Tribhuvan University

Institute of Engineering, Pulchowk Campus

A

Thesis Report

on

Nepali Character and Word Recognition using Neural Network

Submitted by:
Ram Chandra Pandey

069/MSCS/661

Submitted to:

Department of Electronics and Computer Engineering

October, 2016

Tribhuvan University

Institute of Engineering, Pulchowk Campus

A

Thesis Report

on

Nepali Character and Word Recognition using Neural Network

Submitted by:
Ram Chandra Pandey

069/MSCS/661

Submitted to:

Department of Electronics and Computer Engineering

October, 2016

ACKNOWLEDGEMENT

I am very grateful to Department of Electronics and Computer Engineering, Pulchowk Campus for providing finest scenario to present myself with a thesis report entitled "**Nepali character and word recognition using Neural Network**".

I would like to express my enormous gratitude to my supervisor Mr. Babu Ram Dawadi for his guidance, encouragement and support throughout the work.

I would like to thank Prof. Dr. Sashidhar Ram Joshi, Prof. Dr. Subarna Shakya and Dr. Sanjeeb Prasad Pandey for their guidance, advice and suggestions regarding thesis work.

I would like to express sincere gratitude to Dr. Diwakar Raj Pant, HOD Department of Electronics and Computer Engineering, for providing sound academic environment.

I am grateful and express my candid gratitude to Dr. Arun K. Timalsina, Dr. Aman Shakya, Mr. Daya Sagar Baral and Mr. Sharad Kumar Ghimire for providing their valuable suggestions and ideas.

I would like to thank my friends and seniors who helped me directly and indirectly and for their valuable suggestion and helping me carrying out the experiments. Finally I would like to thank all those writers of the paper which I have taken for reference for my thesis.

Ram Chandra Pandey
069MSCS661

ABSTRACT

The OCR systems developed for the Nepali language carry a very poor recognition rate due to error in character segmentation, ambiguity with similar character, unique character representation style. The purpose of this thesis work is to take image of handwritten or printed Nepali characters and words as input, process the character, train the neural network algorithm, to recognize the pattern and convert to digital form of the input. In this thesis, proposing an OCR for Nepali text in Devanagari script, using multi-layer feed forward back propagation Artificial Neural Network (ANN), which will improve its efficiency and accuracy. Adaptive learning rate with Gradient descent algorithm is proposed in Neural net with two hidden layers used with input and output and MMSE is the performance criteria. Various classifiers for training characters need to be created and stored.

Keywords*: Neural Network, Nepali handwritten datasets, handwriting recognition.*

Table of Contents

ACKNOWLEDGEMENT iii
ABSTRACT iv
LIST OF FIGURES vi
LIST OF ABBREVIATIONS vii
1. INTRODUCTION 1
1.1. Background 1
1.2. Devanagari, the script 2
1.3. Problem Definition 5
1.4. Special care for Devanagari 6
1.5. Objectives 7
1.6. Organization of report 8
2. LITERATURE REVIEW 9
3. METHODOLOGY 13
3.1. Image Acquisition 13
3.2. Pre-processing 14
3.2.1. Binarization 14
3.2.2. Edge Detection 14
3.2.3. Skeletonization 15
3.2.4. Thinning the Image 16
3.3. Segmentation 18
3.4. Feature extraction 19
3.5. Classification 20
3.5.1 Morphological operation 20
3.5.2 Training of Classification 21
3.6. Post Processing 23
3.7. Morphological operation 24
3.8. Training of Classification 25
4. OBTAINED RESULTS AND ANALYSIS 26
5. CONCLUSION AND FUTURE WORK 36
5.1. Conclusion 36
5.2. Future Work 36
REFERENCES 38

LIST OF FIGURES

Figure 1.1 Different zones of Nepali Word 2
Figure 1.2 Removal of Core strip 6
Figure 1.3 Similarities between different characters once the Shiro Rekha is removed 6
Figure 1.4 Similarities between different characters once the bottom strip is removed 7
Figure 2.1 Image before binarization (left); Image after binarization (right) 9
Figure 3.1 Grouping of letters on the basis of graphical similarities [14] 13
Figure 3.2 skeletonize after multiple erosion, skeleton give structure for text 15
Figure 3.3 Training and testing process for generic character recognition system 18
Figure 3.4 log sigmoid activation function for character recognition 22
Figure 3.5 Single-layer network Feed forward Network 23
Figure 3.6 Horizontal line Profiles of a document for line segmentation 24
Figure 3.7 Vertical Projection Profiles of a document for word separation 24
Figure 3.8 Nepali alphabets 25
Figure 3.9 Nepali numeric character. 25
Figure 3.10 Peculiarity of Devanagari Character 25
Figure 4.1Image before binarization (top); Image after binarization (down) 26
Figure 4.2 Horizontal line Profiles of a document for line segmentation 26
Figure 4.3 Word segmentation by intensity distribution 27
Figure 4.4. Steps to make individual character 28
Figure 4.5 Steps to make individual character for complex word 28
Figure 4.6. Processing of upper part of line, and crop 29
Figure 4.7.Processing of Body part of line, and crop. 29
Figure 4.8 Dataset of all Nepali characters and components to train NN. 30
Figure 4.9 Dataset of all Nepali characters and components to test NN. 31
Figure 4.10. Trained network architecture for 36 Nepali characters 31
Figure 4.11: Convergence of network for fuzzy data 32
Figure 4.12 Learning Rate versus epoch using gradient descent 32
Figure 4.13 Input image (top) and its approximated output (down) 33
Figure 4.14 Nepali word and sentence recognition 34
Figure 4.15 Nepali word and sentence recognition with certain error 34
Figure 4.16 Complex Nepali word and sentence recognition 35

LIST OF ABBREVIATIONS

ASCII	American Standard for code
BP	Back Propagation
GUI	Graphical User Interface
HGCR	Handwritten Gurumukhi character recognition
ICR	Intelligent Character Recognition
MMSE	Minimum Mean Square Error
NN	Neural Network
OCR	Optical Recognition System
RGB	Red Green Blue
SOM	Self Organizing Map

1. INTRODUCTION

1.1. Background

Unlike English character recognition, Nepali languages are complicated in terms of structure and computations. The Nepali language is derived from Devanagari Script; written from left to right fashion having common features of containing straight line on top ‘Shiro Rekha’. Character or word recognition is the mechanism for converting the text into a notational representation. It is a special problem in the domain of pattern recognition and machine intelligence. Automatic character recognition have many application areas like postal addresses reading, bank check verification, ancient document digitalization, handwritten form verification, forensic and medical analysis, etc. The field of character recognition can be split into two different categories: on-line recognition and off-line recognition. Online mode deals with the recognition of characters captured by a tablet or touch-screen device, and use the digitized trace of the pen to recognize the symbol. In the on-line case captured trajectory, pan up and pen down time, stroke orders, etc. of the written characters. Off-line mode deals with the recognition of the character or word present in the digital image of written text with holistic image. Character or word recognition is currently a hot topic in the research society. The early researches after the digital age were concentrated either upon machine-printed text or upon a small scale of well-separated handwritten symbols. Generally, template matching techniques were used for machine printed character recognition and statistical classification techniques were used for handwritten text recognition [1].

As Nepali scripts has a horizontal line, which connects all the characters. This connector is called 'Shiro Rekha' or headline. Based on the horizontal lines, Nepali scripts can be divided into three zones: top zone, core zone and bottom zone. The core and top zone are separated by the header line. Figure 1.1 shows the image of a word that contains three characters with both lower and upper modifiers.

OCR (Optical Character Recognition) also called Optical Character Reader is a system that provides a full alphanumeric recognition of printed or handwritten characters at electronic speed by simply scanning. Recognition is the mapping of a

low-level vector to a higher-level concept, For example mapping bitmaps to characters. Learning is to find out which low-level vectors correspond to high-level concepts. Intelligent Character Recognition (ICR) has been used to describe the process of interpreting data image for particular alphanumeric text. Images of handwritten or printed characters are turned into ASCII data (machine-readable characters). Usually, OCR uses a modular architecture that is open source, scalable, and workflow controlled. It includes forms definition, scanning, image pre-processing and recognition capabilities [2,3].

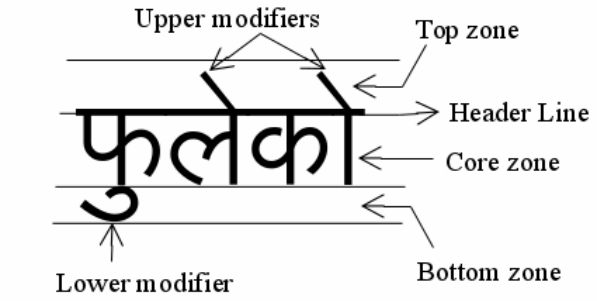


Figure 1.1 Different zones of Nepali Word

Even to this day and age, humanity's concentrated efforts continue for the perfect machine/computer that can emulate the immaculate sensory abilities of the humans that have been perfected over centuries of evolutionary trial-and-run mutations. This daunting task includes, but is not restricted to, conceiving a machine that is able to sense and understand its surrounding like how humans do and produce some kind of a turnout that is useful. The most basic of all human communications writing happens to hold an essential key in bringing about this effect, and any machine, worth its salt, should at least be able to recognize basic human writing on a script level, if not its implications and nuances. Any undertaking that aims to engender such an intelligent computer that recognizes human handwriting comes under the broad purview of what is known as handwriting recognition [3].

1.2. Devanagari, the script

Devanagari is an ancient used to write languages such as Nepali, Sanskrit, Hindi, Marathi and several others, Nepali is being the official language in Nepal.

Algorithms that are aimed at providing high recognition rates for online Devanagari script recognition will prove beneficial to certain percentage of world's population. Although a lot of work has been reported for online handwriting recognition in English and Asian languages such as Japanese and Chinese, there have been very few attempts at online Devanagari handwriting recognition. Thus, the need for more efficient online handwriting recognition algorithms and the under-represented status of Devanagari script set the premise for the work done under this thesis.

Due to these problems, Propose system is required,

- Paper stack up; productivity goes down
- Money expensive
- Type all document from scratch
- Digitalization
- Security and long life
- Space
- Store share find information

Artificial Neural Network (ANN) is nonlinear parallel distributed highly connected mathematical model or computational model network having capability of adaptively, self-organization, fault tolerance, evidential response and closely resemble with physical nervous system. ANN system can perceive and recognize a character based on its topological features such as shape, symmetry, closed or open areas, and number of pixels. The advantage of such a system is that it can be trained on samples and then can be used to recognize characters having a similar (not exact) feature set. The ANN used in this system gets its inputs in the form of Feature Vectors i.e. every feature or property is separated and assigned a numerical value.

Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing neurons working in unison to solve specific problems.

ANNs, like people, learn by example. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons. One of the primary means by which computers are endowed with humanlike abilities is through the use of a neural network. Neural networks are particularly useful for solving problems that cannot be expressed as a series of steps, such as recognizing patterns, classifying them into groups, series prediction and data mining. Pattern recognition is perhaps the most common use of neural networks. The neural network is presented with a target vector and also a vector which contains the pattern information, this could be an image and hand written data. The neural network then attempts to determine if the input data matches a pattern that the neural network has memorized [2].

A neural network trained for classification is designed to take input samples and classify them into groups. These groups may be fuzzy, without clearly defined boundaries. This project concerns detecting free handwritten characters. Either humans or other computer techniques can use neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, to extract patterns and detect trends that are too complex to be noticed. A trained neural network can be thought of as an expert, which can be used in project new situation. Other advantages include:

- Adaptive learning: An ability to learn how to do tasks based on the data given for training or initial experience.
- Self-Organization: An ANN can create its own organization or representation of the information it receives during learning time.
- Real Time Operation: ANN computations can be carryout in parallel, and special hardware devices are being designed and manufactured which take advantage of this capability.
- Fault Tolerance via Redundant Information Coding: Partial destruction of a network leads to the corresponding degradation of performance. However, some network capabilities may be retained even with major network damage.

The firing rule is an important concept in neural networks and accounts for their high flexibility. A firing rule determines how one calculates whether a neuron should fire for any input pattern. It relates to all the input patterns, not only the ones on which the node was trained. Feed-forward ANNs allow signals to travel one way only; from input to output. There is no feedback (loops) i.e. the output of any layer does not affect that same layer. Feed-forward ANNs tend to be straightforward networks that associate inputs with outputs. They are extensively use in pattern recognition.

Feedback networks can have signals travelling in both directions by introducing loops in the network. Feedback networks are very powerful and can get extremely complicated. Feedback networks are dynamic; their 'state' is changing continuously until they reach an equilibrium point. They remain at the equilibrium point until the input changes and a new equilibrium needs to be found. Feedback architectures are also referred to as interactive or recurrent, although the latter term is often used to denote feedback connections in single-layer organizations. After segmentation, feature vector should be form based on this information. Every zone has a feature vector corresponding to it. The contents of each zone feature vector are:

- Number of horizontal lines.
- Number of vertical lines.
- Normalized Area of the Skeleton.

The feature vector is extracted individually for each zone.

1.3. Problem Definition

In latest research [1] are just able to recognize only individual character through OCR, not a complete set of words. In addition to that, these systems have less accuracy and not made for a complete word.

हाम्रो नेपाल राम्रो नेपाल

As in above text, that sentence is made of words, and each word has numbers of components, ie, ‘matras’, above shiro Rekha, ‘ekar or okar or ikar ’ and at lower part ‘ukar’ and etc. During segmentation and hence at recognition, these parts plays the vital role and due to various reason, the efficient system has not made yet.

1.4. Special care for Devanagari

Apart from the precautions stated above some special care has to be taken for Devanagari because of the complicated segmentation process. For character segmentation the script is divided in three parts: top, core (or middle) and bottom and all these parts are recognized separately. This increases the complication because unlike Latin script, descenders and ascenders of the characters (in core strip) won't be treated as the part of the character in Devanagari script. So no differentiating feature can be present in the ascender or descender of the character. These special precautions that need to be taken care of are discussed below.

Removal of Shiro Rekha is the second step in character segmentation. When Shiro Rekha is removed, all the features of the character at the level of Shiro Rekha or above it are also removed from the core strip as shown in figure 1.2.

Figure 1.2 Removal of Core strip

When some important features of the character are at the level of Shiro Rekha or above it gets removed resulting in no recognition or recognizing a different character. For example **भ** has a curve at the level of Shiro Rekha which when removed results in looking like **म**. Similarly **ध** looks like **घ** when the Shiro Rekha is removed which can be seen in figure 1.3.

Figure 1.3 Similarities between different characters once the Shiro Rekha is removed

The step after the removal of top strip in character segmentation is the removal of bottom strip. Bottom strip is the strip which contains the lower matras, halanta and descenders of the letters in the core strip. The most difficult part of this step is to determine where the core strip ends and the bottom strip begin because in Devanagari script the lower matras are connected to the characters the core strip.

Also a few characters like **इ झ** have characteristic features extending to the bottom strip. When these features are removed the character might closely resemble other characters as shown in figure 1.4.

Figure 1.4 Similarities between different characters once the bottom strip is removed

Recognition of characters is much more complicated in Devanagari than in Latin because of the graphical similarities in the letters. The graphical similarities in the letters in Devanagari are much more than that in Latin. Some of the letters have just a difference of a stroke like **ष** just has an additional diagonal stroke as compared to **प**. Also there are others which differ from each other only because of the presence of vertical line like **न** and **म**. Also unlike Latin script, Devanagari has letters which are disjoint horizontally. This should be avoided in the characters in which this can be avoided for example **ख** can also be designed as **ख**. This results in inaccurate recognition [4].

1.5. Objectives

- Develop a Nepali word recognition system using Neural Network.

1.6. Organization of report

This thesis is organized according to a logical structure and flow in such a way as to best present the different aspects of the research conducted. Chapter 2 focuses on the history of handwriting recognition and also contains the literature survey conducted as part of this research. Chapter 3 on design and methodology provides the reader with a clear picture of how existing tools are intelligently used, combined and modified in a sequence of rigorous steps to solve the research problem defined in the scope of this thesis. The results and analysis of each step detailed in chapter 4 are then presented in a logical and pictorial form. In chapter 5 the conclusions that could be drawn based on these results and the possibilities and ideas for future research in this area are discussed.

2. LITERATURE REVIEW

In Tamil Character Recognition by using Kohonen SOM technique to classifies handwritten and also printed Tamil characters. But it was not for joined letters and had less segmentation accuracy [1]. Recognize printed and handwritten characters by projecting them on different sized grids results showed that the precision of the character recognition depends on the resolution of the character projection [2]. The smaller letters were better recognized with the network with smaller resolution. Regardless the difference of the orientation, size and place of the characters, the network still had a 60% precision [2]. Simple pattern recognition can be done using artificial neural network to simulate character recognition. A simple feed-forward neural network model has been trained with different set of noisy data. The back-propagation method is used for learning in neural network. The experimental result shows recognition rate is 70% for noisy data to up to 99% [3]. Combining with several other modes of minimizing the searching space and helping the recognition with dictionary methods, neural networks can be a promising solution. In general, documents contain text, graphics, and images. The procedure of reading the text component in such a document can be divided into three steps: First, document layout analysis in which the text component of the document is extracted. Second, extraction of the characters from text component of the document, and finally recognized the segmented characters [4].

Binarization: printed documents generally are black text on white background. Process of converting colored or gray scale images to bi-level image is often known as binarization or thresholding. Binarization of image on both English and Nepali is shown in figure 2.1.

Figure 2.1 Image before binarization (left); Image after binarization (right)

Segmentation phase is a very crucial stage since this is where most of the errors occur. Even in good quality documents, sometimes adjacent characters touch each other due to inappropriate scanning resolution or the design of characters. This can create problems in segmentation. Incorrect segmentation leads to incorrect recognition. Its phase includes line, word and character segmentation. It occurs in three steps for OCR: line segmentation, word segmentation and character segmentation [5].

Line Segmentation, during letter segmentations, frequent problem occurs due to abnormally written characters (which misguide the system during recognition) therefore segmentation should be so precise. In line segmentation our aim is to separate out the line of text from the image. For this global horizontal projection profile method is used which constructs a histogram of all the black pixels in every row as shown in figure 3. Based on the peak/valley points of the histogram, individual lines are separated. Word Segmentation, after line segmentation the boundary of the line (i.e. the top and bottom of the line) is known [6]. Lots of work has been done in this field with the help of artificial neural network. ANN involves training of all characters. When unknown inputs are given to the system ANN is able to find out the most probable character by generalization. Numerous techniques for character recognition have been investigated based on four general approaches of pattern recognition, as suggested by Raghuraj template matching, statistical techniques, structural techniques, and neural networks. Hidden Markov Model is a complete statistical model that tries to predict the unknown sequence. So it also tries to recognize the unknown character which is given as input. If the difference between unknown input and training data is large, the system may not behave well. Also the HMM model does not capture the correlations between letters. Alexander J. Faaborg proposed a technique to create an adaptive character recognition system using neural network [7].

Back-Propagation neural Network with one hidden layer is used to create the system. System is trained and evaluated with printed and handwritten English alphabets. Author showed in his experimental results that printed text gives better accuracy in recognition than handwritten characters. The back propagation algorithm changes the schematic of the perception by using a sigmoidal function. The advantage of the sigmoidal function is that the sigmoidal function is differentiable. It works well on

simple training problems. However, as the problem complexity increases, the performance of back propagation falls off rapidly because of the fact that complex spaces have nearly global minima which are sparse among the local minima. Gradient search techniques tend to get trapped at local minima. Also BPN suffers from the scaling problem [8].

Neural networks with Back Propagation learning showed results by searching for various kinds of functions. However, the choice of the basic parameter often already determines the success of the training process. The selection of this parameter follow in practical use rules of thumb, but their value is at most arguable. Since first attempts to combine GA and NN started in the late 1980s, other researchers have joined the movement and created a flood of journal articles, technical reports etc [9].

In this paper, use of artificial neural network in applications can dramatically simplify the code and improve quality of recognition while achieving good performance. Another benefit of using neural network in application is extensibility of the system – ability to recognize more character sets than initially defined. Pattern association using back propagation algorithm is essential and helpful to optimize the association of input pattern to output pattern in the neural network. This approach can be used for pattern classification [10].

In this paper, focus on recognition of English alphabet in a given scanned text document with the help of Neural Networks. This paper carries out a study of various feature based classification techniques for offline handwritten character recognition. The feature extraction step of optical character recognition is the most important. At the current stage of development, the software does perform well either in terms of speed or accuracy but not better. It is unlikely to replace existing OCR methods, especially for English text. A simplistic approach for recognition of Optical characters using artificial neural networks has been described [11].

Develop a multi-layered neural network based algorithm using which a computer can learn to identify handwritten characters. A GUI is designed in MATLAB which enables the user to either train or test the network on a “one character at a time” basis. This thesis used a six element feature vector which is found to be sufficient for

reliably identifying all characters that can be entered using a standard US English QWERTY keyboard [12].

The work done under online HWR is vastly focused on Latin alphabet and numeral set. Interest in online HWR commenced in early 1950s, largely due to the advent and advancement in pen computing. Important contributions to the field of online HWR were made in 1960s and 1970s which were a result of availability of better tablets PCs, feature extraction and classification methods. The different hand writing recognition algorithms differed in the way the researchers chose to solve various issues that act as obstacles to the achievement of perfection in handwriting recognition. It can easily note that all online HWR algorithms work on the basic assumption that different handwritten characters in any given script have a high degree of de-correlation. While this is a useful property, another characteristic of handwriting that is at least as useful if not more is that the handwriting samples of the same given character in any particular script are highly correlated. A deep understanding of the variability in handwriting is essential to build better and efficient HWR systems and readers are referred to for a thorough discussion of the variability effects in English handwritten script [13].

Smoothing and filtering of the handwriting samples are vital pre-processing steps that have been implemented in various ways. The most commonly used method for smoothing is averaging of the sample points with its neighbors. As part of filtering, most researchers resample the data points so that consecutive points are all equally spaced. To bring about various types of normalizations, different algorithms have been suggested. Some of the examples are: deskewing algorithms that are carried out to de-slant the characters, size normalization algorithms in order to reduce or expand characters from different users which tend to be of different sizes to the same size as the case may be, stroke length normalization equalizes the number of sample points in each stroke so that all strokes have uniform length [14].

3. METHODOLOGY

Neural network learning is based on learning from examples and their respective classes. And in supervised learning, main goal is to build a classification system from a set of patterns available. Because of the variety of patterns and the difficulties in expressing empirical rules, character recognition is very often based on training a system with patterns. Neural networks are especially suitable for this recognition purpose. A typical handwriting recognition system consists of pre-processing, segmentation, feature extraction, classification and recognition, and post processing stages.

The anatomy of a letter can be defined as a system which depicts the structural form of a letter; describing key features of a letter in a typeface. The first attempt of graphical classification of Devanagari script was grouped letters on the basis of graphical similarities as shown in figure 3.1.

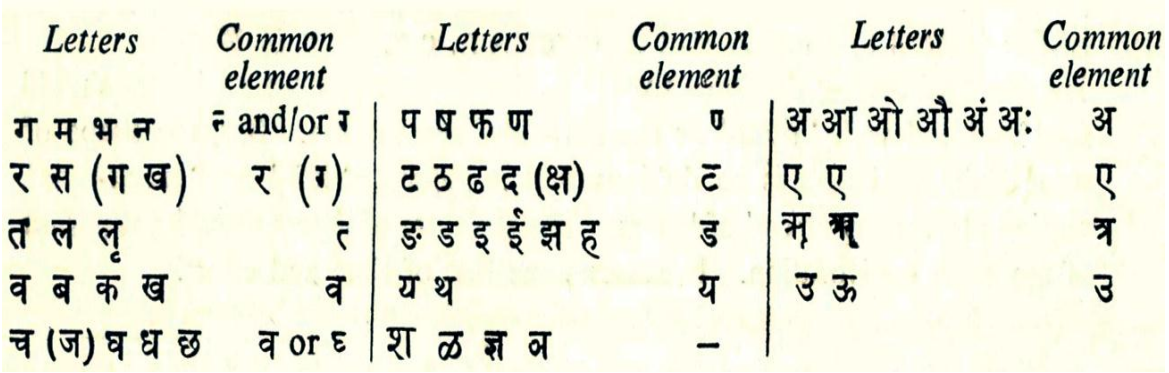


Figure 3.1 Grouping of letters on the basis of graphical similarities [14]

3.1. Image Acquisition

In Image acquisition, the recognition system acquires a scanned image as an input image. The image should have a specific format such as JPEG, BMT etc. This image is acquired through a scanner, digital camera or any other suitable digital input device. Here, Samples will read to the system through a scanner.

3.2. Pre-processing

The pre-processing is a series of operations performed on the scanned input image. Preprocessing converts the image into a form suitable for subsequent processing and feature extraction. The pre-processing is a series of operations performed on scanned input image. It essentially enhances the image rendering it suitable for segmentation. The role of pre-processing is to segment the interesting pattern from the background. Generally, noise filtering, smoothing and normalization should be done in this step. The pre-processing also defines a compact representation of the pattern which essentially enhances the image rendering it suitable for segmentation. The various tasks performed on the image in pre-processing stage. Binarization process converts a gray scale image into a binary image using global thresholding technique. Detection of edges in the binarized image using sobel technique, dilation the image and filling the holes present in it are the operations performed in the last two stages to produce the pre-processed image suitable for segmentation. The various tasks performed on the image in pre-processing stage. Binarization process converts a gray scale image into a binary image using global thresholding technique. Detection of edges in the binarized image using sobel technique, dilation the image and filling the holes present in it are the operations performed in the last two stages to produce the pre-processed image suitable for segmentation.

3.2.1. Binarization

Binarization or thresholding needed to find whether such pixel have significant intensity or not.

3.2.2. Edge Detection

The components of the gradient is found using

$$\frac{\partial f(x,y)}{\partial x} = \Delta x = \frac{f(x+dx,y)-f(x,y)}{dx}$$
$$\frac{\partial f(x,y)}{\partial y} = \Delta y = \frac{f(x,y+dy)-f(x,y)}{dy} \quad \text{……..3.1}$$

Where, f(x,y) is the intensity at x, y coordinate , and the difference in intensity in corresponding coordinate determines edge which is found by different filter, i.e. Spatial or frequency domain filter.

Sobel operators for a 3×3 mask are given as:

$$\Delta x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$\Delta y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & 2 & -1 \end{bmatrix} \quad \text{.....3.2}$$

Edge detection helps to extract useful features for pattern recognition. The Sobel operator makes the operator less sensitive to noise.

3.2.3. Skeletonization

Skeletonization is a process of reducing object regions in a binary image to a skeletal remainder that largely preserves the extent and connectivity of the original object while throwing away most of the original object pixels. It creates single pixel wide connected object boundary that preserves Euler number of the original object. Skeletonization calculates the medial axis skeleton so that points of this skeleton are at the same distance of its nearby borders. Image skeletonization technique used for the binary image skeletonization is the medial axis transformation technique, which iteratively delete boundary points of a region of the object [3].

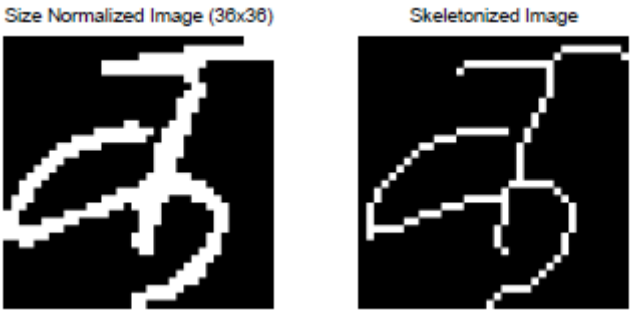


Figure 3.2 skeletonize after multiple erosion, skeleton give structure for text

Mainly, skeletonize image of text that provides nature of text, which further works as feature that distinguished between varieties of characters. Furthermore, erosion, dilution, opening and closing while coping with noise and broken edges in text should be done.

3.2.4. Thinning the Image

Since the algorithm is based on the geometrical and structural properties of the Nepali characters, here thin the image to single pixel width so the contours are brought out more vividly. In this way, the attributes to be studied later will not be affected by the uneven thickness of edges or lines in the symbol. Thinning is a morphological operation that is used to remove selected foreground pixels from binary images. The key here is the selection of the right pixels. For this, three categories of classification:

Critical Pixels – Pixels whose removal damages the connectivity of the image. Any pixel which is the lone link between a boundary pixel and the rest image is a Critical Pixel. Its removal will isolate the boundary pixel. Hence it should not be removed.

End Pixels – Pixels whose removal shortens the length of the image. An end pixel is connected to two or less pixels. Remembering that talking about 8-connectivity here. Different considerations have to be taken for 4-connectivity.

Simple Pixels – Pixels which are neither Critical nor End pixels. These are the ones that can be removed for thinning.

A smarter method than manually looking for suitable pixels is to use the Hit and Miss transform. Like other morphological operators, the behavior of the thinning operation is determined by a Structuring Element. The binary structuring elements used for thinning are of the extended type described under the hit-and-miss transform (*i.e.* they can contain both ones and zeros). The thinning operation is related to the hit-and-miss transform and can be expressed quite simply in terms of it. The thinning of an image *I* by a structuring element *J* is:

thin(I,J)= I – hit-and-miss (I,J) …….3.3
where the subtraction is a logical subtraction

In everyday terms, the thinning operation is calculated by translating the origin of the structuring element to each possible pixel position in the image, and at each such position comparing it with the underlying image pixels. If the foreground and background pixels in the structuring element exactly match foreground and background pixels in the image, then the image pixel underneath the origin of the structuring element is set to background (zero). Otherwise it is left unchanged. Note that the structuring element must always have a one or a blank at its origin if it is to have any effect.

The choice of structuring element determines under what situations a foreground pixel will be set to background, and hence it determines the application for the thinning operation. In fact, the operator is normally applied repeatedly until it causes no further changes to the image (*i.e.* until convergence).

The hit-and-miss transform is a general binary morphological operation that can be used to look for particular patterns of foreground and background pixels in an image. It is actually the basic operation of binary morphology since almost all the other binary morphological operators can be derived from it. As with other binary morphological operators it takes as input a binary image and a structuring element, and produces another binary image as output.

The hit-and-miss operation is performed in much the same way as other morphological operators, by translating the origin of the structuring element to all points in the image, and then comparing the structuring element with the underlying image pixels. If the foreground and background pixels in the structuring element exactly match foreground and background pixels in the image, then the pixel underneath the origin of the structuring element is set to the foreground color. If it doesn't match, then that pixel is set to the background color.

3.3. Segmentation

The most basic step in Character Recognition is to segment the input image into object from noisy background. This step separates out sentences from text and subsequently words and letters from sentences also. In the segmentation stage, an image of sequence of characters is decomposed into sub-images of individual character. The pre-processed input image is segmented into isolated characters by assigning a number to each character using a labeling process. This labeling provides information about number of characters in the image. Each individual character is uniformly resized into pixels. Normalization: After extracting the character need to normalize the size of the characters. There are large variations in the sizes of each Character. Hence, a method to normalize the size is needed. In this proposed system, the pre-processed input image needs to isolate characters by assigning a number to each character using a labeling process, known as segmentation. This labeling provides information about number of characters in the image. Each individual character need to be uniformly resized pixels for classification and recognition stage.
The training and testing process diagram is shown in figure 3.3.

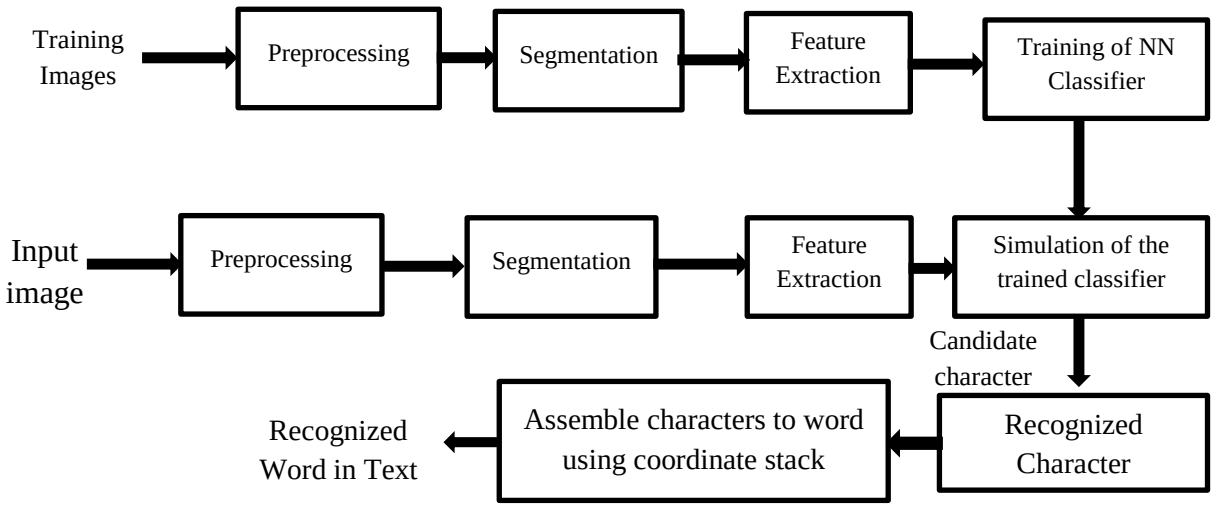


Figure 3.3 Training and testing process for generic character recognition system

3.4. Feature extraction

Each character has some features, which play an important role in pattern recognition. Nepali characters have many particular features. Feature extraction describes the relevant shape information contained in a pattern so that the task of classifying the pattern is made easy by a formal procedure. Feature extraction stage in HGCR system analyses these Punjabi character segment and selects a set of features that can be used to uniquely identify that character segment. Mainly, this stage is heart of HGCR system because output depends on these features. Feature extraction is the name given to a family of procedures for measuring the relevant shape information contained in a pattern so that the task of classifying the pattern is made easy by a formal procedure. Among the different design issues involved in building a recognizing system, perhaps the most significant one is the selection of set of features.

Feature extraction for exploratory data projection enables high dimensional data visualization for better data structure understanding and for cluster analysis. In feature extraction for classification, it is desirable to extract high discriminative reduced dimensionality features, which reduce the classification computational requirements. However, feature extraction criteria for exploratory data projection regularly aim to minimize an error function, such as the mean square error or the inter pattern distance difference whereas feature extraction criteria for classification aim to increase class separable as possible. The optimum extracted features calculated for exploratory data projections are not necessarily the optimum features regarding class separable and vice versa. In particular, two or more classes may have principal features that are similar. Moreover, feature extraction for exploratory data projection is used for two or three-dimensional data visualization, whereas classification usually needs more than two or three features. Consequently, feature extraction paradigms for exploratory data projection are not generally used for classification and vice versa [13].

After extracting the features, the data should be represented in one of two ways, either as a boundary or as a complete region. When the focus is on external shape characteristics such as corners and inflections then boundary representation is appropriate. While regional representation is appropriate when the focus is on internal properties such as textures or skeletal shape. In some applications like character

recognition these representations coexist, which often require algorithm based on boundary shape as well as skeletons and other internal properties. In terms of character recognition descriptors such as holes and bays are powerful features that help differentiate one part of the character from another. This description also called feature selection, deals with extracting features which results in some quantitative information of interest or features that are basic for differentiating one class of objects from another.

Feature extraction step plays an important role in the recognition procedure. A good feature vector should represent characteristic of the class that helps distinguish it from other classes, while remaining discriminant to characteristic differences within the class. Hundreds of features are available in the literature. Extraction of features i.e. geometry, of a character forms a vital part of the recognition process. Feature extraction collects the required information details of any characters. There are two techniques employed based on the efficiencies obtained, while training the neural network. They are feature extraction based on character geometry and feature extraction using gradient features.

3.5. Classification

During classification, a character is placed in the appropriate class to which it belongs. In training, the back propagation training algorithm subtracts the training output from the target (desired answer) to obtain the error signal. It then goes back to adjust the weights and biases in the input and hidden layers to reduce the error. Feed forward means there are no paths where signals travel backwards or sideways. Here more than 70 characters (36 Nepali alphabets, 12+ vowels, and 10 numeric characters, 10+ matras) to be distinctly classify, so it needs many samples and hence time to train the neurons.

3.5.1 Morphological operation

Morphology is a broad set of image processing operations that process images based on shapes. Morphological operations apply a structuring element to an input image, creating an output image of the same size. In a morphological operation, the value of

each pixel in the output image is based on a comparison of the corresponding pixel in the input image with its neighbors. By choosing the size and shape of the neighborhood and can construct a morphological operation that is sensitive to specific shapes in the input image.

The most basic morphological operations are dilation and erosion. Dilation adds pixels to the boundaries of objects in an image, while erosion removes pixels on object boundaries. The number of pixels added or removed from the objects in an image depends on the size and shape of the structuring element used to process the image. In the morphological dilation and erosion operations, the state of any given pixel in the output image is determined by applying a rule to the corresponding pixel and its neighbors in the input image. The rule used to process the pixels defines the operation as dilation or erosion. Also find properties of image regions i.e. ‘Shape’, 'Area', 'Centroid', 'Filled Area' and 'Major Axis Length' as features. Centroid of Processed image works as Feature to recognize.

3.5.2 Training of Classification

Train the neural network such that with input data and target data clustered and returns the network after training it. Feed forward back propagation algorithm is a method that depends on the gradient value of the moment. The learning starts when all of the training data was showed to the network at least once. For every network learning algorithm, consists of the modification of the weights and use the gradient of the criteria field to determine the best weight/modification to minimize the mean square error. Lastly, Compare trained and test classifier and if trained classifier has less MMSE with test, then corresponding character is known as recognized character.

3.5.2.1 Feed forward Backpropagation Algorithm

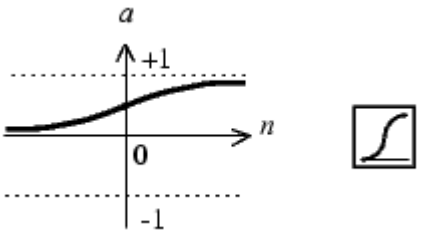
Feed-forward Back-propagation neural network learning is for training text as inputs. There are many variations of the back-propagation algorithm. The simplest implementation of back-propagation learning updates the network weights and biases

in the direction in which the performance function decreases most rapidly. One of the iteration of this algorithm can be written:

$x_{k+1} = x_k - \alpha_k g_k$ 3.4

Where X_k is a vector of current weights and biases, g_k is the current gradient, and $\acute{\alpha}_k$ is the learning rate.

In gradient descent method, the gradient is computed and the weights are updated after each input is applied to the network. Fast learning convergence is needed for input weight and layer weight, Momentum constant need to be higher, in addition to learning rate adaption of gradient. Therefore, to get faster convergence, Gradient descent with momentum and adaptive learning rate back-propagation will be used. In addition, accuracy of recognition is network performance function will be evaluated by measuring performance according to the sum of squared errors of neural net.



$a = logsig(n)$

Figure 3.4 log sigmoid activation function for character recognition

In proposed system multilayer networks will use the log-sigmoid transfer function. Sigmoid output neurons are often used for pattern recognition problems, while linear output neurons are used for function fitting problems.

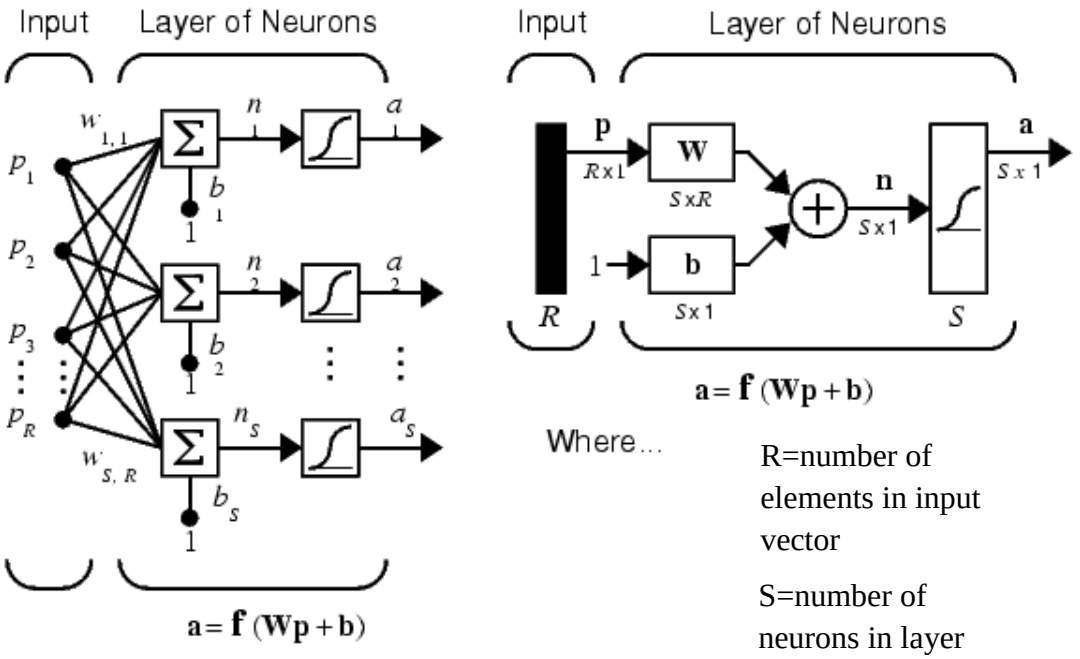


Figure 3.5 Single-layer network Feed forward Network

After extracting each line from a de-noised whole sheet image, start and end of each character word can be found from vertical projection profile. Here our logic is end of character ends with a maximum or symmetry with maximum or some additive length tailed from maximum defined by some higher pixel. This separates character, and works as test samples. Then each test character will be analyzed with stored classifier and calculated MSE.

3.6. Post Processing

Evaluate MMSE with Combines the train and test classifier and does for all trained classifiers. If MMSE is less then defined value then directly goes for next character recognition, which reduces execution time. Following steps have been followed in the training of characters system. Preprocessing: First scanned colored RGB image is converted to gray scale and then gray scale image is converted to binary. Preprocessing has done to improve the accuracy of the recognition algorithm. Main steps in preprocessing are salt and pepper noise removal, binarization, and skew correction. Then boundary of each character is detected. From histogram analysis ‘shiro rekha’ is detected as it has highest number of lower values of intensity pixels in text as shown on right portion of figure 3.6.

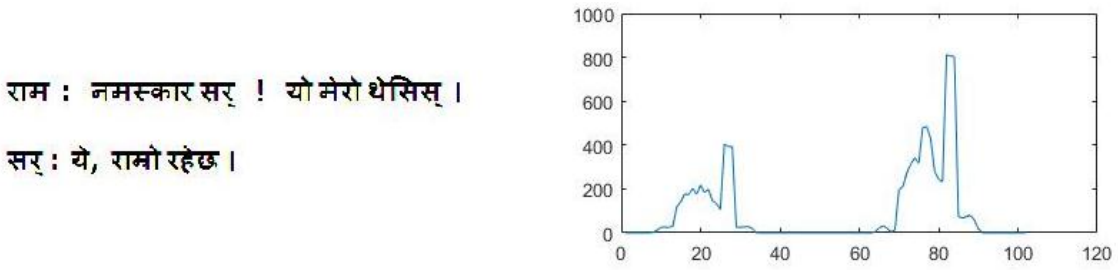


Figure 3.6 Horizontal line Profiles of a document for line segmentation

Word segmentation is done in the same way as line segmentation but in place of horizontal profiling, vertical projection profiling is done as shown in figure 3.7.

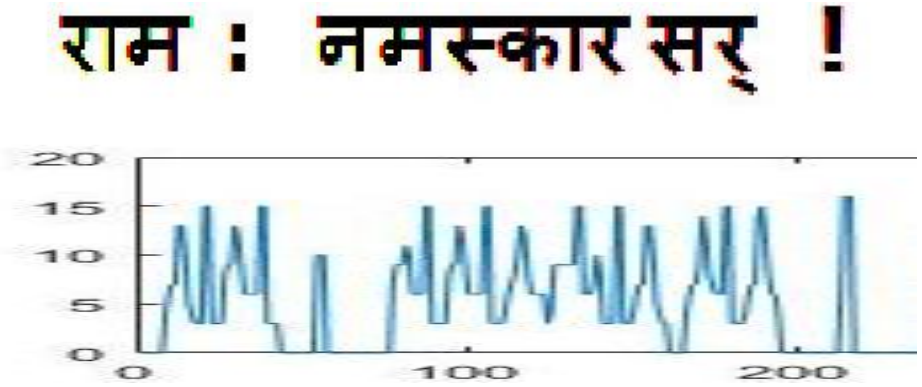


Figure 3.7 Vertical Projection Profiles of a document for word separation

3.7. Morphological operation

It is used to create morphological structuring of square element completed with erosion and then dilates to address each character. It was useful to de-noised image as well.

The main sources of noise in the input image are as follows:

- Noise due to the quality of paper on which the printing is done.
- Noise induced due to printing on both sides of paper or the quality of printing
- Noise added due to the scanner source brightness and sensors.

Here found the boundary of the image which was done by finding blank spaces at left/right/top/bottom. For that, measured properties of image regions i.e. ‘Shape’, 'Area', 'Centroid', 'Filled Area' and 'Major Axis Length'. And result is reshaped to 5 x 7 character representations in single vector from binary image. Centroid of Processed image works as Feature to recognize.

3.8. Training of Classification

Train the neural network such that with input data and target data clustered and returns the network after training it. Feed forward back propagation algorithm is a method that depends on the gradient value of the moment. The learning starts when all of the training data was showed to the network at least once. For every network learning algorithm, consists of the modification of the weights and use the gradient of the criteria field to determine the best weight/modification to minimize the mean square error. Lastly, Compare trained and test classifier and if trained classifier has less MMSE with test, then corresponding character is known as recognized character. Training data sets are shown in figure 3.8, figure 3.9 and figure 3.10.

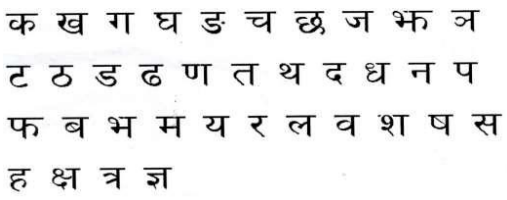


Figure 3.8 Nepali alphabets

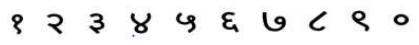


Figure 3.9 Nepali numeric character.

Figure 3.10 Peculiarity of Devanagari Character

After extracting each line from a de-noised whole sheet image, start and end of each character word can be found from vertical projection profile. Here our logic is end of character ends with a maximum or symmetry with maximum or some additive length tailed from maximum defined by some higher pixel. This separated character was taken as test samples. Then each test character is analyzed with stored classifier and calculated MSE.

4. OBTAINED RESULTS AND ANALYSIS

This thesis has main purpose to make accurate Nepali word recognition system that uses preprocessed image (e.g. with handwritten or printed text) as input and processes using Feed forward back propagation neural network.

De-noising: thresholding, first scanned colored RGB image is converted to gray scale and then gray scale image is converted to binary. Preprocessing has done to improve the accuracy of the recognition algorithm. Main steps in preprocessing are salt and pepper noise removal, binarization, and skew correction. Binarization of image on Nepali is shown in figure 4.1.

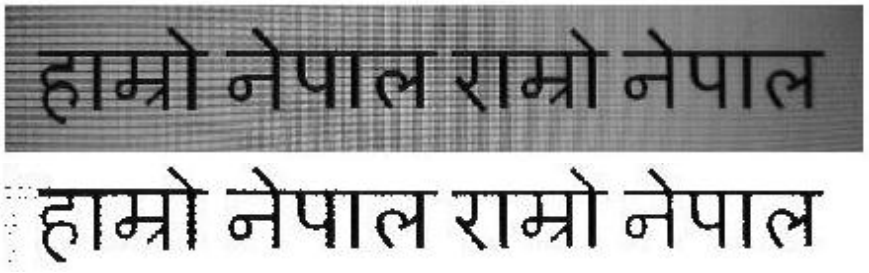


Figure 4.1Image before binarization (top); Image after binarization (down)

Then boundary of each character is detected. From histogram analysis 'sheuro rekha' is detected as it has highest number of lower values of intensity pixels in text as shown on right portion of figure 4.2.

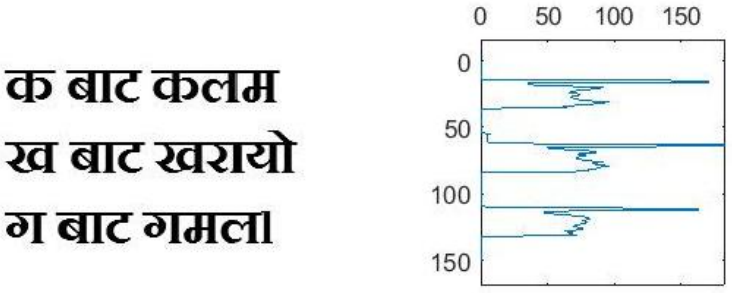


Figure 4.2 Horizontal line Profiles of a document for line segmentation

Intensity distribution gave the position of line, which had been used to separate different lines. Segmented separated single line is shown in figure 4.2, in addition

with its horizontal and vertical intensity distribution plot. In upper part of the figure 4.3, horizontal histogram is present. This figure clearly shows that “Shiro-Rekha” has maximum value as in all Nepali characters have Shiro-Rekha in its top. This position of ShiroRekha is further used in detection of upper and lower part and hence segmentation.

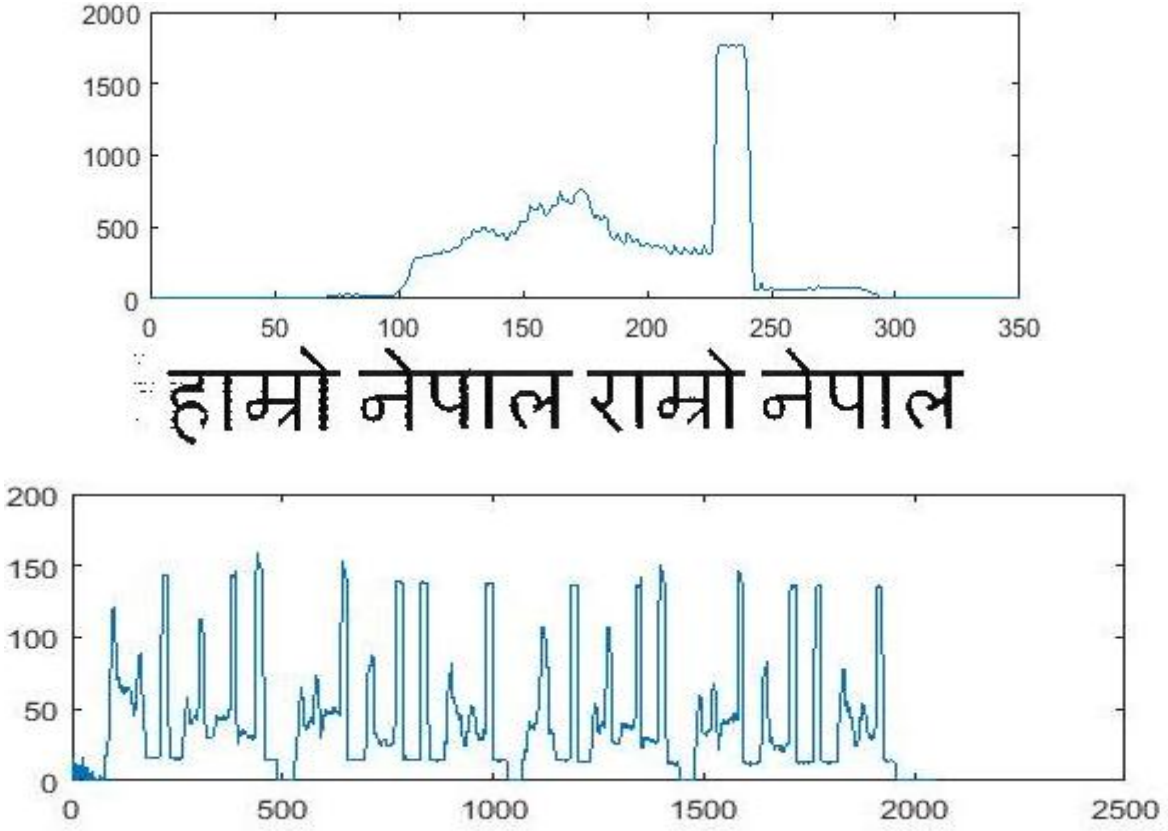


Figure 4.3 Word segmentation by intensity distribution

Vertical distribution is shown in lower part of figure 4.3. This plot has got meaning in finding character position. It shows the distribution varies character to character, so individual character can be identified with this shape of distribution only. But, it will arise several problems as two different characters can have similar shape. Therefore, this has two main advantages, firstly, to separate word to word as the Shiro line breaks while separating words, secondly, as a supporting feature to identify a character through its shape.

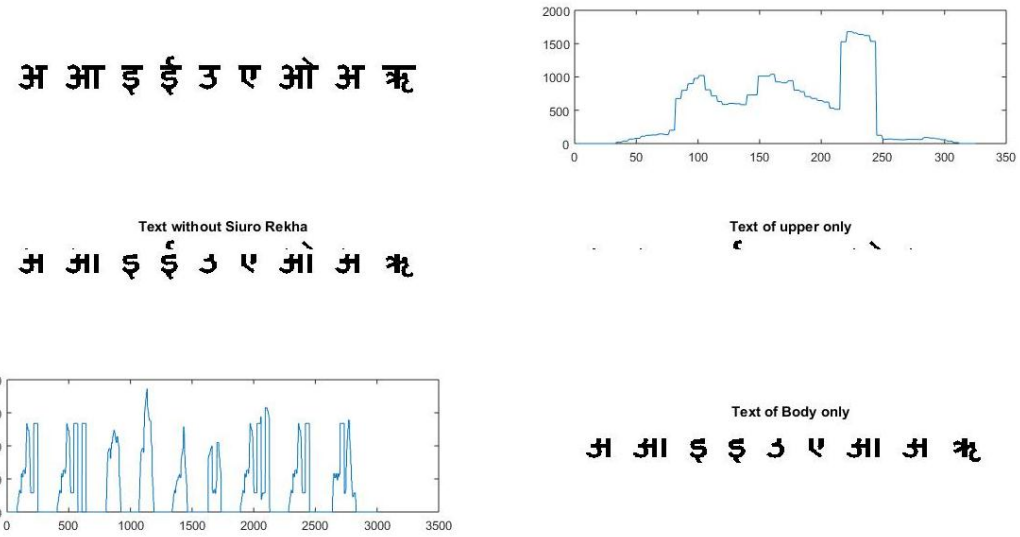


Figure 4.4. Steps to make individual character

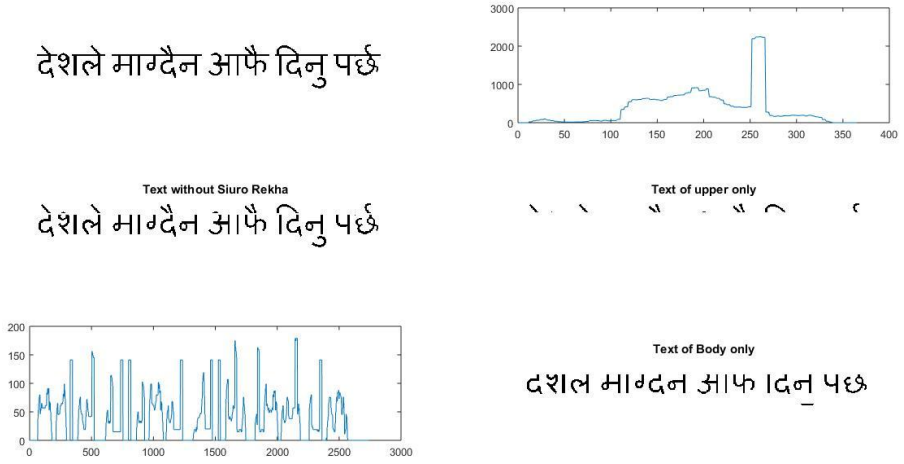


Figure 4.5 Steps to make individual character for complex word

Stepwise illustration of finding different part of a single line is shown in this figure 4.4 and figure 4.5. Basically, it shows step from de-noised scanned image is fed to the system to separation of different part. From horizontal distribution, the position of Shiro line was detected, which separates different portion of text. While removing Shiro, it was very easy to distinguish various segments in segmentation. In segmentation, firstly area of three major different segments was estimated and then each of the segments were placed in different matrix and processed to find its features.

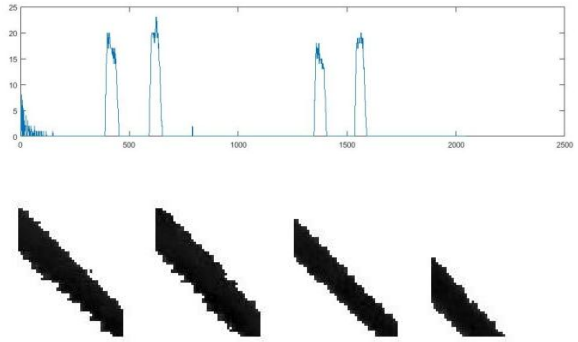


Figure 4.6. Processing of upper part of line, and crop

After finding region of different part of text, segmentation was done to find individual component of upper part. For that, taking only upper part in consideration, presence of each component was found and stored the component character along with its coordinate. Histogram of upper part and obtained character is shown in figure 4.6.

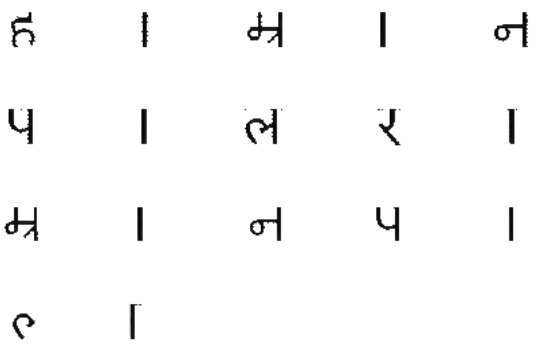


Figure 4.7.Processing of Body part of line, and crop.

As different part of text region has been stored separately, segmentation was done to find individual component character of body part. For that, taking only body part in consideration, presence of each component was found and stored the component character along with its coordinate. Each character has different shape, size, and length, for example, ‘Ra’ is shorter than any other character and so on. This creates problem in segmentation. Here in this thesis proposed an approach to find character by continuity check, which is based on hypothesis that from start to end of each character, there exist a continue intensity distribution. In addition to this hypothesis, Dictionary will reduce the inefficiency occurred due to segmentation. Obtained character is shown in figure 4.7.

To recognize Nepali word, one should be able to recognize character and all probable cases for the word, i.e. components present above or below to shiro and lower part of word respectively. For that case to work using computer, training of the component and character must take place in sufficient number. Epochs, iteration required to perform learn, determined by training rate where if learning rate is inversely proportional to epochs. As nature of characters in Nepali language is complex in terms of size, shape, geometry, or other, which leads to ambiguity in similar looking characters and finally the consequence will be inaccurate word recognition.

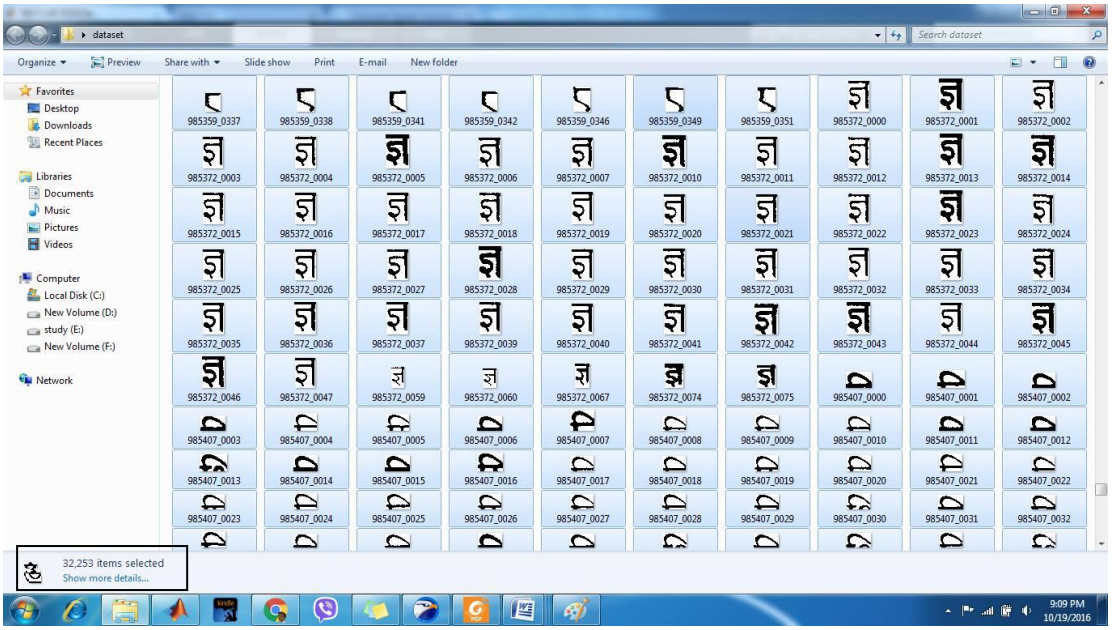


Figure 4.8 Dataset of all Nepali characters and components to train NN.

In contrast, training should be done in large number dataset with different characteristics. For this thesis, number of dataset for altogether different character, component used is more than 32K. Here, number of training data set chosen varies for different characters, for example, 'cha' has trained with less number of dataset as looks very unique and distinct from all other, whereas for 'nga' and 'da', dataset were very large, around 40 for each.

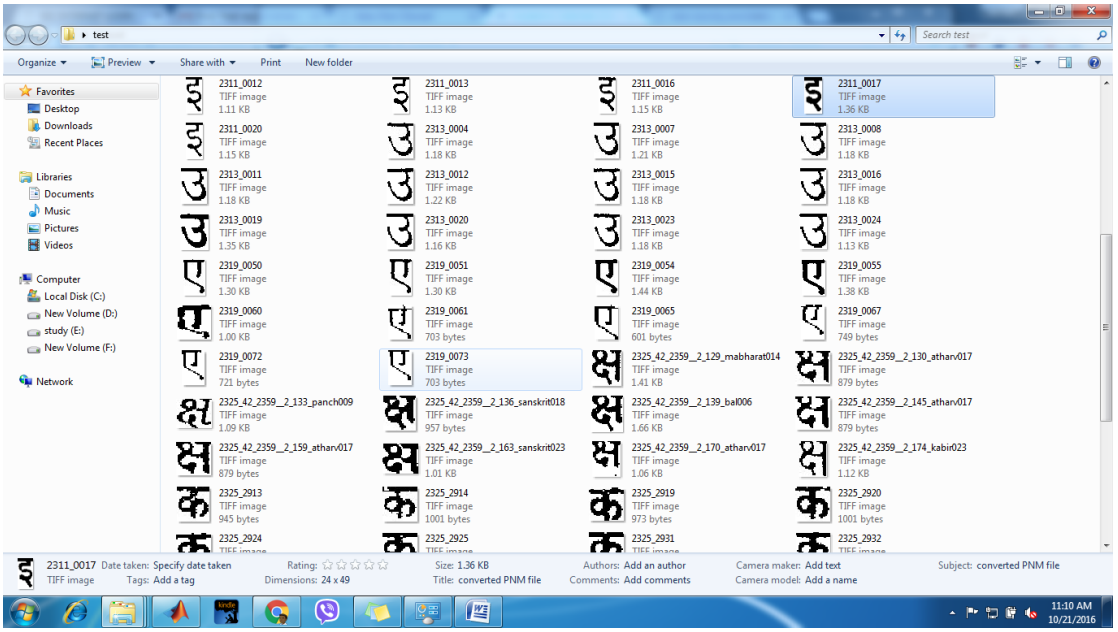


Figure 4.9 Dataset of all Nepali characters and components to test NN.

Test was done in large number dataset with different characteristics. For this thesis, number of dataset for altogether different character, component used is around 300 as shown in figure 4.9. This number of character is just for validating the recognition. Here, number of test data set chosen varies for different characters, for example, 'cha' has trained with less number of dataset as looks very unique and distinct from all other, where as for 'nga', 'da', and 'e' dataset were very large.

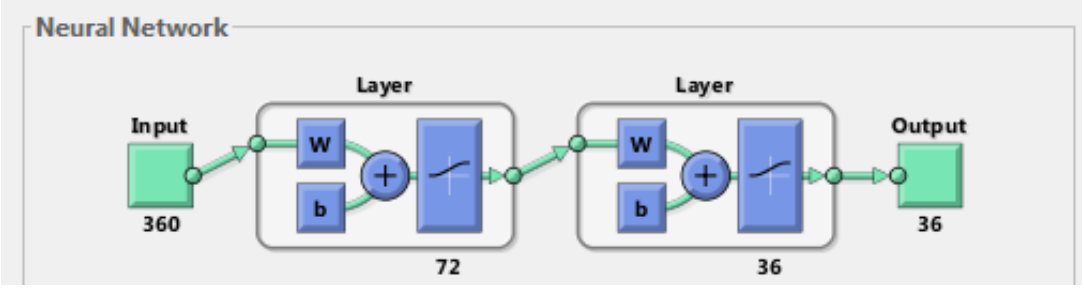


Figure 4.10. Trained network architecture for 36 Nepali characters

The segmented character should be tested in trained NN. So, to set weight of each neuron to recognize 36 different characters, dataset of 360 characters have been trained, considering training a character, it needs 10 differently written characters. The resultant neural network architecture is shown in figure 4.10.

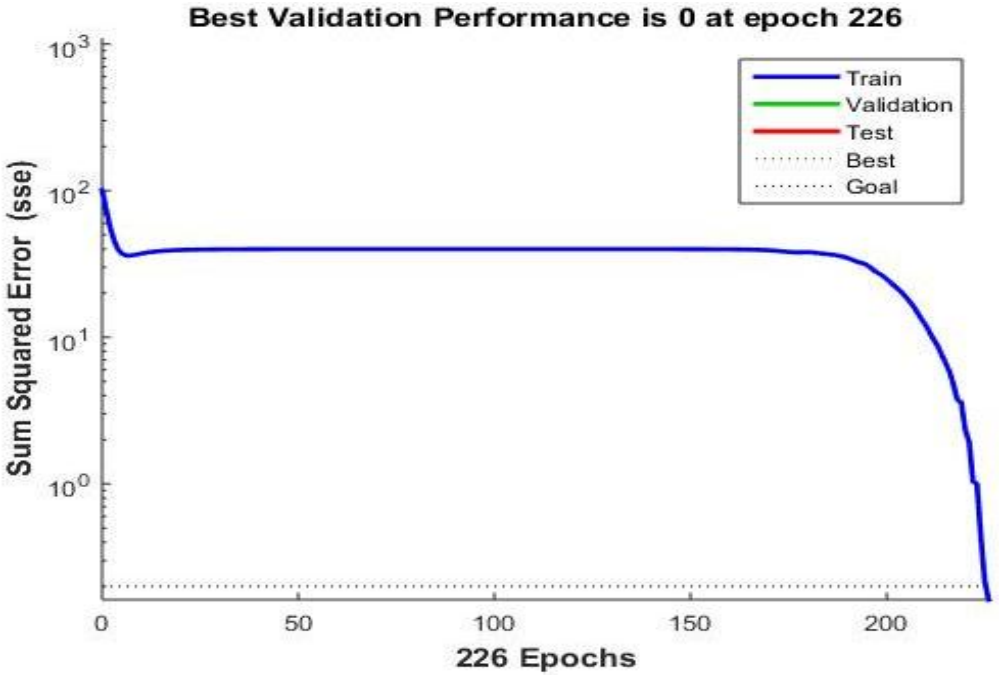


Figure 4.11: Convergence of network for fuzzy data

Epochs, iteration required to perform learn, determined by training rate where if learning rate is inversely proportional to epochs. As nature of characters in Nepali language is complex, leads to ambiguity in similar looking characters and finally the consequence will be inaccurate word recognition. In contrast, training should be done in large number dataset with different characteristics. In figure 4.11, SSE is shown for anga, as there was less uniqueness in that character.

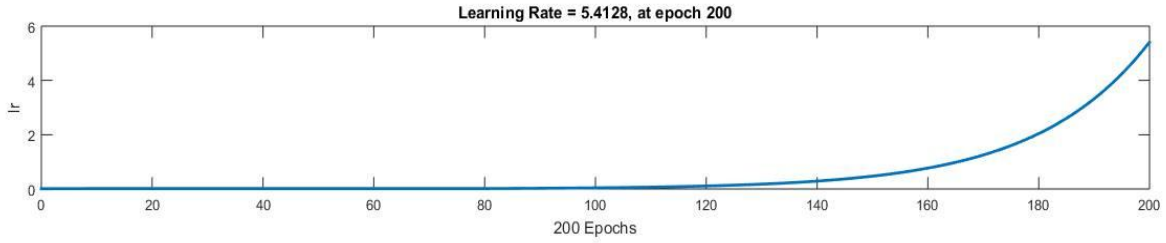


Figure 4.12 Learning Rate versus epoch using gradient descent

Creating template for Nepali fonts is hard because for train image, image size can be varying, i.e. Ka, KHA are horizontal and anga is vertical in shape. So Program need to run for each character but real condition is whole word is connected by horizontal line; is treated as noises. In addition, accuracy of network will be further increased using dictionary where it searches nearest Nepali word.

Computer uses training of the component and character place in sufficient number, to learn character. But for individual dataset, the character should be train until it recognize above definite accuracy level. Epochs means iteration required to perform learn, which determined by training rate shown in figure 4.12. In this thesis, Gradient descent algorithm to estimate learning rate is used, which have higher conversion rate, if learning rate is increases than number of epoch's decreases. As nature of characters in Nepali language is complex, leads to ambiguity in similar looking characters and finally the consequence will be inaccurate word recognition.

Individual segmented character were preprocessed and resized before feeding to trained Neural Network. NN provided the corresponding similar character identity, and from the identity the component is recognized in character level.

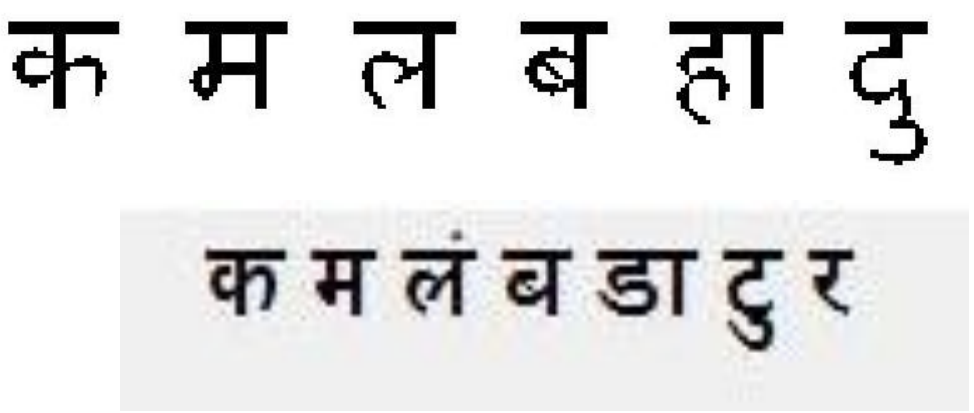


Figure 4.13 Input image (top) and its approximated output (down)

The trained neural network performed well on simple examples and successfully recognized using other methods previously. Those word which does not have upper and lower part is comparatively simple and high accurate in recognition. Mostly for character like, 'ha', 'ree', are typically with higher accuracy as indicated on figure 4.13 and 4.14.

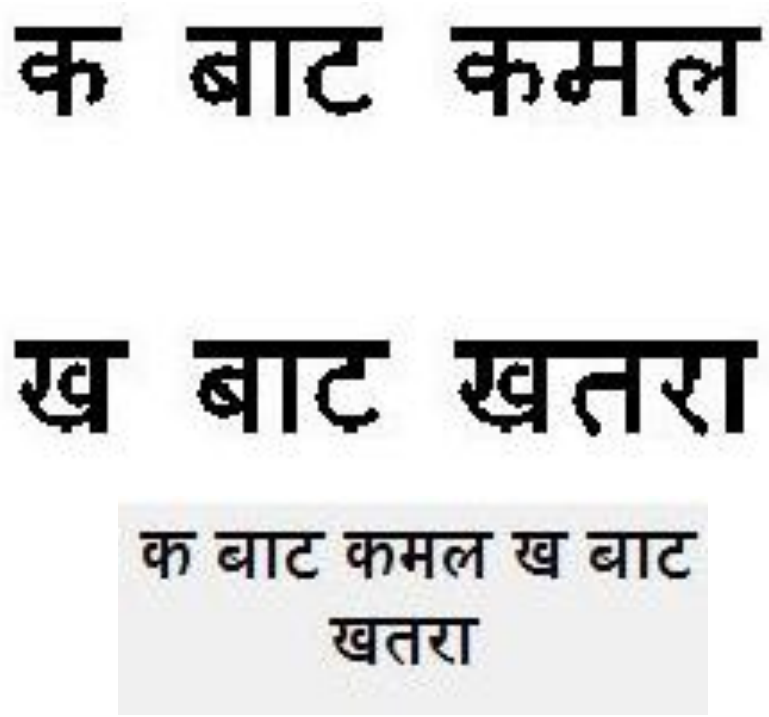


Figure 4.14 Nepali word and sentence recognition

As for complex cases, the network didn't work well. In this thesis, various distortions to the training sample to approximate it to the words received in receipts but it was unstable.

Figure 4.15 Nepali word and sentence recognition with certain error

Figure 4.15 describes inaccurate occurred due to complexity and problem obtained from less efficient segmentation. During segmentation, when an approximate character width in the string, word is divided by the length of the word by the character width and get the approximate number of characters, it made problem. Therefore character size is approximated with continuity check (8

connected), but there was still lots of problem as it does not work for ‘kha’, ‘ga’, and ‘ana’. Secondly, problem was due to large number of similar looking character in Devanagari like 'i' 'e', 'da', 'nga', and 'gha' 'dha'. While training these characters makes similar classifier and which lead less accuracy. Problem occurred during segmentation and neural network training for large number of training dataset which directly affect the thesis result and its performance as in figure 4.16.

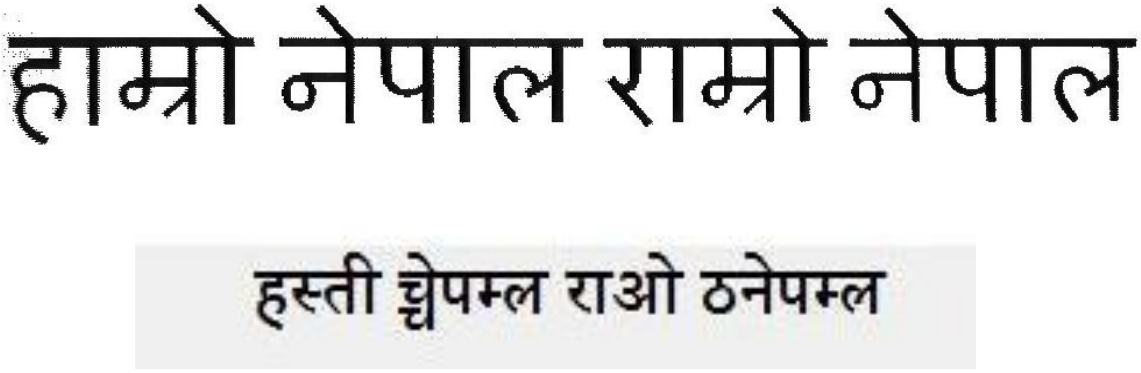


Figure 4.16 Complex Nepali word and sentence recognition

5. CONCLUSION AND FUTURE WORK

5.1. Conclusion

In this thesis, a solution for Nepali text recognition is demonstrated. The proposed system is a continuation of already proposed techniques with refinements in different modules using Neural Network for Nepali word. Segmentation technique is explored thoroughly in this study and its pros and cons have been revealed. Training with huge number of dataset has done and this lead higher recognition accuracy, about 90% for simple word. Optical Character Recognition is an open and challenging field for researchers to implement new ideas. The method described in this thesis is useful in training Nepali Character dataset. For creating complete Optical Character Recognition (OCR) system, this method is not very reliable due to similar looking characters and problematic on segmentation process.

The recognition accuracy of the prototype implementation is promising, but more work need to be done. In particular, no fine tuning of the system has been done so far. This character segmentation method also need to be improved so that it can handle a larger variety of touching characters, which occur fairly often in images obtained. Both of these challenges are font dependent. Hence, a detailed font study can help in finding good solutions for these challenges. Moreover, using this dataset exiting Nepali OCR can improve its accuracy.

5.2. Future Work

- Dictionary search for identified word and suggest nearest possible word. Even our propose segmentation process is clever enough to separate component and character, still may encounter problems in different steps. To limit its problem and increase its accuracy, dictionary mapping will be beneficial where similar looking word are place in cluster, and prompt result while searches in its database and accuracy increase for complex word.

- The research undertaken as part of this master's program, as presented in this thesis, is a complete, uncompromised and an accurate solution to the problem originally defined at the beginning of the program. Due to the very nature of any master's program, the critical parameter which governs the quantity of work that can be done beyond solving the original work is time.

- Handwritten complete sentence with complex words are recognition and translation, this extension would appear more complicated and complex word than the previous one.

REFERENCES

1. A. K. Pant, S. P. Pandey and S. R. Joshi “Off-line Nepali Handwritten Character Recognition using Multilayer Perceptron and Radial Basis Function neural networks,” Third Asian Himalayas International Conference on Internet (AH-ICI), pp 1-5, 2012
2. Banumathi, P., and G. M. Nasira. "Handwritten Tamil character recognition using artificial neural networks." Process Automation, Control and Computing (PACC), 2011 International Conference on. IEEE, 2011.
3. Arnold, Rókus, and Póth Miklós. "Character recognition using neural networks." Computational Intelligence and Informatics (CINTI), 2010 11th International Symposium on. IEEE, 2010.
4. Mani, Nallasamy, and Bala Srinivasan. "Application of artificial neural network model for optical character recognition." Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation., 1997 IEEE International Conference on. Vol. 3. IEEE, 1997.
5. Alpaydin, E. "Optical character recognition using artificial neural networks." Artificial Neural Networks, 1989., First IEE International Conference on (Conf. Publ. No. 313). IET, 1989.
6. M.Gunasekaram, S.Ganeshmoorthy,”OCR Recognition System Using Feed Forward And Back Propagation Neural network”, Department of MCA ,Park College of Engg & Tech, Coimbatore.
7. Puri, S. (2010). U.S. Patent No. 7,747,070. Washington, DC: U.S. Patent and Trademark Office.
8. C. Y. Suen, M. Berthod, and S. Mori, “Automatic recognition of handprinted characters: The state of the art,” Proceedings of IEEE, vol. 68, no. 4, pp. 469–487, Apr. 1980.
9. V. K. Govindan and A. P. Shivaprasad, “Character recognition: A review,” Pattern Recognition, vol. 23, no. 7, pp. 671–683, 1990.
10. R. M. Bozinovic and S. N. Srihari, “Off-line cursive script word recognition,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 11, no. 1, pp. 68–83, Jan. 1989.
11. S. Mori, C. Y. Suen, and K. Yamamoto, “Historical review of OCR research and development,” Proceedings of IEEE, vol. 80, no. 7, pp. 1029–1058, Jul. 1992.

12. C. C. Tappert, C. Y. Suen, and T. Wakahara, “The state of the art in on-line handwriting recognition,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-12, no. 8, pp. 787–808, Aug. 1990.
13. J. R Ward and T. Kuklinski, “A Mode l for Variability Effects in Handprinting with Implications for the Design of Handwriting Character Recognition System”, *IEEE Transactions on Systems, Man and Cybernetics, Vol. 18, No.3,* May/June 1988.
14. H. Arakawa, K. Odaka and 1. Masuda, “On-line recognition of handwritten characters Alphanumeric, Hiragana, Katakana, Kanji”, *Proc. 4th Int. Joint Con$ Pattern Recognition, pp. 810-812,* Nov 1978.
15. C. Kan and M. D. Srinath, “Invariant character recognition with Zernike and orthogonal fourier-mellin moments,” Pattern Recognition, vol. 35, no. 1, pp. 143–154, Jan. 2002.
16. N. Sharma, U. Pal, F. Kimura, and S. Pal, “Recognition of off-line handwritten devnagari characters using quadratic classifier,” in Indian Conference on Computer Vision, Graphics and Image Processing, 2006,
17. K. C. Santosh and C. Nattee, “Template-based nepali handwritten alphanumeric character recognition,” Thammasat International Journal of Science and Technology (TIJSAT), Thammasat University, Thailand,