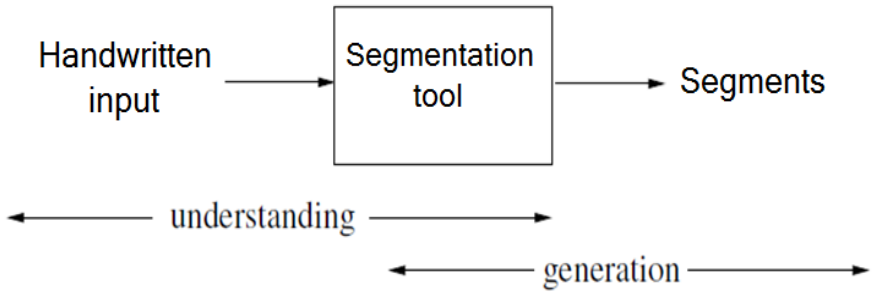

Figure 1.3 Block Diagram of Segmentation

Segmentation is very important wrong segmentation will give the false recognition. Segmentation can be stroke basis, profile basis like horizontal and vertical, shape based or stroke id basis. Shape based segmentation may arises problem in the case of the character of identical style, in Devanagari many characters having similar shape. In this Figure1.3, shows that the input pattern first understands by the segmentation algorithm then generate the segment.

1.3Online and Off-line Recognition

Handwriting recognition is classified in two classes online and offline. Off-line recognition is slightly complex as compared to online recognition, because offline recognition takes a whole image as the input and applies processing on the image in order

to recognize the text, whereas online recognition takes strokes as the input instead of taking the whole image [3]. The main dependency on the handwriting of the user, also depend on the language used for example if we use Hindi so 52 classes and if use English so 26 classes. OCR never works on handwritten data. The result is not much accurate, it further applied to spelling checker to achieve more accuracy. In OCR use of matrix matching for identifying the characters. Intelligent character recognition (ICR) is also a one of the offline character recognition technique. ICR is more accurate as compare to OCR, input is provided by manually by typing or in the form of the printed image. Personal digital assistance, Smartphone's are today's and tablets are basic need, these are the very intelligent devices. They are the online input source for the handwriting recognition with the help of digital ink get the signal and make the character based on the movement of the pen.

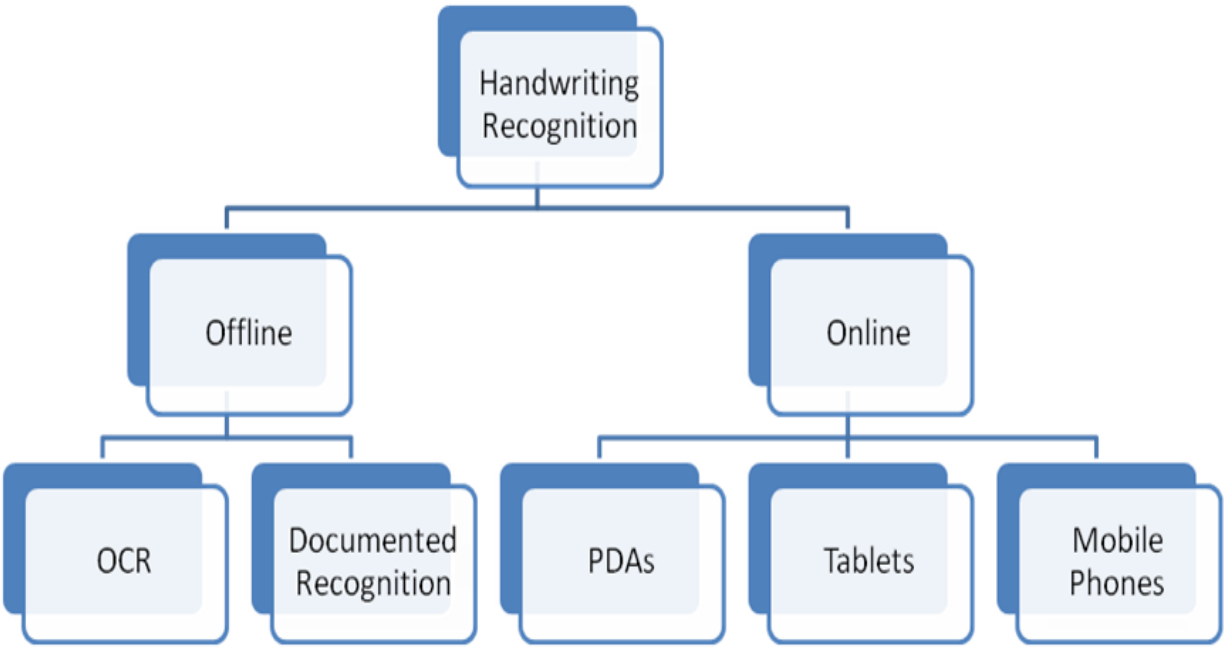

Figure 1.4 Hierarchy of handwriting recognition

Handwriting recognition is having two branches, one is offline and another one is online recognition. In figure 1 we can see the Hierarchy of handwriting recognition.

Handwriting recognition is basically classified in two classes online and offline. Offline recognition is slight complex as compare to online because in offline the main dependency on the handwriting of the user, also depend on the language used for

example if we use Hindi so 52 classes and if use English so 26 classes . OCR never works on handwritten data. It takes print images as input and applying preprocessing on it and give the result. The result is not much accurate, it further applied to spelling checker to achieve more accuracy. In OCR use of matrix matching for identifying the characters.

Intelligent character recognition (ICR) is also a one of the offline character recognition technique. ICR is more accurate as compare to OCR, input is provided by manually by typing or in the form of the printed image.

In figure 1 show the hierarchy of the handwriting recognition the major two module is offline and online and then they also further group like OCR,ICR ,image, document or in printed form, as well in online PDA, tablets, Smartphone other many digital devices those support the writing feature.

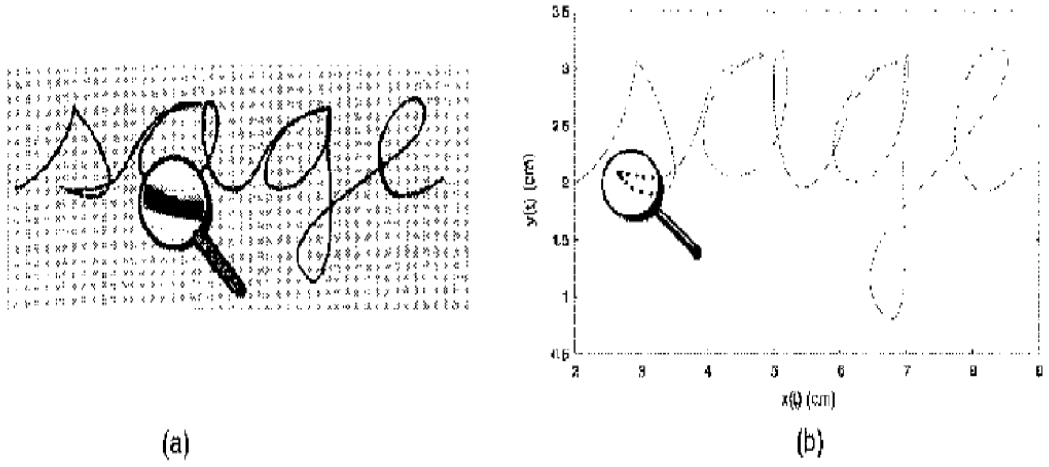

Figure 1.5 (a) Off-line word. The image of the word is converted into pixels using a scanner, (b) On-line word. The x, y coordinates of the pen tip are recorded as a function of time with a digitizer [1].

In recognition, text the problem occurs due to cursive, different writing styles, overlapping characters. Rate of recognition also depends on the segmentation, so we have to also focus on the preprocessing of text. In online coordinates with time, represent the pen movement. In offline less accuracy due to extracting the feature .

In Handwriting Recognition, we have to provide the training to the machine for correct classification. To provide the training there are many learning are used like, supervised

learning, unsupervised learning, reinforcement learning, etc. In machine learning two things are very essential, Hypothesis set and learning algorithm. For learning model both things are required. We provide unknown target function of the machine and on the basis of training examples it will learn. Learning algorithm sets the final hypothesis and decision are tacked by this function.

In supervised learning the machine, learn from labeled data. Input and correct output are already known in this learning. The classification of character is based on this training data and mapping the character according this data. In handwriting recognition, we provide different users handwritten data for training.

Unsupervised is total different from supervised, in these no training data is provided priory. Clustering, k-means are used in this unsupervised learning. Only input is available no labeled data is given.

Reinforcement learning is also used for machine learning reinforcement is encouraged from psychology. Based on previous experience it will tack decision like toddler child. Learning is to finding out the pattern, we give example to memorize.

Choice of learning is also very important. It is depend on the training type, which type of training you provided.

1. Which type of recognition we want single character, whole word or entire line?
2. Learning function precision is dependent on input pattern representation because input object is analysis by feature vectored.
3. Choice of learning algorithm also depends on the structure of learning function like SVM, Decision tree.
4. Validation set, control parameter also required to optimize the performance.
5. For performance measured, there are function or matrices needed, which is separate from the training set.

Personal digital assistance, Smartphone's are today's and tablets are basic need, these are the very intelligent devices . They are the online input source for the handwriting recognition . With the help of digital ink get the signal and make the character on the basis of the movement of the pen. Figure.1.5 shows typical input signals that can be analyzed in both cases online and offline.

1.4Handwritten Input

Input provided for handwriting can be different type like OCR optical character recognition in our supply text in the form of printed for eg. Pass book copy, passport any printed image or typed data.

Online handwriting recognition is dissimilar from offline, in this the input is not given from pen instead of it may be by smart phones, by PDAs or by tablets. In case of offline we get input on paper in the firm of writing but in case of offline we get the coordinate, starting and ending point of pen and movement of pen [4]. Online we have the two dimension pixel available in offline we have complex input.

1.5 Elements of the Handwritten character recognition interface

Some required elements for online HWR consist of:

- The basic element for writing is a pen or a stylus
- A interface like touch screen to take an input and for showing output display.
- A middle ware which make the interface between the stylus and the hardware , convert the input in the digital text.

Handwriting recognition system has some issue because of the following reasons:

- Large symbolic set
- Shape similarity so difficult to identify
- Different writing styles
- Writing speed
- Cursive writing still after separating
- Character complexity

- Poor reliability of extracting stroke features due to variation in handwriting

1.6 Organization of Thesis

This thesis is organized as follows

Chapter 1: Introduction of handwriting recognition and its types.

Chapter 2: Earlier work on the handwritten character recognition is represented in this chapter.

Chapter 3: What are the various problems we have faced in the handwritten character recognition is represented in this chapter.

Chapter 4: The objective and the methodologies used to fulfill these objectives are represented in this chapter.

Chapter 5: This chapter represents implementation and result of the proposed work.

And the last chapter is devoted to the conclusion.

Chapter Summary

Handwritten character recognition is process of taking handwritten character as input and then understanding it. Handwritten character recognition and its uses are discussed in detail in this chapter. For the better handwriting recognition there is need to understand the patterns of script, training models, good segmentation algorithms. As discussed in this chapter handwritten character segmentation is a process of dividing a character into multiple parts and then extracting the feature of that character. After extracting the feature of the handwritten character, the strokes are used for the handwritten character recognition. There are two type of handwriting character recognition that are online and offline handwritten recognition. These types of handwriting recognition are also discussed in this chapter. Also the handwritten input and various elements of the handwritten character segmentation are discussed briefly in this chapter.

Chapter 2
Literature Survey

Handwriting recognition for character classification apply some basic steps like taking the input and apply preprocessing of the data after that feature extraction, matching of the character and at last post processing. For recognition different policy used like Hidden Markov model, Neural network, Genetic algorithms and Fuzzy logic etc.

2.1 Model Design and Training

Many models are made for handwriting recognition, each model itself has some importance. All models have some common steps like first take the input, it can be online or offline after that preprocessing the data, apply some characteristic extraction algorithm train the system and finally classify or we can say matching the character. Many techniques are proposed for Devanagari Character Recognition some of the techniques are:

2.1.1 Template Matching: In template matching technique, the templates are stored in the background or we can say in the backend, the input patterns which are drawn by the user is matched to the stored template with the confidence vale. The best matched template will be selected on the basis of their threshold value. They based matching or matching with a higher similarity will be the result [5].

2.1.2 Statistical Approach: In the method we first extracted the future of the character and divide it into mine-dimension vectors [6]. Each vector class does different not have any common feature. They are mutually exclusive to each other. According to the father of the character it belongs to the vector class.

2.1.3 Structural Approach: In Structural approach, use of the predefined rule is used. Based on the features of the character classification is done. In this first we need to train our system based on our rules and on the basis of this it will give the final output [5].

2.1.4 Genetic Algorithm: With the help of genetic algorithm, we can easily classify the character with more accuracy. In this technique for each character, we make a

graph, with the help of graph understanding of the character easily. In this process single input character is applied and make graphs of each character. GA processes crossbreed graph after that applied the fitness function. The fitness function chooses best matching character. This will give the optimal solution. Its accuracy is approximately 98.43% [6].

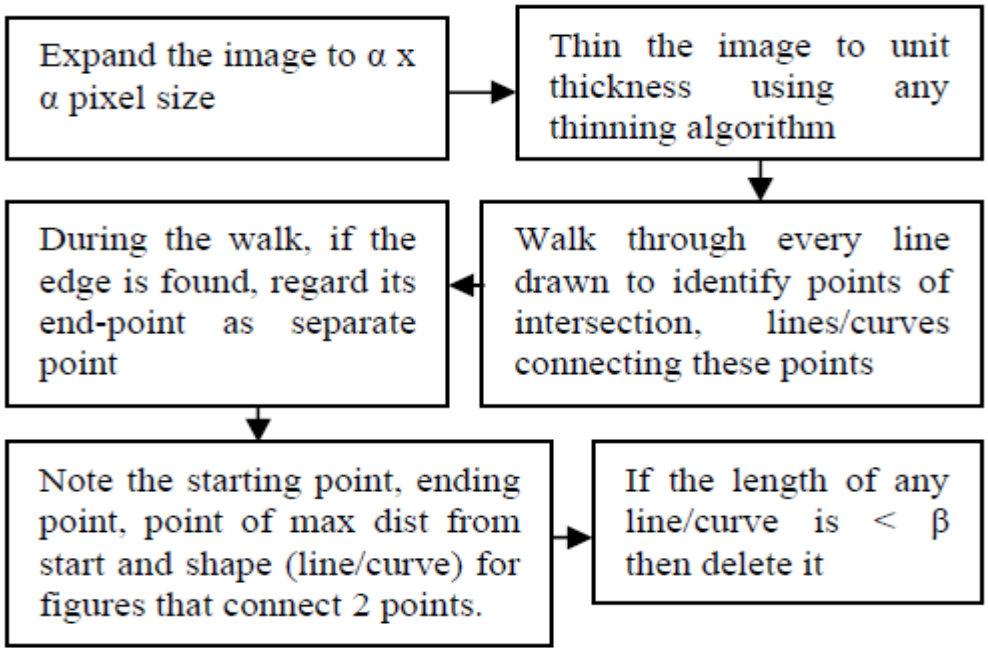

Figure2.1 the Graph Generation Algorithm [4]

With the help of Genetic algorithm we can easily classify the character with more accuracy. In this technique for each character we make a graph, with the help of graph understanding of the character easily. In this process single input character is applied and make graphs of each character. GA processes crossbreed graph after that applied the fitness function. The fitness function chooses best matching character. This will give the optimal solution. Its accuracy is approximately 98.43% [7].

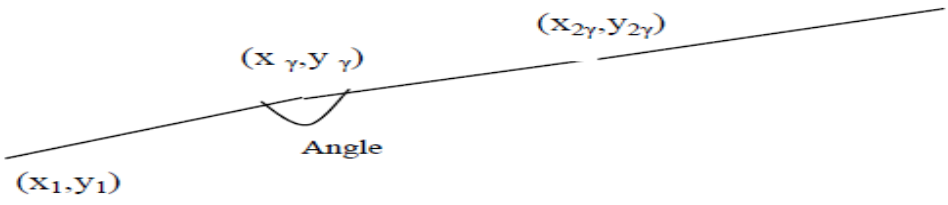

Figure 2.2 Differences between curve and line [7].

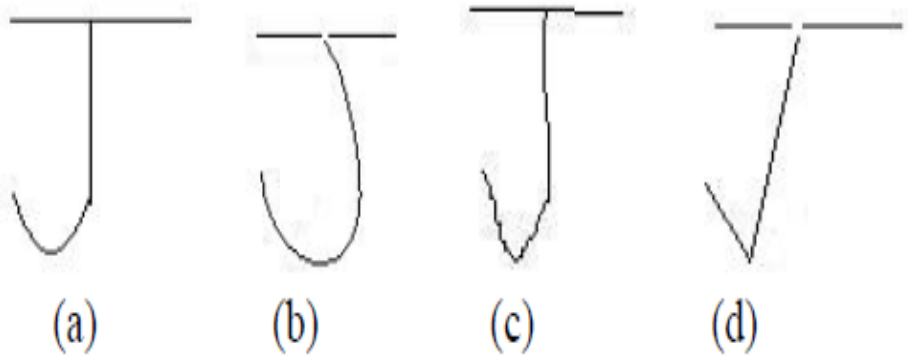

Figure 2.3 Two Ways J was written (a, c) and the generated graph (b, d) [4]

2.1.5 Allograph Modeling for Online Handwritten In the online handwriting user drawn by the stylus, the order of stroke and number of strokes both is playing very important role in writer dependent system. In the other technique clustering of stroke set are supervised to minimize this dependency allograph technique is used. Allograph will give a high recognition rate in the writer independent system. It is based on constrained stroke clustering to find minimum stroke cluster set [9].

2.1.6 Convolutional Neural Networks In this Convolutional Neural Networks approach tries to make the system more robust in terms of number of strokes, certain, characters in any order. In this approach they used 10 different CNN configuration. For convergence, two techniques are used exponential decay and inverse scale approach, the accuracy on this proposed approach is 99.8 % of trained data set [10].

2.1.7 Hidden Markov: This model is basically based on the probability distribution in which the stochastic process is playing a major role. Change of state produce the output, the output symbol is also in two categories if finite alphabet so it is a discrete otherwise continuous (real valve) so on this basis we can classify the HMM can be discrete or continuous [7]. The average recognition of Takahashi, K *et. at.*was90.0% and the best recognition rate was 98.0% (mdb0005) [8]. The hidden Markov model is basically based on the probability distribution in which the stochastic process is playing a major role. Change of state produces the output, the output symbol is also in two categories , if finite alphabet so it is a discrete otherwise continuous (real valve) so on this basis we can classify the HMM can be discrete or continuous.[5].

HMM Feature - Many HMM features are used to classify different language like Bangla, Telugu they are having large set 11 vowels and 40 consonants, 18 vowels and 36 consonants respectively.

Some feature of HMM used in classifying different models-

- Stroke based when pen start to write and down record this series of co-ordinates
- Look up table maintain for identification the unidentified character
- Pen trajectory is recorded
- Time domain
- Probability for each symbol
- Frequency domain

2.1.8 Generalized Hausdorff Image Comparison In this paper present the Hausdorff image comparison for the OCR recognition system. In this approach, first identify the script then character segmentation and finally recognition of character. This approach has identified the different font and script from bilingual or multilingual document. Identification of script is word level and for improve the performance SVM is used.The average accuracy of this approach is 87.82% at character level using Hausdorff image comparison. [11]

2.1.9 Nebulous Stroke Models It is a shape-based method, which is used for segmentation of the character. In every script, there are many different types of characters shape are present. Although they are different character, but some common shape may be present [11]. On the basis of these similar shapes nebulous stroke model recognizes these characters.

There is problem with these approaches is that it segment the character before the recognition. In this the single letter is made by the combination of different strokes and this limits the number of segmentation of a single character.

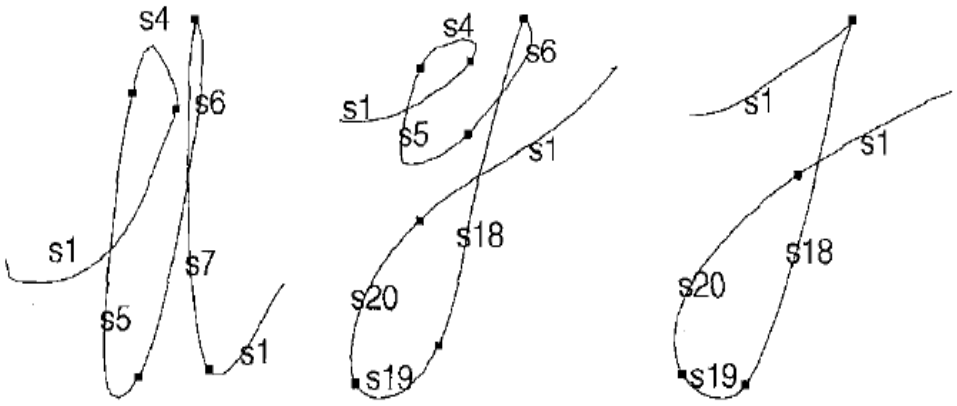

Figure 2.4

Stroke segmentation of three letter samples [12]

2.1.10 Artificial Neural Network: In the ANN the many model are used as the interconnection network. ANN is basically used for training the pattern and according to the model we choose it will work [1].

The average accuracy when we use this model is around 75% [8]. In feature extraction process the coming frame sequence which comes from pen trajectory his frame is captured the 6 coordinate. In Figure 17 in this first we train the system on 5 multilayer system in which feature extraction on the basis of threshold value it will give the classification result.

2.2 Steps of handwriting recognition

2.2.1 **Preprocessing** Preprocessing as name suggested that it is the first step which applied on input data. It removes the redundant variation, perform rotation on input if needed. Some steps like [12] scaling up, scaling down, smoothing, also remove the noise if present in the data, normalize the data. Figure 3 also show the some step which performing in the preprocessing. Duplicate point remove, elimination of hooks, smoothing then normalization.

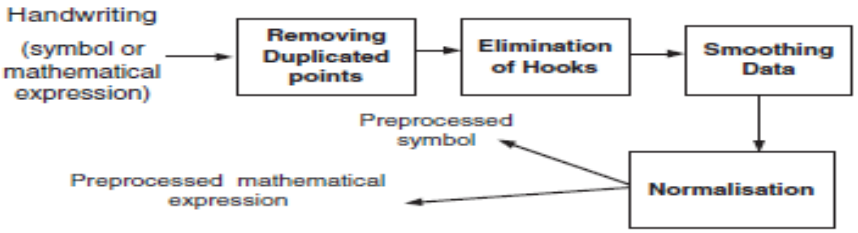

Figure 2.5 Preprocessing steps [2]

2.2.2 Removing duplicated Points

In duplicate point removing method we search for two or more point having the identical coordinate, if it is found so one of them is to be removed. This is the basic first step of preprocessing.

2.2.3 Elimination of Hooks

Elimination of hooks is needed for making quality image. Hooks is commonly present at the starting and end of the character. Hooks are present due to angle variation in the writing the character. In this given example the character w having the sharp turn so due to this pointed turn the hooks are coming we have to remove this hooks.

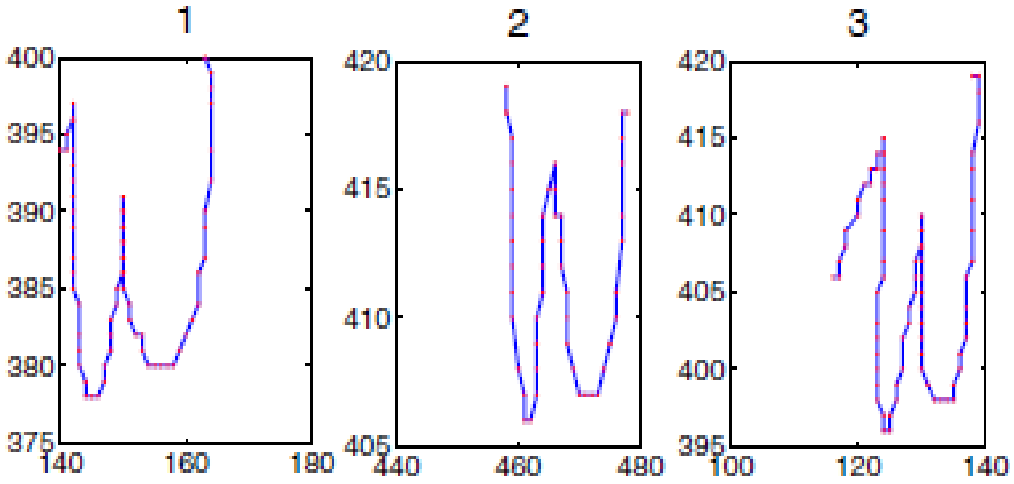

Figure 2.6 the raw character w [2].

2.2.4 Interpolating points

When writer wrote the image with either online or offline it may possible there is some missing point in the drawing. We add the point linking two successive points if it is missing. Suppose two point X1 and X2 and in between this point is missing so we have to add some point.

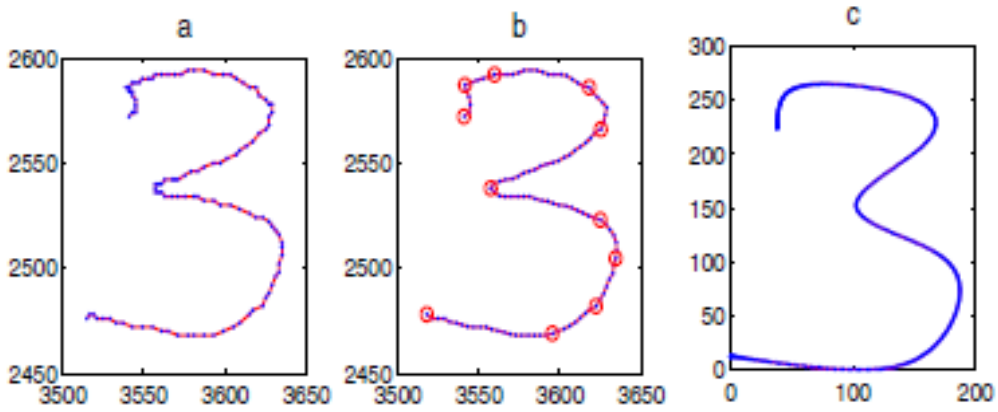

Figure 2.3 preprocessing. (a) Raw data, (b) the data with the control points after removing the hooks and duplicated points, and (c) the normalized data. The control points are marked by the red circles. The points are represented by blue dots [13].

2.2.5 Normalization and Smoothing Data

Basically normalization is performed on image to trim down the size to fit in the square box. In figure 4 we can see the variability of character W so we have to remove this and convert to the smooth curve. In figure 6 the smoothing is done.

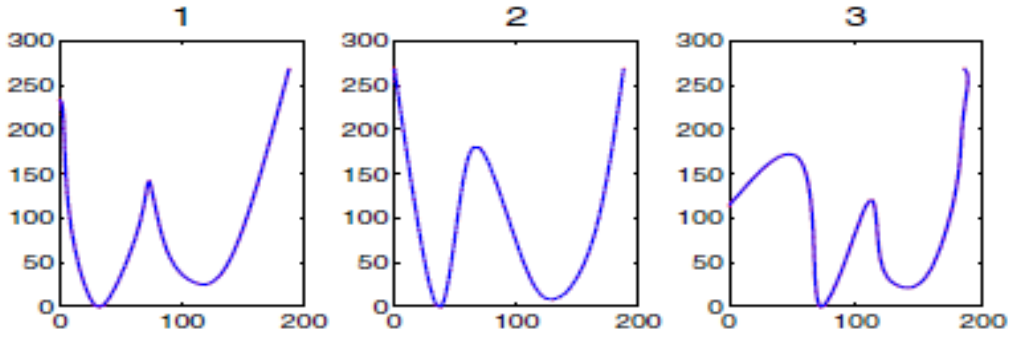

Figure 2.7 the preprocessed character *w* [2].

2.3 Feature extraction

In feature extraction we identify the characters on the basis of its feature to make better recognition system. Feature extraction is not effortless task because the variation in character, ambiguity in the handwriting.

2.3.1 Structural

Structural feature extraction used the information of the character on the basis of loops and the division of the character. Crossing point in the character, ending point and the statistically these are the feature.

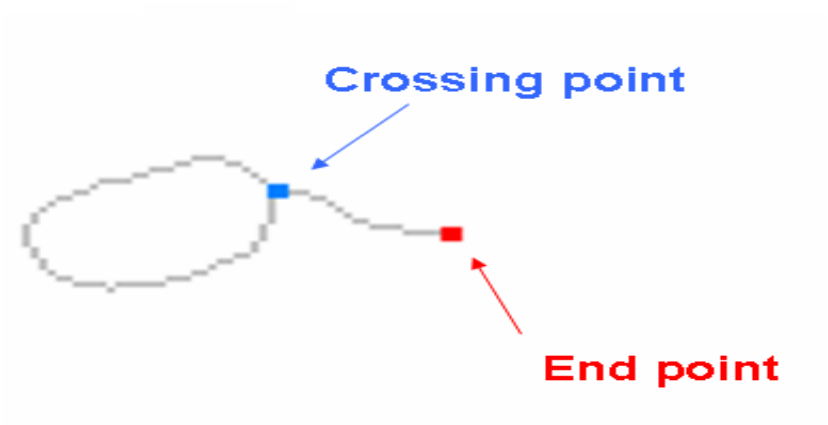

Figure 2.8 crossing point and end point [3]

Loop, Endpoint: Where the pen down mean stop to writing is the ending point and loop is a union of inner A and the end point.

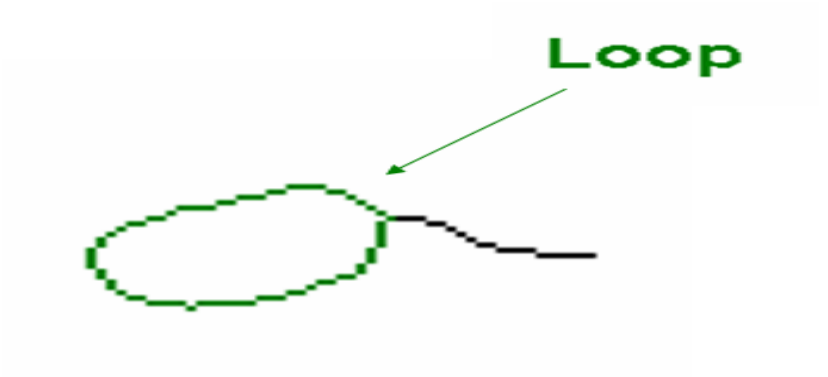

Figure 2.9 loop [3]

2.3.2 Statistical

Statically feature is basically based on the variation in the style of the character, used techniques are-

Zoning**:** Feature vectors are used in this technique and we portioning the image in AxB part so we can easily achieve the all over characteristic of the image instead of overall.

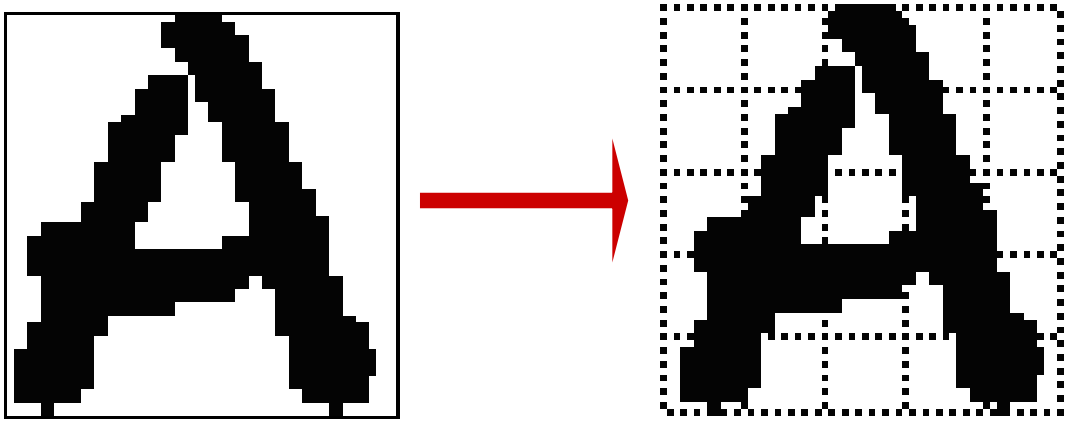

Figure 2.10 also considering the pixel in the forefront and in the background as a feature of the image [3].

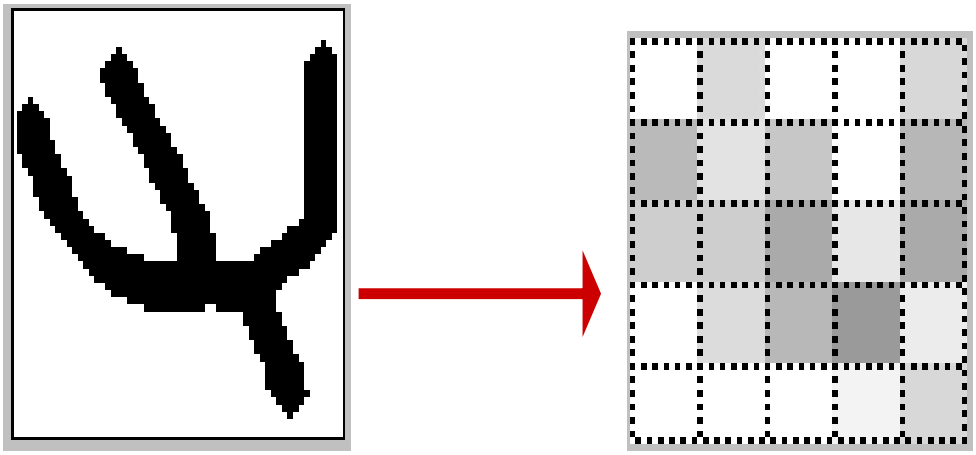

Figure 2.11 Darker squares indicate higher density of zone pixels projections and profiles [3].

Crossings and distances In this technique we calculate the distance of the boundaries upper site, lower site, leftmost and rightmost site and crossing refers the number of moves from backdrop to forefront pixel

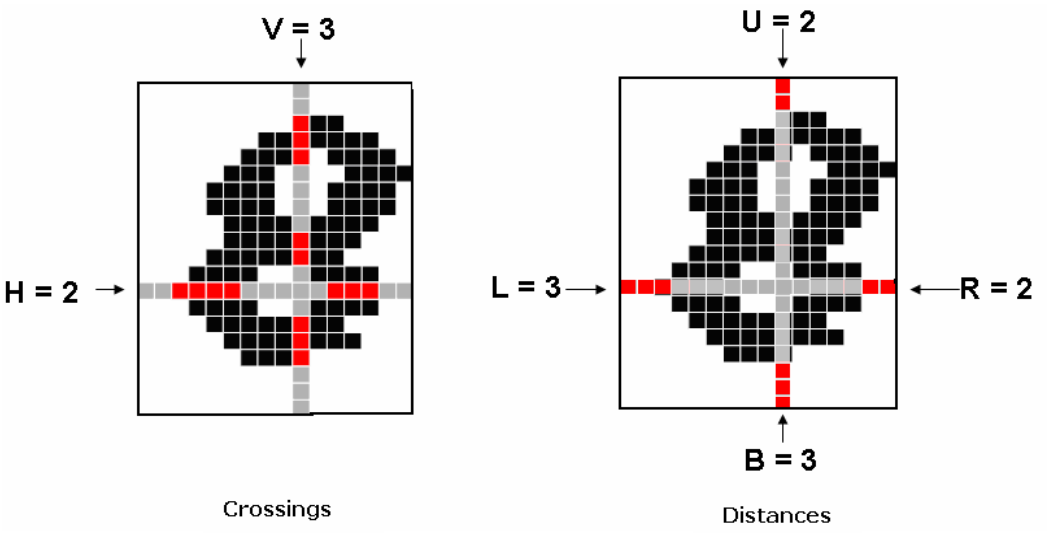

Figure 2.12 crossing distance [3]

2.4 Post processing

- Character joining technique
- The action associated with the character
- Dictionary algorithms

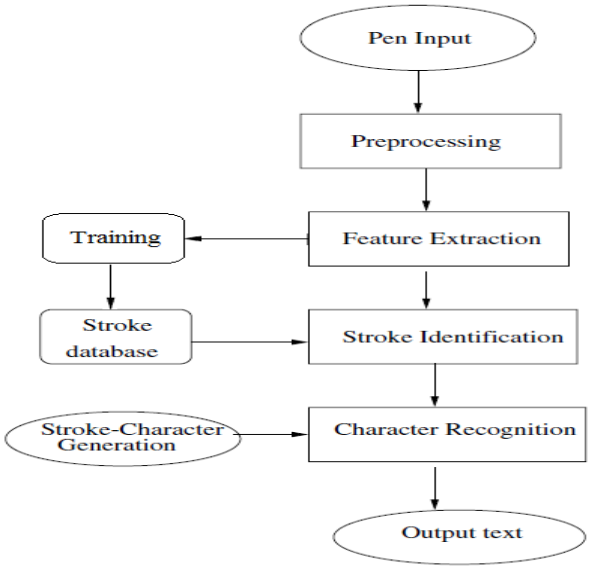

Figure 2.13 Model Design and Training

In the post processing step we try to optimize the result by appling some technique on the recognition result. The spelling checker used to increase the efficiency of the result. Some other method also applied character joining technique to make association between the words to make word recognition strong [14].The diagram 2.13 shows the all the steps from initial to final output taking the input to output text receiving.

Chapter 3

Problem Statement

The intention of this project is to make an application to recognize handwritten Devanagari characters. Machine and human interaction are very essential in today's scenario. We need this interaction in search engine, social media, AI, cognitive computing everywhere.Machine and computer together make tasks easy; it is a very emerging research area.

- Many language recognition tools are available in current scenario, but in Devanagari, very little work is done.
- In this project offline and online both types of input we can apply either an image or with the help of a mouse we can draw the character.
- The Java based project used Hopfield neural network for the training data.
- In this project we can provide training, also from input data at the time of drawing the character.
- When we apply a distorted pattern as input the recognition accuracy may decrease to improve this here uses Hopfield.

Developing the Devanagari Handwritten character recognition and segmentation system has some greater challenge because of the following reasons:

- **The character set is very large**

  In Devanagari the vowel, consonants, matra, Chandra Bindu, Visarg and many more different symbols are present. To make a system which recognizes all of these complicated shapes is a difficult task.
- **Similarity between the character is high**

  Many similarities in shapes so it is difficult to segment the character and recognize the desired result. This ambiguity arises a problem in the segmentation of Devanagari character.

Chapter 4

Objective and Methodology

In today's computing environment there is need to machine to understand the humans thought either by in the form of the written form of documents, in a form of speech or by command. We want to communicate with the machine Naturally so there is term introduce called "Natural Language Processing. Human For this there is a need of an application which recognize the character.

The intention of this project to make an application for recognizing handwritten Devanagari characters. Machine and human interaction are very essential in today's scenario. We need this interaction in search engine, social media, AI, cognitive computing everywhere. Machine and computer together make tasks easy; it is a very emerging research area.

4.1 Objectives

Communication with the machine can be done in many ways and one of them is handwriting. With the help of handwritten input we can give input to the computer. Reason to choose the Devanagari is, it is on the third of the most spoken languages. Devanagari is used in India's many regions. Hindi is used by 1.4 billion people and maximum number of users is from India. Every day technologies are evolving and need to communicate with the devices that are known by people is the major concern.

Internet usage is growing at very rapidly the number of web sites grows increases day by day. Many of the websites are based on Devanagari, so need to be make an application which recognizes the Devanagari content and search on the basis this content. That's why I have an idea to develop the application which take online handwritten input and perform proper recognition of Devanagari character.

4.2 Methodology

The network is to type according to recalling the principle static and dynamic, auto associative Hopfield is dynamic in nature. Theoretically in static network there is no delay in output these are called feedforward network, input is given and instantly output

is coming $Y_k=M_1$ [X^k] This equation shows mapping of input vector X to output vector Y, M is an operator symbol and K shows the index of recursion [9]. Dynamic network is recursive in nature and use the feedback mechanism $Y^{k+1}=M_2$ [X^k,Y^k] Preset instant k of the present input X^k and output Y^k to produce the output in the next instant k+1.

Associatively can be hetero or homo it depends on the requirements, if we used for retrieval of missing pair then use hetero. In hetero two vector pair is used input is different and corresponding retrieval is different like milk is associated with bread $X^{(me)} \neq Y^{(me)}$ i=1,2, ,….p. In case of auto associatively $X^{(i)}=Y^{(i)}$ i=1,2,….p the practical application of auto-associative is recovering undistorted pattern in response to distorted pattern [9]. After writing the character on the screen we have to bring the characters in a standard size because the writer may use a small portion of writing areas. Or may be whole. We want the recognition of the character is size independent. For standard size, we have to define some specific that how many pixels are considered as one pixel [15].

Step 1: Weight matrix W is calculated by "equation 1".

52 characters and 10 numbers are bipolar binary vector stored pattern. We want to store sets of states P^m, m=1, 2,…, n

$$W = \sum_{m=1}^{n} P^{(m)} P^{(m)t} - nI \qquad (1)$$

$$W_{ij} = \left(1 - \delta_{ij}\right) \sum_{m=1}^{p} P_i^m P_j^m \qquad (2)$$

Step 2: δ_{ij}=1 when i=j and δ_{ij}=0 if i≠jδ_{ij} is a usual kroKroneckernction, no need to remember individual vector $P^{(m)}$ only remember the W_{ij} weight [9].

Step 3: Increment m=m+1 if m is less than the no of pattern and calculates weight.

Chapter 5

Implementation

The Hopfield artificial neural network is a recurrent neural network. It is proposed by John Hopfield in 1982 which is based on auto associative memory. Hop field is a dynamic system in which we have seen the movement of input pattern towards the stored output pattern [9]. Unlike the other static neural network, Hopfield is a dynamic and having a memory to store the patterns.

Associative memory can store large number of patterns and capable to memorize them.After storage we recall the pattern with the help of search argument, this key pattern contains the portion of information about the particular pattern set. In associative neural network, learning requires a priori information. The weight matrix is also formed in advance.

Advantages of this model is-

- Large pattern can store in an associative memory model.
- Local damage can't affect the recalling of pattern, it provides robust storage.
- It provides the ability to add and eliminate association
- Parallel search performs in stored data.

5.1 Basic Concepts

The network is to type according to recalling the principle static and dynamic, auto associative Hopfield is dynamic in nature. Theoretically in static network there is no delay in output these are called feedforward network, input is given and instantly output is coming $Y_k=M_1[X^k]$ This equation shows mapping of input vector X to output vector Y, M is a operator symbol and K shows the index of recursion [9]. Dynamic network is recursive in nature and use the feedback mechanism $Y^{k+1}=M_2\ [X^k,Y^k]$ Preset instant k of the present input X^k and output Y^k to produce the output in the next instant k+1.

Associatively can be hetero or homo it depends on the requirements, if we used for

retrieval of missing pair then use hetero. In hetero two vector pair is used input is different and corresponding retrieval is different like milk is associated with bread $X^{(i)} \neq Y^{(i)}$ i=1,2, ,….p. In case of auto associatively $X^{(i)}=Y^{(i)}$ i=1,2,….p the practical application of auto-associative is recovering undistorted pattern in response to distorted pattern [9].

5.2 Algorithm

After writing the character on the screen we have to bring the characters in a standard size because writer may use small portion of writing area.or may be whole. We want the recognition of character is size independent. For standard size we have to define some specific that how many pixels are consider as one pixel [16].

- horizontal_ratio = (Width of character image area) / (Width of down sampled image)
- vertical_ratio = (Height of character image area) / (Height of down sampled image)
- binary_metrix_conversion(x, y) = black, if there is a black pixel in the area starting from (x*horizontal_ratio,y*vertical_ratio) to (x*horizontal_ratio+horizontal_ratio, y*vertical_ratio+vertical_ratio); white otherwise[3].

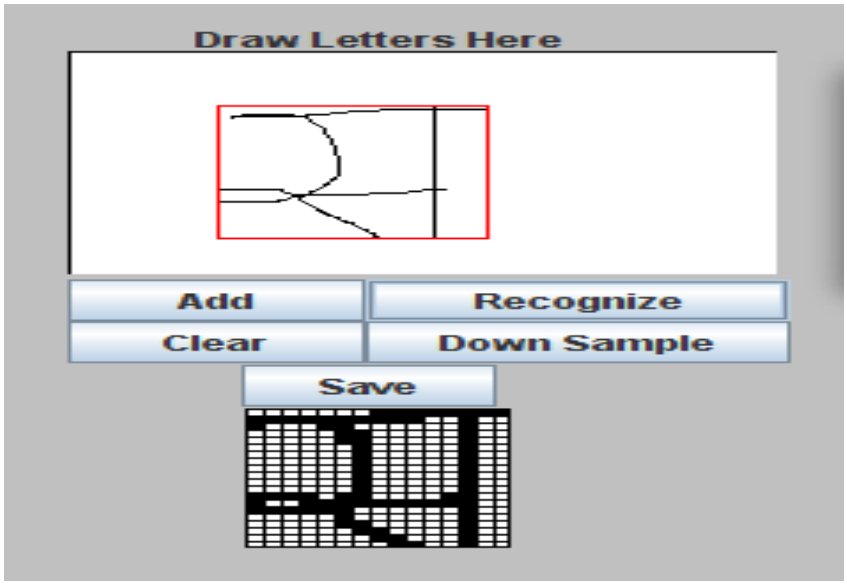

Figure 5.1 Representation of down sample image of Devanagari character.

5.3 Training of Network

Step 1: Weight matrix W is calculated by "equation 1".

52 characters and 10 numbers are bipolar binary vector stored pattern. We want to store sets of states P^m, m=1,2, …,n

$$W = \sum_{m=1}^{n} P^{(m)} P^{(m)t} - nI \quad (1)$$

$$W_{ij} = \left(1 - \delta_{ij}\right) \sum_{m=1}^{p} P_i^m P_j^m \quad (2)$$

Step 2: δ_{ij}=1 when i=j and δ_{ij}=0 if i≠jδ_{ij} is a usual kronecker function, no need to remember individual vector $P^{(m)}$ only remember the W_{ij} weight [9].

Step 3: Increment m=m+1 if m is less than the no of pattern and calculates weight.

5.4 Update a node

The use of Hopfield network is to store patterns in the memory and recall the correct pattern when distorted, partial correct or correct input is given. The computation is done in this model by iteration of the input pattern, which is given to nodes of the network. Each pixel is behaving like a node in the network. We train the network by 52 characters and 10 digits of Devanagari script and assign some weight to it [10].

$$Y_i^{k+1} = \mathrm{sgn}\left(\sum_{j=1}^{n} W_{ij} Y_i^k\right) \quad (3)$$

Here k represents the index of the iteration and the I represent the number of neurons which is updated.

In the below diagram shows the whole procedure in the form of flow chart training, updating weight, calculation, update weight etc.

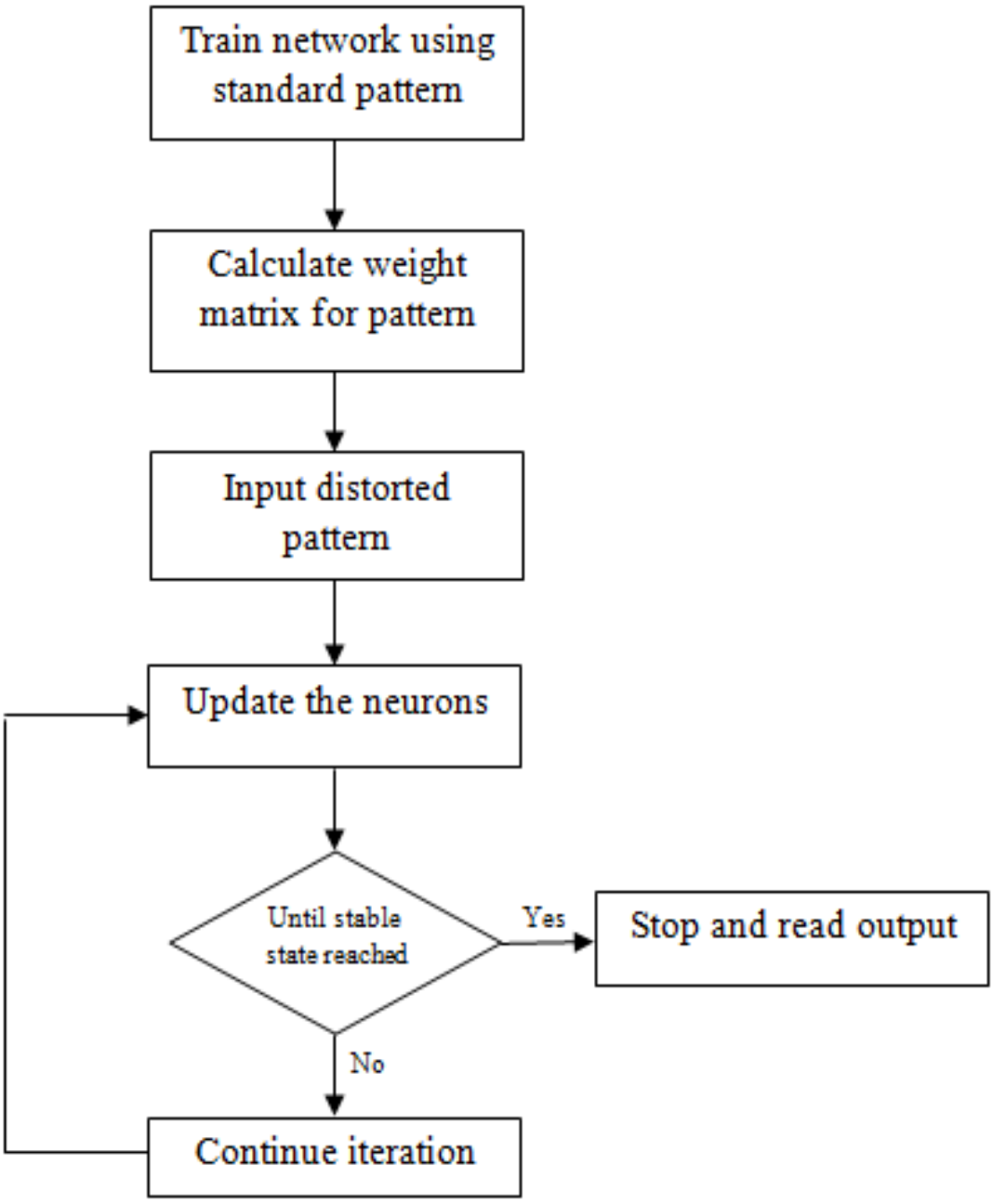

Fig. 5.2. Flow chart of Hopfield Network, it shows how recognition processes works.

In this model we have a weight matrix for 52 characters that's meant we can recognize these patterns. First, we calculate the weighted sum of the input to the other nodes, if this calculated value is positive, then output is 1 otherwise, output is 0.

Input to the neuron is changing and we have to update the node's value it is very tricky. Hopfield is working on both updating mode asynchronous and synchronous [17].

In asynchronous mode, only one neuron is updated at a time or changes a state, and then delays Δ produced in the output, but in case of synchronous mode all the neuron change the state at the same time.

We go through all the nodes in the network and none of them is changed then we stop the iteration. And on the basis of the final value of the nodes we get the desired result.

5.5 Energy Function

Hopfield network defines an energy function for each node for evaluating memory performance. The energy is positive when the W and Y are different in sign. The purpose of this energy function is to get stable state, and we know stable state is having low energy. We focus on that the energy of Hopfield either is decreased or remain same. Update of neurons cause change in sign then definitely energy will decrease otherwise same [18].

$$E(Y) = -\left(\frac{1}{2}\right) Y^{t} W\, Y$$

The energy function plays very important role for the stability of neuron in the network. When initial input is given the energy valve is zero, at the end of the convergence energy value decrease and it is somewhere around -5400 for a single character.

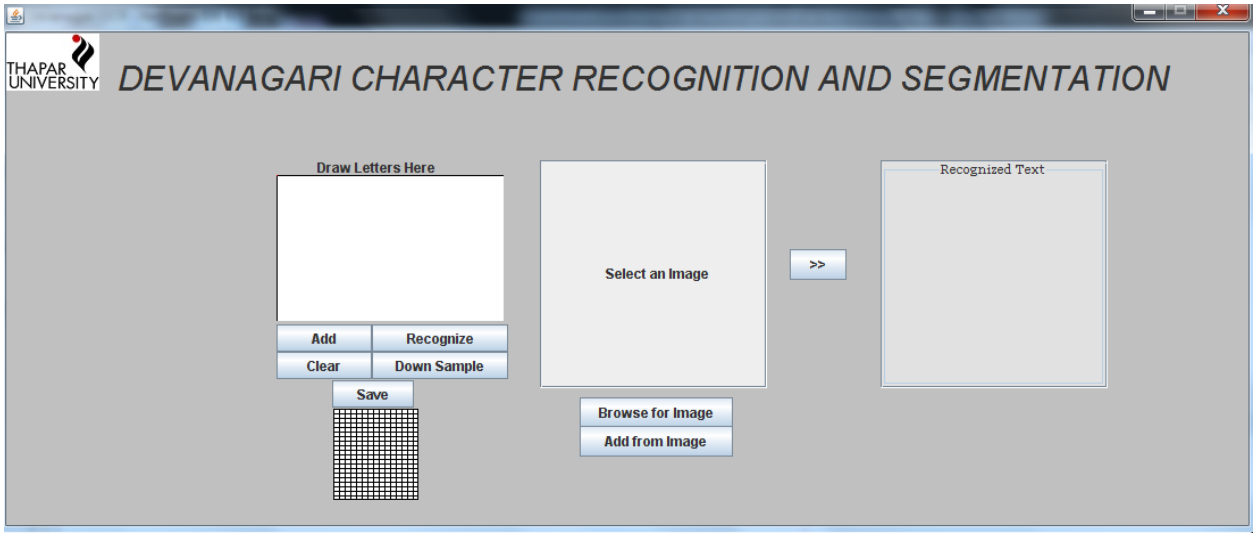

Figure 5.3 Components of the handwritten character recognition &segmentation tool

GUI of the project- Here the Draw letter, Letters known, Recognize, Training all the options are available.

In the above diagram, there is a writing area where the user writes the character. There is also select image area foe selecting the scan image from system, letters known area having the letters which are given in the time of training. Load the Train Character from sample.dat file. Click to Load Button and it will automatically load the trained character.

Recognition of the character- With the help of mouse Draw the character on the screen and click to the recognize button it will give you the Recognition Successfully message with the character.

The character you added above is already coming to the Letter, Known and if you draw again this added character it will recognize it properly.

The character drawn and it is successfully recognized shown in the figure 5.2..

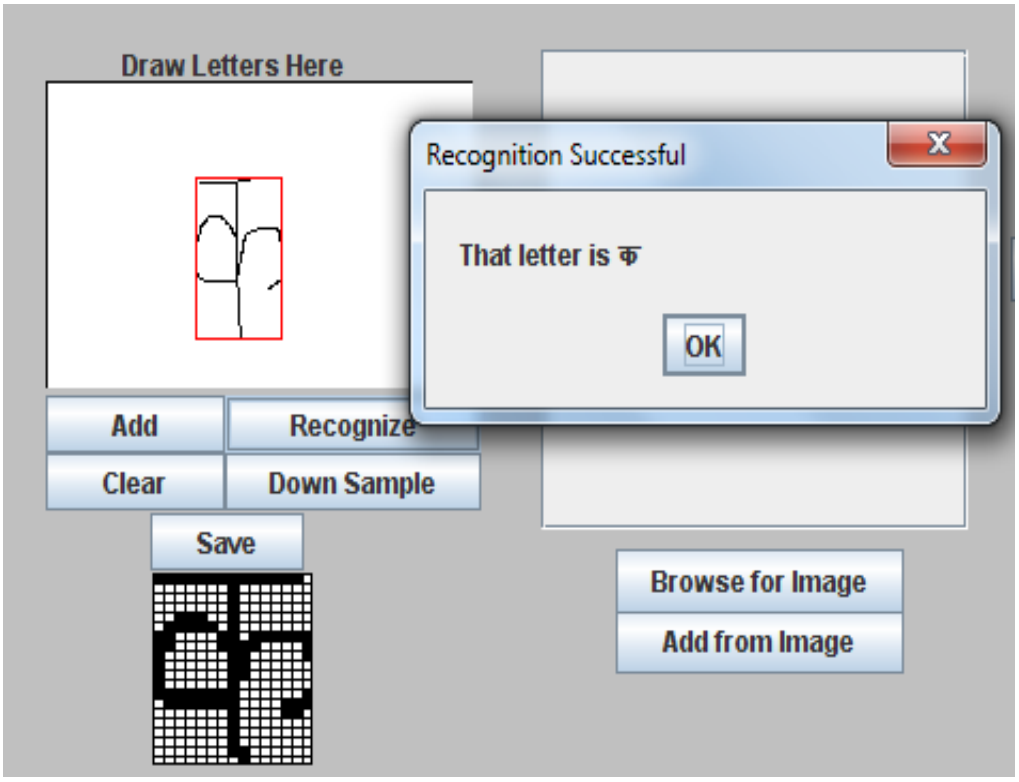

Figure 5.4 Draw letter and corresponding recognition.

It can tack number as a input, in figure 5.4 shows the correct recognition of numerical value 7 in Devanagari.

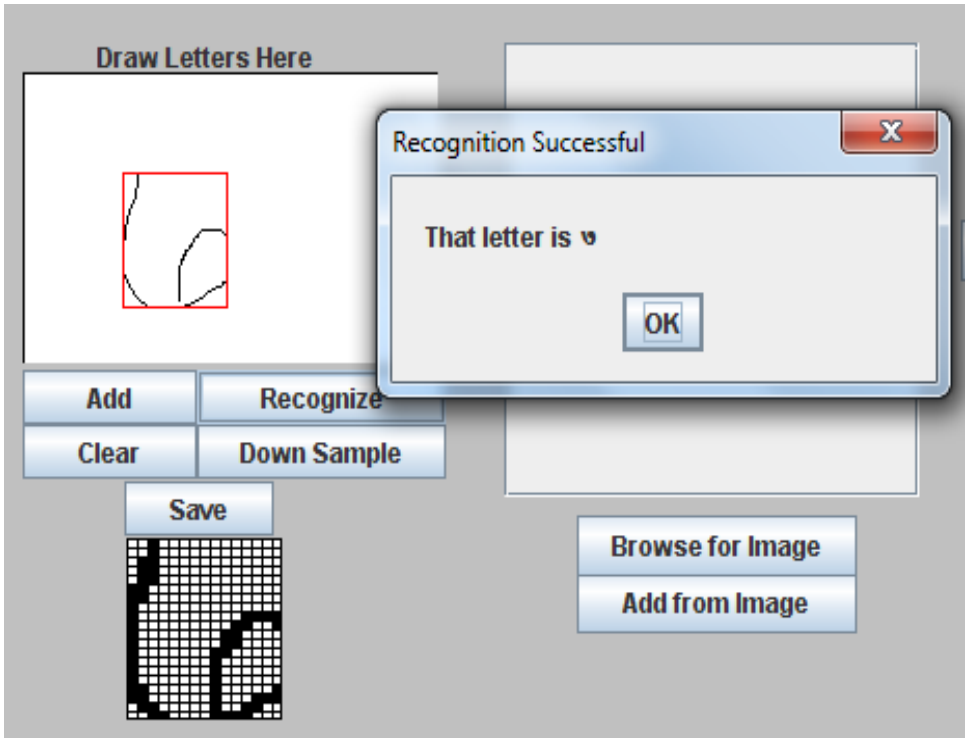

Figure 5.5 Devanagari numeric recognition.

For recognition of Devanagari character recognition and segmentation, we used a simple Hopfield neural network used. It is the one of the simplest neural network. There is no hidden layer present in this network.

Each input neuron is associated some weight and connected to the output neuron. Each weight is different and on the basis of these weights the output neuron is selected.

When we select an image for recognize the text there are three steps performed. Image is having number of lines, characters, and words. After browsing the image

- First, we find the number of lines in the whole image
- Then lines are segmented into the words
- After word segmentation, the individual word is segmented into the character.
- For making the character standard size need the normalization.

This standard size image is converted into the matrix of down sample and given as input to the Hopfield model.

5.6 Segmentation

Find Line- In segmentation of the image we used horizontal projection profile and vertical projection profile to segment the image. Horizontal projection profile shows

number of black pixel in the line and vertical projection show number of black pixels in vertical column. Vector of horizontalProjection[k] is used for store the black pixel till line not end if value of hp[k] is zero mean white pixel is found and perform segmentation. Integer variable imageHeight is used for calculating the height of image [19].

Algorithm 1. Find Lines in the input image.

Input : int pixels [], int imageWidth, int imageHeight.

Output: returnLines.

1. Vector lines;
2. Int array hp[] ; (Horizontal projection array used for segmentation)
3. **For** each i =0 to imageWidth * imageHeight;
    a. red value= pixel rightshift by 16 ending with ff;
    b. green value= pixel rightshift by 8 ending with ff;
    c. blue value= pixel ending with ff ;
4. calculate y= sumof(r,g,b);
5. **if** y greater than 150 , white ;
6. **if** (i%iw==0) then set hp[i/iw]=0;
7. **if** !white set hp[i/iw]++;
8. **for** i=0 to imageHight(ih);
    a. **if** hp[i] != 0, line start;

        count lines;

    b. **hp[i]**==0, nothing will do continue iteration;

The threshold value is defined for pixel (mean black) and no pixel mean (white). On the basis of this the matrices of 0 and 1 (white 1 and black 0) is formed.

Here we calculate the red, green , blue (RGB) values with the help of ending with 8 bit 1's and left shift by 8 and 16 bit.

This function is used for horizontal and vertical projection profile. When there is no pixel is available in vertical portion it will segment the character assuming that it is the starting of other character.

In above image integer ih represent the image height and hp is vector which store the segmentation information of lines. The value of hp[array] is zero means there is end of line and need to be segmented.

Find Words - Vector of verticalProjection[k] is used for store the black pixel of vertical column value of vp[k] is zero mean white pixel is found and perform segmentation of word. Integer variable imageWidth is used for calculating the width of image.

Algorithm 2. Find words in each line.

Input : int pixels [], int imageWidth, int imageHeight.

Output: Words.

1. **FindWords** (p,iw,ih);
2. **Create** vector of vp[];
3. **For** i to imageWidth;
    a. **Set** vp[i]= 0; (Vertical projection array used for segmentation)
4. **For** each i =0 to imageWidth * imageHeight;
    a. red value= pixel rightshift by 16 ending with ff;
    b. green value= pixel rightshift by 8 ending with ff;
    c. blue value= pixel ending with ff ;
    d. calculate y= sumof(r,g,b)/3;
    e.
5. **if** y = threshold greater than 150 , white= true;
6. **if** (!white) then vp[i%iw] continue;
7. **for** 0 to imageHight(ih);
    a. **if** vp[i] != 0;
    b. **count** words;
    c. **while** vp[i]&&iw !=0;
    d. **increment** i word start;

e. **while** vp[i]&&iw ==0;

f. word end;

return word;

Find Character – For finding the individual character we have to remove the header line (s*hirorekha*) [20]. For calculating header finds the maximum value of black pixel in the row. After identifying the header line it is very easy to get the character.

Algorithm 3. Find character within each line.

Input : int pixels [], int imageWidth, int line_end;.

Output: character;.

1. **FindCharacter** (p,iw,le);
2. **Count** set zero;
3. Vp[]= wordend-wordstart+1;
4. **For** i to wordlength;
    a. **Set** vp[i]=0;
5. **for** i to iw*lineEnd + wprdEnd;
    a. red value= pixel rightshift by 16 ending with ff;
    b. green value= pixel rightshift by 8 ending with ff;
    c. blue value= pixel ending with ff ;
    d. calculate y= sumof(r,g,b)/3;
    e. white=false;
6. **if** y > threshold , white= true;
7. **if** (!white) then vp[i%iw-wordStart] continue;
8. **if** (i=wordEnd)%iw ==0;
    a. i + = iw+wordLength;
    b. add character;

character add to temp;

To evaluate the working of the proposed method, it is applied on the distorted Devanagari characters and numerals. Input pattern, which are applied in the network, are 10 to 50 percent distorted. Based on initial pattern the convergence of the network depends, if an initial pattern is more distorted it will take more iteration to converge. In these figures, we show the down sample images of the character and numeric.

Chapter 6

Conclusion and Future Scope

AI and Machine learning is very popular in today's scenario, many companies are making the robots and to communicate with this we need to develop "Natural Language Processing " so that the machine can understand the language in which we are communicating, understand what we are writing that is "Handwriting Recognition" . In future the need of handwriting recognition is very growing. We make many applications like action associated with this to give control to anything, dictionary based application which first recognize the character and the predict the word related to that is also very popular in today's.

For accurate character recognition, choosing of correct model is very essential, otherwise it leads to false recognition. Many models are proposed for handwriting recognition, but many of these experience problems when distorted or noisy input is given. Shape based recognition model is also not efficient because of similarity in characters. That is why in this paper use of Hopfield because it works well for distorted. The average accuracy of the characters is 92.91% and the best recognition rate is 96.29%. To achieve more accuracy hybrid model is suggestible like can use Hopfield with CNN (Convolutional neural network). In future, try to implement this hybrid technique for recognition of the whole character.

In this Devanagari Character Recognition the accuracy rate is not high and this is only performed recognition of a single character. In case of slant character as input the performance is degraded. This application is basically giving output on the basis of trained character which are given to the Hopfield. The Hopfield iterates the neuron and update the final value based on this entire pattern is recognized [22]. Handwriting character recognition and segmentation is based on Java and used the netbeans IDE. A user can give the image as well single character as input on the writing area. After tacking the input, it performs the segmentation. It counts the number of lines, then

segment each line to find the words and finally each word is segmented into the single character. This single character goes to the recognition process.

In cognitive computing, AI, machine learning also uses of NLP to understand the human machine interaction. Many applications are available which work on handwriting recognition like puzzle problem, Sudoku is also well-liked, crossword and many other verification application like signature verification etc [23].Another problem is skewness, disjoin character detection which is sometimes give bad recognition. To remove this all constraints and make it independent from all fused, skew, slant and all other type of problem, want to add some more functionality based on the stroke based recognition. In that assign uniquely number to the each stroke and based on the stroke combination, identify the character. For this first make, the rule based file in which all the combination of the stroke and corresponding to that the Unicode value of character is stored. This is the future task to enhance the accuracy of the project and remove the constraints of this project.

References

[1].RejeanPlamondon and Sargur N. Srihari “On-Line and Off-Line Handwriting Recognition” A Comprehensive Survey 1EEE Transactions on pattern analysis and machine intelligence. vol. 22, no. 1.pp. 63-84 january 2000

[2] T. Artieres and P. Gallinari. Stroke level HMMs for online handwriting recognition. In IWFHR ’02: Proceedings of the Eighth International Workshop on Frontiers in Handwriting Recognition (IWFHR’02), page 227, Washington, DC, USA, 2002. IEEE Computer Society.

[2] Hidetoshi Miyao and Minoru Maruyama “Writer Adaptation for Online Handwriting Recognition System Using Virtual Examples” Shinshu University, 4-17-1 Wakasato, Nagano 380-8553, Japan 2009 10th International Conference on Document Analysis and Recognition

[3] T. Artieres, S. Marukatat, and P. Gallinari.Online handwritten shape recognition using segmental hidden markov models.IEEE Trans. Pattern Anal. Mach. Intell., 29(2):205–217, 2007.

[5] Goyal, Pulkit, Sapan Diwakar, and Anupam Agrawal. "Devanagari character recognition towards natural human-computer interaction." Proceedings of India HCI 2010. No. EPFL-CONF-168804. 2010

[6] Vyas, Mayur, and Karun Verma. "A platform independent methodology for on-line handwritten character segmentation." Advanced Communication Control and Computing Technologies (ICACCCT), 2014 International Conference on. IEEE, 2014

[7] A. L. Knoll, “Experiments with characteristic loci for recognition of hand printed characters,” IEEE Trans. Comput., vol. C-18, pp. 366-372, Apr. 1969.

[8] P. W. Becker and K. A. Nielsen, “Pattern recognition using dynamic pictorial information,” IEEE Trans. Syst., Man, Cybern., vol. SMC-2, pp. 434-437, July 1972.

[9] S. A. Guberman and V. V. Rozentsveig, “Algorithm for the recognition of handwritten text,” Avtomarika i Telernekanika, vol. 5, pp. 122-129, May 1976.

[10] Schantz, Herbert F. (1982). The history of OCR, optical character recognition. [Manchester Center, Vt.]: Recognition Technologies Users Association. ISBN 9780943072012.

[11] R. Nag, K.H. Wong, and F. Fallside, “Script Recognition Using Hidden Markov Models,” Proc. TCASSP ’86, vol. 3, pp. 2,071-2,074, Japan, Apr. 1986.

[12] Jianying Hu, Michael K. Brown and William Turin, “HMM Based On-Line Handwriting Recognition” IEEE Transactions on pattern analysis and machine intelligence, vol. 18, no. 10, pp. 1039-1045 October 1996.

[13] J. B. Kruskal, “An overview of sequence comparison: Time warps, string edits, and macromolecules,” SIAM Rev., vol. 25, pp. 201-237, Apr. 1983.

[14] D. Sankoff and J. B. Kruskal, Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison. London: Addison-Wesley, 1983.

[15] C.C. Tappert, ”Cursive Script Recognition by Elastic Matching,” IBM J. Research Development, vol. 26, pp. 765-771, Nov. 1982.

[16] J. J. Denier van der Gon, 1. P. Thuring, and J .Strackee, “A handwriting simulator” Phys. Med. Bioi., vol. 6, pp. 407-414. 1962.

[17] J. J. Denier van der Gon and J . P. Thuring, “The guiding of human writing movements,” Kybernetik, vol. 2, pp. 145-148, Feb. 1965.

[18] “Handwriting generation and recognition,” in P. A. Kolers and M. Eden, Ed., Recognizing Patterns. Cambridge, MA: M.I.T. Press, 1968, pp. 138-154.

[19] “Handwriting and pattern recognition,” IRE Trans. Inform. Theory, vol. IT-8, 1962.

[20] R.L. Hoffman and J.W. McCullough, "Segmentation Methods for Recognition of Machine-Printed Characters," IBM I. Research and Development, pp. 153-65, Mar. 1971.

[21] H.S. Baird, S. Kahan, and T. Pavlidis, ”Components of an Omni- font Page Reader,” Proc. Eighth Int’l Conf. Pattern Recognition, Paris, pp. 344-348,1986.

[22] M. Cesar and R. Shinghal, ”Algorithm for Segmenting Hand- written Postal Codes,” Int’l J. Man Machine Studies, vol. 33, no. 1,pp. 63-80, July 1990..

[23] L. Hong, and B. Verma. "Binary segmentation algorithm for English cursive handwriting recognition." Pattern Recognition 45.4 (2012): 1306-1317.

User's	Draw Character	Correct Recognize	Accuracy
1	490	483	91.02%
2	471	460	93.56%
3	454	452	96.29%
4	463	456	88.88%
5	535	524	91.85%
6	572	565	90.27%
7	608	602	94.44%
8	426	418	93.65%
9	481	478	96.29%
10	478	468	
11	590	583	91.02%
12	671	660	93.56%
13	554	552	96.29%
14	463	456	88.88%
15	435	424	91.85%
16	572	565	90.27%
17	535	24	91.85%
18	472	65	90.27%
19	608	102	94.44%

20			

List of Publications

1. Gaurav Pagare and Karun Verma, "Associative Memory Model for DistortedOn-line Devanagari Character Recognition," accepted at 5th International Conference on Advances in Computing and Communications (ICACC-2015), Kochi, 2015 will be published in "*IEEE Computer Society's CPS ".*

Video Presentation Link

https://www.youtube.com/watch?v=LtXQwuAtKKU&feature=youtu.be