
**TRIBHUVAN UNIVERSITY**

**INSTITUTE OF ENGINEERING**

**THAPATHALI CAMPUS**

**A Major Project Report On**

# Nepali Character Recognition Using ANN

**Submitted By:**

Akash Ranpal(43304)

Dipesh Shrestha(43313)

Kshitiz Bajgain(43319)

Shiva Aryal(43339)

**Submitted To:**

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

THAPATHALI CAMPUS

KATHMANDU, NEPAL

August, 2018


**TRIBHUVAN UNIVERSITY**

**INSTITUTE OF ENGINEERING**

**THAPATHALI CAMPUS**

# A Major Project Report
# On
# Nepali Character Recognition
# Using ANN

**Submitted By:**

Akash Ranpal(43304)

Dipesh Shrestha(43313)

Kshitiz Bajgain(43319)

Shiva Aryal(43339)

**Submitted To:**

Department of Electronics and Computer Engineering
Thapathali Campus
Kathmandu, Nepal

In partial fulfillment for the award of the Bachelor's Degree in Electronics and Communication Engineering

**Under the Supervision of**

Department of Electronics and Computer Engineering

Thapathali Campus

August, 2018

**DECLARATION**

We hereby declare that the report of the project entitled **"Nepali Character Recognition Using ANN"** which is being submitted to the **Department of Electronics and Computer Engineering, IOE, Thapathali Campus**, in the partial fulfillment of the requirements for the award of the Degree of Bachelor of Engineering in **Electronics and Communication Engineering**, is a bona fide report of the work carried out by us. The materials contained in this report have not been submitted to any University or Institution for the award of any degree and we are the only author of this complete work and no sources other than the listed here have been used in this work.

Akash Ranpal(071/BEX/304) _____

Dipesh Shrestha(071/BEX/314) _____

Kshitiz Baigain(071/BEX/320) _____

Shiva Aryal(071/BEX/340) _____

**Date:** August, 2018

**CERTIFICATE OF APPROVAL**

The undersigned certify that they have read and recommended to the **Department of Electronics and Computer Engineering, IOE, Thapathali Campus**, a major project work entitled "**Nepali Character Recognition Using ANN**" submitted by **Akash Ranpal, Dipesh Shrestha, Kshitiz Bajgain** and **Shiva Aryal** in partial fulfillment for the award of Bachelor's Degree in Electronics and Communication Engineering. The Project was carried out under special supervision and within the time frame prescribed by the syllabus.

We found the students to be hardworking, skilled and ready to undertake any related work to their field of study and hence we recommend the award of partial fulfillment of Bachelor's degree of Electronics and Communication Engineering.

_____

Project Supervisor

Department of Electronics and Computer Engineering, Thapathali Campus

_____

External Examiner

Dr. Surendra Shrestha

Department of Electronics and Computer Engineering, Pulchowk Campus

_____

Project Coordinator

Er. Saroj Shakya

Department of Electronics and Computer Engineering, Thapathali Campus

_____

Er. Janardan Bhatta

Head of the Department,

Department of Electronics and Computer Engineering, Thapathali Campus

August, 2018

## COPYRIGHT

**ACKNOWLEDGEMENT**

First of all, it is our radiant sentiment to place on the records, the deepest sense of gratitude to Head of Department, **Er. Janardan Bhatta** at the Department of Electronics and Computer Engineering, Thapathali Campus, IOE for providing the necessary resources and constant motivations to complete our project. This project would have been impossible without the proper guidance, supervision, technical supports and aids on resource assembly by our Project Supervisor, **Department of Electronics and Computer Engineering.**

We are indebted to our subject teachers, seniors and classmates for their regular backups in every requirements and circumstances we went through to complete this project. We would like to thank **Future Lab** for providing space and resources throughout the project development. It was always a pleasure to work in the inspirational and incentive environment provided by Robotics and Automation Center, Thapathali Campus and its members. We are always grateful to them.

Lastly we are thankful to Department of Electronics and Computer Engineering, Thapathali Campus for granting us the opportunity to do this project in order to update us to recent technological developments and furnish our practical skills related to the theoretical knowledge we acquired in the college.

Akash Ranpal(071/BEX/304)

Dipesh Shrestha(071/BEX/314)

Kshitiz Bajgain(071/BEX/320)

Shiva Aryal(071/BEX/340)

**ABSTRACT**

Advancement in Artificial Intelligence has led to the developments of various “smart” devices. The biggest Challenge in the field of image processing is to recognize documents both in printed and handwritten format. Besides, Devanagari characters are more challenging to recognize by the computer system because of its different letter format. Optical Character Recognition (OCR) is a type of document image analysis where scanned digital image that contains either machine printed or handwritten script input into an OCR software engine and translating it into an editable machine readable digital text format. A Neural network is designed to model the way in which the brain performs a particular task or function of interest. We have applied feature extraction technique for calculating the feature. Features extracted from character are directions of pixels with respect to their neighboring pixels. The inputs are given to the Convolution Neural Network (CNN) with hidden layers and output layer. We have used CNN for feed-forward method in the neural network of multiple layers. This system will be suitable for converting handwritten characters into structural text form.

*Keywords:Convolution Neural Network, Character Recognition, Multi-layer perceptron, supervised learning, TensorFlow*

**TABLE OF CONTENTS**

## List of Figures



## List of Tables

**List of Abbreviations**

| | |
|---|---|
| ANN | Artificial Neural Network |
| CNN | Convolution Neural Network |
| CPI | Counts per Inch |
| DIP | Dual In-line Package |
| GPU | Counts per Inch |
| IPS | Inches per Second |
| IRP | Input/output Request Packet |
| IT | Information Technology |
| OCR | Optical Character Recognition |
| PID | Packet Identifier |
| ReLU | Rectified Linear Unit |
| USB | Universal Serial Bus |

# 1. INTRODUCTION

The handwriting recognition refers to the identification of handwritten characters. Handwriting recognition has become an acute research area in recent years for the ease of access of computer applications. Traditional handwritten character recognition techniques enable computer to receive and interpret intelligible handwritten input from sources such as papers, documents, touch-screens or pictures. During the recent years many popular studies and applications merged for bank check processing, mailed envelops reading, and handwritten text recognition in documents and videos. Until now, it is still a difficult task for a machine to recognize human handwritings with significant accuracy, especially under variable circumstances such as variations in writings, variable sizes, and different patterns for different people, etc. Also several research works have been focusing on new techniques and methods that would reduce the processing time while providing the higher recognition accuracy.

The Nepali letters set comprises of 36 basic alphabets which are simple to write but challenging task to recognize and becomes very complex when they are handwritten. We have built a program that takes the input from a pen with mouse sensor, works on the same principle as the optical mouse does and recognizes the drawn image with artificial neural network techniques.

## 1.1 Background Introduction

Machine learning is a technology that gives computer system the ability to learn with data, without being explicitly programmed. Machine learning refers to the ability of machine to learn anything instructed, by itself, without extra use of command. It is mostly used for pattern recognition, computational learning theory in artificial intelligence, market basket analysis and many other programming explicit algorithms where good performance is difficult or not feasible.

Artificial neural networks (ANNs) are computing systems vaguely inspired by the biological neural networks that constitute animal brains. Such systems learn tasks by considering examples, generally without task-specific programming.

An ANN is based on a collection of connected units or nodes called artificial neurons (a simplified version of biological neurons in an animal brain). Each connection between artificial neurons can transmit a signal from one to another. The artificial neuron that receives the signal can process it and then signal artificial neurons connected to it. Optical Character Recognition, or OCR, is a technology that enables you to convert different types of documents, such as scanner paper documents, PDF files or image captured by digital camera into editable and searchable data.

## 1.2 Scope and Application

Almost everyone uses English as a standard language for typing. Everyone has knowledge of this particular language. But in context of our country we (almost 50% of population) lack knowledge of English language and prefer Nepali (being our national language). Moreover in our government offices, Nepali language is used as official language. So, for sending and retrieving any official letter one must know how to type in Nepali. In most of the text editor, typing in Nepali is difficult. If we use any text editor for Nepali typing, we won't get any separate key for all the words. Moreover, we have to memorize certain combination for most of the symbols used in Nepali typing (such as Chandra-bindu, Nepali vowel, and many more). It is tedious and tiresome for learning Nepali typing. So, we have built a system which helps in writing Nepali words in the screen instead of typing. There is a special pen which uses almost any smooth surface for writing and those written words are shown in the desktop application. It will be easy for all those who can write in Nepali and find hard to type. Main application of our project will be in government offices, banks and any organization which uses Nepali as primary or secondary official language.

## 1.3 Motivation

People feeling difficult for Nepali typing as compared to English typing has been the main driving factor for the thought of our project. So, to run with the need of digitization world with convenient effort, use of digital pen has been adopted instead of typing.

## 1.4 Problem Definition

The idea of our project has arisen as still educated people like us find Nepali typing on keyboard difficult. So, we have designed a system on why not write in usual paper display those written in the computer in text format.

## 1.5 Objectives

- To remove typing of Nepali character by writing means.
- Simple input from mouse based gestures.

# 2. LITERATURE REVIEW

Depending on the versatility, robustness and efficiency, commercial OCR systems may be divided into the following four generations [Line, 1993; Pal & Chaudhuri, 2004]. It is to be noted that this categorization refers specifically to OCRs of English language.

## 2.1 Generations of OCR

### 2.1.1 First generation OCR systems

Character recognition originated as early as 1870 when Carey invented the retina scanner, which is an image transmission system using photocells. It is used as an aid to the visually handicapped by the Russian scientist Tyurin in 1900. However, the first generation machines appeared in the beginning of the 1960s with the development of the digital computers. It is the first time OCR was realized as a data processing application to the business world [Mantas, 1986]. The first generation machines are characterized by the "constrained" letter shapes which the OCRs can read. These symbols were specially designed for machine reading, and they did not even look natural. The first commercialized OCR of this generation was IBM 1418, which was designed to read a special IBM font, 407. The recognition method was template matching, which compares the character image with a library of prototype images for each character of each font.

### 2.1.2 Second generation OCR systems

Next generation machines were able to recognize regular machine-printed and hand printed characters. The character set was limited to numerals and a few letters and symbols. Such machines appeared in the middle of 1960s to early 1970s. The first automatic letter sorting machine for postal code numbers from Toshiba was developed during this period. The methods were based on the structural analysis approach. Significant efforts for standardization were also made in this period.

### 2.1.3 Third generation OCR systems

For the third generation of OCR systems, the challenges were documents of poor quality and large printed and hand-written character sets. Low cost and high performance were also important objectives. Commercial OCR systems with such capabilities appeared during the decade 1975 to 1985.

### 2.1.4 OCRs Today (Fourth generation OCR systems)

The fourth generation can be characterized by the OCR of complex documents intermixing with text, graphics, tables and mathematical symbols, unconstrained handwritten characters, color documents, low-quality noisy documents, etc. Among the commercial products, postal address readers, and reading aids for the blind are available in the market. Nowadays, there is much motivation to provide computerized document analysis systems. OCR contributes to this progress by providing techniques to convert large volumes of data automatically. A large number of papers and patents advertise recognition rates as high as 99.99%; this gives the impression that automation problems seem to have been solved. Although OCR is widely used presently, its accuracy today is still far from that of a 14 seven-year old child, let alone a moderately skilled typist [Nagy, Nartker& Rice, 2000]. Failure of some real applications show that performance problems still exist on composite and degraded documents (i.e., noisy characters, tilt, mixing of fonts, etc.) and that there is still room for progress. Various methods have been proposed to increase the accuracy of optical character recognizers. In fact, at various research laboratories, the challenge is to develop robust methods that remove as much as possible the typographical and noise restrictions while maintaining rates similar to those provided by limited-font commercial machines [Belaid,1997]. Thus, current active research areas in OCR include handwriting recognition, and also the printed typewritten version of non-Roman scripts (especially those with a very large number of characters).[1]

## 2.2Different Models of Character Segmentation in OCR Systems

Character segmentation is an operation that seeks to decompose an image of a sequence of characters into sub-images of individual symbols. It is one of the decision processes in a system for OCR that decides a pattern isolated from the image is that of

a characters. The difficulty of performing accurate segmentation is determined by the nature of the material to be read and by its quality. Segmentation is the initial step in a three-step procedure. (Casey & Lecolinet, 1996):

Given a starting point in a document image:

1) Find the next character image.

2) Extract distinguishing attributes of the character image.

3) Find the member of a given symbol set whose attributes best match those of the input, and output its identity.

This sequence is repeated until no additional character images are found. A character is a pattern that resembles one of the symbols the system is designed to recognize. But to determine such a resemblance the pattern must be segmented from the document image. Casey & Lecolinet (Casey & Lecolinet, 1996) have classified the segmentation methods into three pure strategies based on how segmentation and classification interact in the OCR process. The elemental strategies are:

1) The classical approach, in which segments are identified based on "character-like" properties. This process of cutting up the image into meaningful components is given a special name, “dissection".

2) Recognition-based segmentation, in which the system searches the image for components that match classes in its alphabet.

3) Holistic methods, in which the system seeks to recognize words as a whole, thus, avoiding the need to segment into characters.

### 2.2.1 Dissection Techniques

By dissection is meant the decomposition of the image into a sequence of sub images using general properties of the valid characters such as height, width, separation from neighboring components, disposition along a baseline etc.

Dissection is an intelligent process in that an analysis of the image is carried out; however, classification into symbols is not involved at this point. The segmentation stage consisted of three steps:

1) Detection of the start of a character.

2) A decision to begin testing for the end of a character.

3) Detection of end-of-character.

### 2.2.2 Projection Analysis

It can serve for detection of white space between successive letters. The analysis of the projection of a line of print has been used as a basis for segmentation of non-recursive writing. When printed characters touch, or overlap horizontally, the projection often contains a minimum at the proper segmentation column (Casey & Lecolinet, 1996). A peak-to-valley function have been designed to improve this method. A minimum of the projection is located and the projection value noted. A vertical projection is less satisfactory for the slanted characters.

### 2.2.3 Splitting of Connected Components

Analysis of projections or bounding boxes offers an efficient way to segment no touching characters in hand- or machine-printed data. However, more detailed processing is necessary in order to separate joined characters reliably. The intersection of two characters can give rise to special image features. Consequently dissection methods have been developed to detect these features and to use them in splitting a character string image into subimages. Only image components failing certain dimensional tests are subjected to detailed examination. One approach consists in using recognition as a validation of the segmentation phase and re-segmenting in case of failure. A different approach, based on the concept of precognition, follow connected component analysis with a simple recognition logic whose role is not to label characters but rather to detect which components are likely to be single, connected or broken characters (Casey & Lecolinet, 1996).

### 2.2.4 Recognition Driven Segmentation

This approach also segment words into individual characters which are usually letters. It is quite different from dissection based approach. Here no feature-based dissection algorithm is employed. Rather, the image is divided systematically into many overlapping pieces without regard to content. These are classified as part of an attempt to find a coherent segmentation/recognition result. Letter segmentation is a by-product of letter recognition, which may itself be driven by contextual analysis. The main interest of this category of methods is that they bypass the segmentation problem: No complex “dissection" algorithm has to be built and recognition errors are basically due to failures in classification. The basic principle is to use a mobile window of variable width to provide sequences of tentative segmentations which are confirmed (or not) by character recognition. Multiple sequences are obtained from the input image by varying the window placement and size. Each sequence is assessed as a whole based on recognition results. In recognition-based techniques, recognition can be performed by following either a serial or a parallel optimization scheme. In the first case, recognition is done iteratively in a left-to-right scan of words, searching for a "satisfactory" recognition result. The parallel method proceeds in a more global way. It generates a lattice of all (or many) possible feature-to-letter combinations. The final decision is found by choosing an optimal path through the lattice (Casey & Lecolinet, 1996). Recognition-based segmentation consists of the following two steps:

1) Generation of segmentation hypotheses (e.g. windowing)

2) Choice of the best hypothesis (verification step)

### 2.2.5 Holistic Technique

Holistic techniques may be used to recognize words without segmentation. Holistic methods in essence revert to the classical approach with words as the alphabet to be read. Recognition consists of comparing a lexicon of word descriptions against a sequence of features obtained from an un-segmented word image. The detected features, shown below the images, include loops, oriented strokes, ascenders and descenders. A holistic process recognizes an entire word as a unit. A major drawback of this class of methods is that their use is usually restricted to a predefined lexicon:

Since they do not deal directly with letters but only with words, recognition is necessarily constrained to a specific lexicon of words. This point is especially critical when training on word samples is required: A training stage is thus mandatory to expand or modify the lexicon of possible words. This property makes this kind of method more suitable for applications where the lexicon is statically defined (and not likely to change), like check recognition. They can also be used for on-line recognition on a personal computer (or notepad), the recognition algorithm being then tuned to the writing of a specific user as well as to the particular vocabulary concerned. Finally, holistic methods, usually follow a two-step scheme:

1. The first step performs feature extraction.

2. The second step performs global recognition by comparing the representation of the unknown word with those of the references stored in the lexicon. (Chaudhuri & Pal, 1997)

### 2.2.6 Related work

Chaudhuri and Pal (Chaudhuri & Pal, 1997) have proposed projection profile based technique for Bangla and Hindi character segmentation. For character segmentation, the position of headline (dika) is noted and if an imaginary line drawn from any point on this line touches the middle and lower zone boundary without touching a black pixel then the boundary of a character is found. For kerned characters, piecewise line segmentation is attempted. Veena and Sinha (Bansal & Sinha, Segmentation of Touching Characters in Devanagari, 1998) have considered the problem of conjunct segmentation in the context of Devanagari script. The conjunct segmentation algorithm process takes the image of the conjunct and the co-ordinates of the enclosing box. The position of the vertical bar and pen width are also inputs to the algorithm. For extracting the second constituent character of the conjunct, the continuity of the collapsed horizontal projection is checked. The collapsed horizontal projection corresponding to a Devanagari character image has continuity. Bansal & Sinha (Bansal & Sinha, A complete OCR for printed Hindi text in Devanagari Script, 2001) have proposed a projection profile technique for character segmentation. Words are divided into top and bottom strip and then vertical projection is computed to

extract character/symbol and top modifiers. Collapsed Horizontal Projection is defined for the segmentation of conjuncts/touching characters and shadow characters. Ma & Doermann (Ma & Doermann, 2003) identified Hindi words and then segmented into individual characters using projection profile technique (isolating top modifiers, separating bottom modifiers, and extracting core characters). Composite characters are identified and further segmented based on the structural properties of the script and statistical information. The Collapsed Horizontal Projection Technique is adopted from Bansal & Sinha (2001) for conjunct segmentation. Veena Bansal and R. M. K. Sinha (Bansal & Sinha, Segmentation of touching and fused Devanagari Characters, 2002) presents a two pass algorithm for the segmentation and decomposition of Devanagari composite (touching and fused) characters/symbols into their constituent symbols. The proposed algorithm extensively uses structural properties of the script. In the first pass, words are segmented into easily separable characters/composite characters. Statistical information about the height and width of each separated box is used to hypothesize whether a character box is composite. In the second pass, the hypothesized composite characters are further segmented. For segmentation of composite characters, the continuity of collapsed horizontal projection is checked. Agrawal, Ma & Doermann (Agrawal, Ma, & Doermann, 2010) have generated the character glyphs from font files and passed them through the feature extraction routines. For each character segmented in the document image, feature extraction is performed. With the objective of grouping broken characters, segmenting conjuncts, and touching characters, the technique of font-model-based intelligent character segmentation and recognition was developed. For each word, connected component analysis is performed. Kompalli et al (Kompalli, Nayak, & Setlur, Challenges in OCR of Devanagari Documents, 2005) have proposed a projection profile based method for character segmentation from words. Words are separated into ascenders, core components, and descenders. Gradient features are used to classify segmented images into different classes: ascenders, descenders, and core components. Core components contain vowels, consonants, and frequently occurring conjuncts. Core components are pre-classified into four groups based on the presence of a vertical bar: no vertical bar, vertical bar at the center, right or at multiple locations. Four neural networks are used for classification within these groups. Due to ascender and core character separation, characters may be divided into multiple segments during OCR. Positional information from segmented images is used to reconstruct the original character. For recognition

of valid but not frequently occurring conjuncts, Kompalli et al (2005) have attempted to segment the conjunct characters into their constituent consonants and classify segmented images. For the segmentation valid but not frequently occurring conjuncts, authors have examined breaks and joins in the horizontal runs (HRUNS) of a candidate conjunct character and build a block adjacency graph (BAG). Adjacent blocks in the BAG are selected from left to right as segmentation hypothesis. Both left and right images obtained from each segmentation hypothesis are classified using conjunct/vowel classifiers. The segmentation hypothesis with highest confidence is accepted. Post processing is carried out using a lexicon with 4,291 entries generated from the Devanagari data set. Kumar (Kumar & Sengar, 2010) presents projection profile technique for printed Devanagari and Gurmukhi script character segmentation. Initially, horizontal histogram of segmented line is computed and the position of header line is located. This separates the word into top and bottom strip. Vertical projection histogram for each strip is computed for the segmentation of top modifiers and characters. In this paper conjuncts/fused characters are not considered. The results are for clean documents consisting no conjuncts/fused characters. A projection profile technique is proposed in (Dongre & Mankar, 2011) for the segmentation of Devanagari Text Image. To normalize the image against thickness of the character the input image is thinned. Then the vertical projection histogram is computed and the locations containing single white pixels are noted. These points are taken as the boundaries for individual characters. The proposed method skips the process of header line removal. In case of character segmentation, words are segmented into more symbols than actually present in the word.

Kompalli et al. (Kompalli, Setlur, & Govindaraju, Design and Comparison of Segmentation Driven and Recognition Driven Devanagari OCR, 2006) have extended their previous work (Kompalli et al, 2005) and two different approaches: segmentation driven and recognition driven segmentation are compared for OCR of machine printed, multi-font Devanagari text. Kompalli & Setlur (2006) have proposed recognition driven approach that combines classifier design with segmentation using the hypothesis and test paradigm. Word images are examined along horizontal runs (HRUNS) to build a Block Adjacency Graph (BAG). Given the BAG of a word, histogram analysis of block width is used to identify the longest blocks as headline (dika) and isolate ascenders from core components. Regression over the centroids of

these core connected components is used to determine a baseline for the word. It uses the classifier to obtain hypotheses for word segments like consonants, vowels, or consonant-ascenders. If the confidence of the classifier is below a threshold the algorithm attempts to segment the conjuncts, consonant-descenders and half-consonants. Thus, the classifier results are used to guide the further segmentation. Kompalli et al (Kompalli, Setlur, & Govindaraju, Devanagari OCR using a recognition driven segmentation framework and stochastic language models, 2009) have proposed a novel graph-based recognition driven segmentation methodology for Devanagari script OCR using hypothesize and test paradigm. This work is further improvement to their previous work (Kompalli et al, 2006). A BAG is constructed from a word image and ascenders, and core components are isolated. The core components can be isolated characters that does not need further segmentation or conjuncts and fused characters that may or may not have descenders. Multiple hypotheses are obtained for each composite character by considering all possible combinations of the generated primitive components and their classification scores. A stochastic model (describes the design of a Stochastic Finite Automata (SFSA) that outputs word recognition results based on the component hypotheses and n-gram statistics) for word recognition has been presented. It combines classifier scores, script composition rules, and character n-gram statistics. Post-processing tools such as word n-grams or sentence-level grammar models are applied to prune the top n choice results. They have not considered special diacritic marks like avagraha, udatta, anudatta, special consonants such as, punctuation and numerals. Symbols such asanusvara, visarga and the reph character often tend to be classified as noise.

For Nepali HTK OCR, (Shakya, Tuladhar, Pandey,& Bal, 2009) (Bal, 2009) the projection profile technique have been adopted for character segmentation. The process includes removal of headerline and upper modifiers and then applying Multi-factorial analysis technique which uses fuzzy factors like degree of similarity, thickness, middleness, cross counts etc. to segment basic characters. The method is able to segment isolated characters along with half and conjoined characters. For the classifier, Hidden Markov Model (HMM) from HTK toolkit is used. (Rupakheti & Bal, 2009) have adopted projection profile technique for Nepali Tesseract OCR. Headerline width is identified and then vertical projection histogram of word to be

segmented is computed. Then the histogram analysis is done to mark starting and ending boundary of character fragment by taking header line as a threshold value that qualifies the segment to be separated.

(Bishnu & Chaudhuri, 1999) have proposed a recursive contour following method for segmenting handwritten Bangla words into characters. Based on certain characteristics of Bangla writing styles, different zones across the height of the word are detected. These zones provide certain structural information about the constituent characters of the word. Recursive contour following solves the problem of overlap between successive characters. (Garain & Chaudhuri, 2002) have proposed a method for segmenting the touching characters in printed Bangla script. With a statistical study they noted that touching characters occur mostly at the middle of the middle zone, and hence certain suspected points of touching were found by inspecting the pixel patterns and their relative position with respect to the predicted middle zone. The geometric shape is cut at these points and the OCR scores are noted. The best score gives the desired result. Habib (Murtoza, 2005) have proposed a projection profile technique for Bangla Character Segmentation. The width of the headline is variable because of print style (font size). So sometimes headline cannot be removed clearly. Here two morphological operations: thinning and skeletonization has been tried to overcome this problem. These operations removes pixels and pixels remaining makeup the image skeleton. Character can be separated by using connected components which is considered as input of recognition step.

The Arabic OCR Framework proposed by Nazly and others (Sabbour & Shafait, 2013) takes raw Arabic script data as text files as input in training phase. The training part outputs a dataset of ligatures, where each ligature is described by a feature vector. Recognition which takes as input an image specified by the user. It uses the dataset of ligatures generated from the training part to convert the image into text. It contains versions of degraded text images which aim at measuring the robustness of a recognition system against possible image defects, such as, jitter, thresholding, elastic elongation, and sensitivity. The performance of system is reported to be 91% for Urdu clean text and 86% for Arabic clean text.

Most of the researchers have adopted projection profile technique for character segmentation. For Devanagari character segmentation, this technique includes two phases: preliminary segmentation segments words into basic charactersand compound characters/shadow characters/fused characters. In general, preliminary segmentation includes detection of headline and use of its reference to isolate ascenders, core components, and descenders. For segmentation of compound characters, Bansal & Sinha (1998, 2001, 2003), have proposed continuity checking of Collapsed Horizontal Projection. Kompalli et al (2005) have proposed graph analysis for compound character segmentation. (Ma & Doermann, 2003) have used Structural Properties and statistical information of script is for further segmentation of compound characters. Kompalli et al (2006, 2009) have proposed graph based recognition-driven character segmentation technique to overcome the problem regarding the compound character segmentation which is usually difficult using projection profile techniques. Various character segmentation approaches for Devanagari OCR are summarized in table 6. Literature shows that Devanagari as well as Bangla OCR system mostly have adopted projection profile techniques for character segmentation. The Arabic OCR by Nazly and others (Sabbour & Shafait, 2013) proposed ligature identification and recognition approach. Which eliminates the ligature segmentation step. [13]

# 3. REQUIREMENT ANALYSIS

## 3.1 Functional Requirements

The expected functions of this system are as follows:

- The system should be able to draw the image from the mouse gesture.
- The system should be able to predict the word equivalent to the drawn image in text format.

## 3.2 Characteristics Requirements

The characteristics features for this system are:

- **Accuracy:** The system should be as accurate as possible. It should predict the corresponding word from the drawn image correctly.
- **Performance:** The system is intended to respond extremely fast as the prediction process uses saved model.
- **Wide Ranged:** The system is intended to predict wide range of Nepali characters accurately.

## 3.3 Feasibility Study

The recognition of Nepali character is a “must need” in the context of the Nepalese society today. Firstly, the cost requirement to complete this project is not much high and is easily affordable. Also, the commercial application of the obtained product would be a wise deal for consumers also. Secondly, the time limitations provided by syllabus is enough for completion of this task. The hardware required for the project are also easily available and their assembly is challenging but not that big deal. Similarly, the software required for this project are self-made and available with us. Hence, this project is a feasible one and its output is also a good product for commercial implementation.

# 4. SYSTEM ARCHITECTURE AND METHODOLOGY

## 4.1 Block Diagram

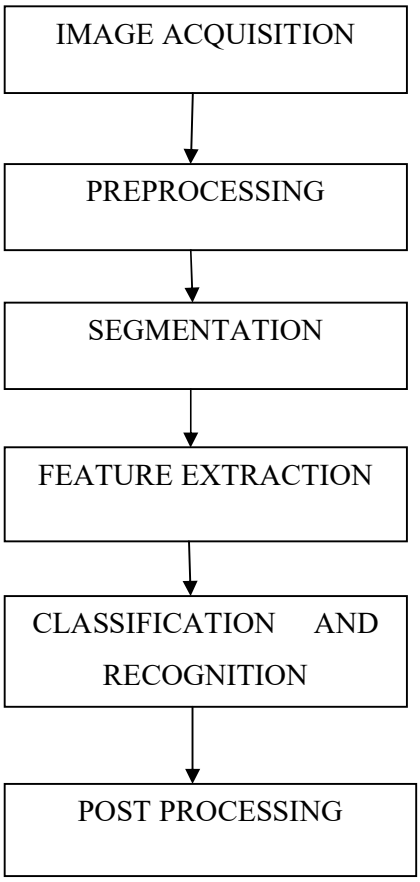


Figure 1: Block Diagram of our Recognition System

## 4.2 Data Flow diagram

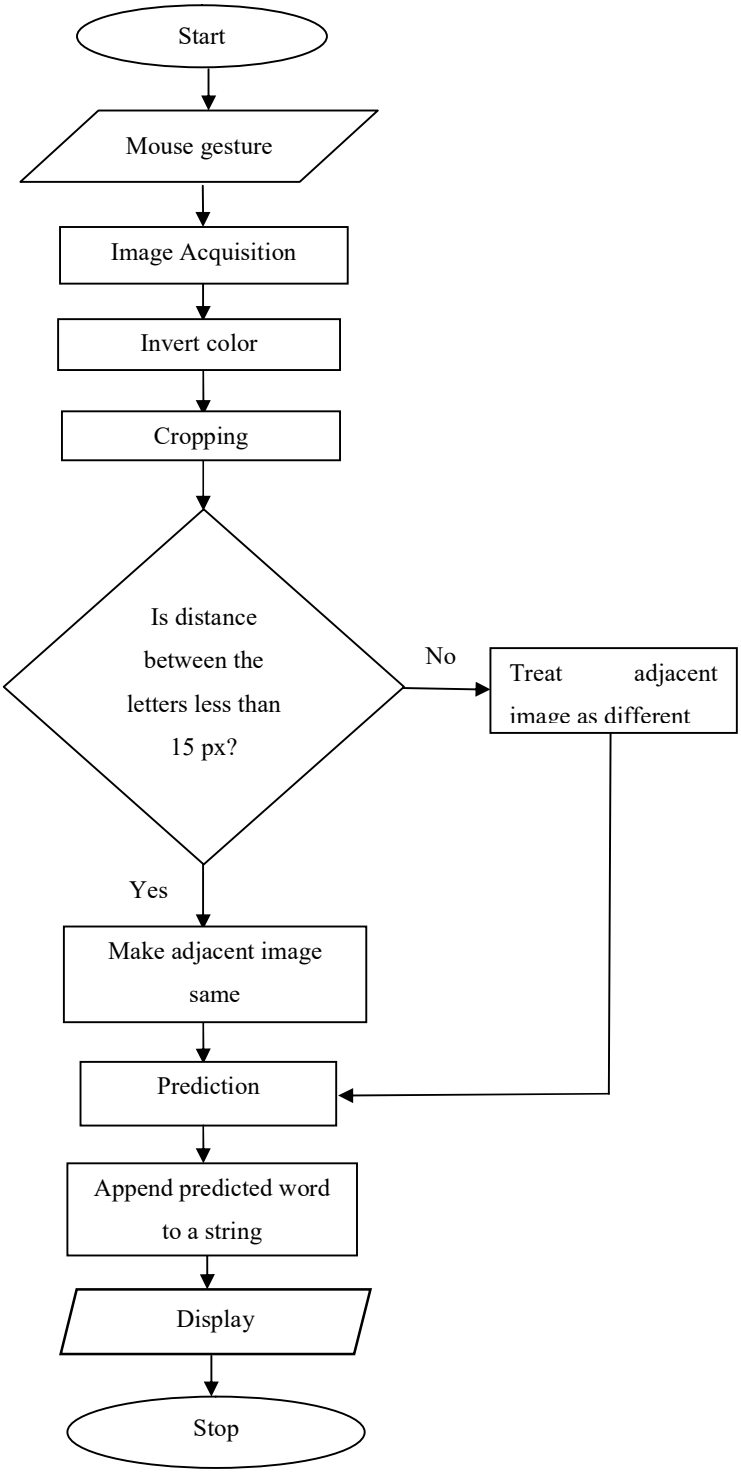


Figure 2: Data Flow Diagram

# 5. IMPLEMENTATION DETAILS

## 5.1 Hardware Requirements

### 5.1.1 M16175-h

M16175 is a low-cost CMOS optical sensor SOC for USB optical computer mouse. It provides an all in one solution including controller and sensor.

M16175 is based on algorithm which measures changes of sequential surface images and then determines the movement. It has 1000 DPI packages and its max motion speed can reach 24 inches per second and max acceleration can reach 20g.

M16175 is in a 12-pin optical DIP package and provides full mouse function including three buttons, X-Y motion and Z-axis wheel. It has a build-in LED driver and internal oscillator to minimize the external components.

M16175 is a USB interface SOC sensor and has completely USB HID version 1.1 compatibility. It is compatible with Microsoft 3D IntelliMouse.

### 5.1.2 Pin Configuration

#### 5.1.2.1 Pin Assignment

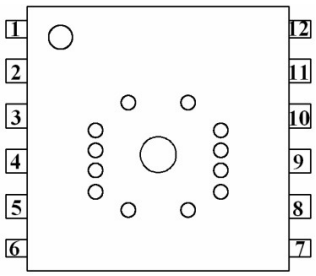


Figure 3: M16175-h Mouse Sensor (top view)

#### 5.1.2.2 Pin Description

| Pin | Name | Direction | Description |
|---|---|---|---|
| 1 | VDD33 | | Analog voltage reference |
| 2 | RB | Input | Right button input |
| 3 | MB | Input | Middle button input |
| 4 | LB | Input | Left button input |
| 5 | LED | Output | Led driver output |
| 6 | LEDVSS | | Led driver ground |
| 7 | DP | I/O | USB interface D+ |
| 8 | DM | I/O | USB interface D- |
| 9 | ZB | Input | Z axis inputB |
| 10 | ZA | Input | Z axis inputA |
| 11 | VDD | | Power supply,4.5V~5.5V |
| 12 | GND | | System ground |

Table 1: Pin Description of M16175-h Mouse Sensor

#### 5.1.2.3 System Architecture and operation

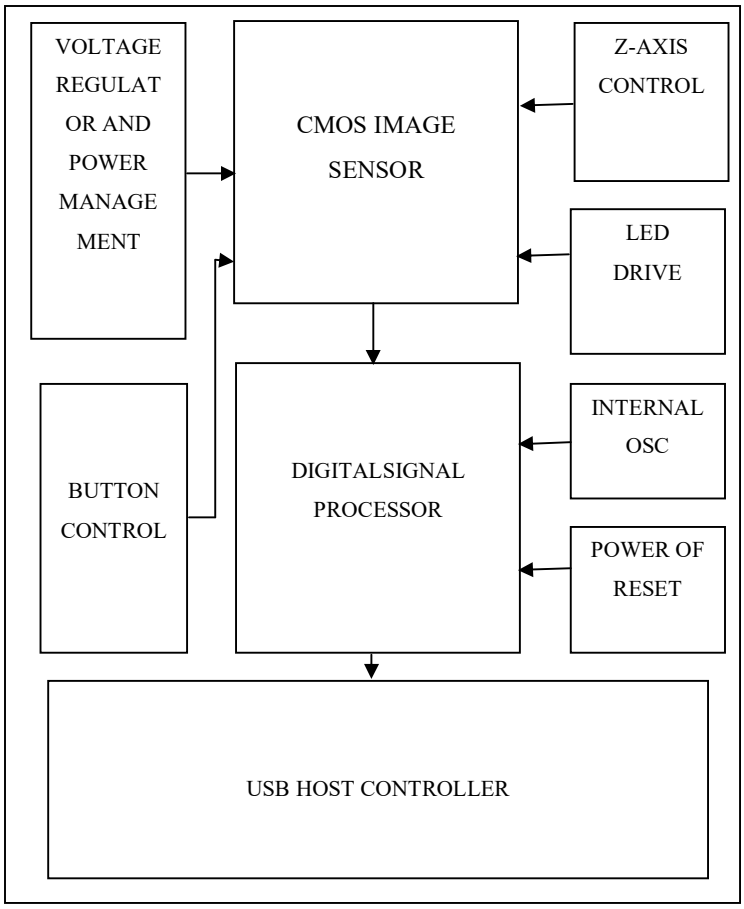


Figure 4: System Architecture of M16175

M16175 is a CMOS process optical sensor single chip which embedded USB controller and non-mechanical motion estimation module. It is in a 12-pin optical DIP package and provides full mouse function including three buttons, X-Y motion and Z axis wheel. It has a built-in LED driver and internal oscillator to minimize the external components.

### 5.1.3 USB communication

When a device is attached to the USB system, it gets assigned a number called its address. The address is uniquely used by that device while it is connected and, unlike the traditional system, this number is likely to be different to the address given to that device the last time it was used. Each device also contains a number of endpoints, which are a collection of sources and destinations for communications

between the host and device. Endpoints operate in simplex mode, meaning that they are either an input or output, but not both. For example, a simplistic model of a keyboard could have a keypad as output endpoint number 1, and the LED key lock display as receiving endpoint 1. All USB devices have one of each of their 16 possible input and output endpoints reserved as "zero end points". These are used for the auto-detection and configuration of the device when it is connected, and are the only accessible endpoints until this occurs. In addition each endpoint sets, upon connection, its own set of characteristic requirements concerning its requirements when accessing the bus.

The combination of the address, endpoint number and direction are what is used by the host and software to determine along which pipe data is travelling. A pipe is simply a data path between an endpoint and the associated portion of the controlling software, such as between the Keyboard LEDs and the BIOS routine which determines what LEDs should be lit. A special pipe is defined to connect to the zero endpoints, and is called the Default Control Pipe.

When the software requires data transfer to occur between itself and the USB, it sends a block of data called an I/O Request Packet (IRP) to the appropriate pipe, and the software is later notified when this request is completed successfully or terminated by error. Other than the presence of an IRP request, the pipe has no interaction with the USB. In the event of an error after three retry attempts, the IRP is cancelled and all further and outstanding IRPs to that pipe are ignored until the software responds to the error signal that is generated by sending an appropriate call to the USB. How exactly this is handled depends upon the type of device and the software.

As suggested by the name Universal Serial Bus, data transmission in the bus occurs in a serial form. The actual data is sent across the bus in packets. Each packet is a bundle of data along with information concerning the source, destination and length of the data, and also error detection information. Since each endpoint sets, during configuration, a limit to the size of the packet it can handle, an IRP may require several packets to be sent. Each of these packets should be the maximum possible size except for the final packet. The USB host has a built in mechanism so that the software can tell it when to expect full sized packets.

In the event that a less than maximum size packet is received earlier than expected, an error is assumed and the pipe is stalled with all IRPs being cancelled until the problem is dealt with by the controlling software. If an endpoint is busy, but no error has occurred, it responds with a special signal labelled NAK (Negative acknowledge), which tells the other end of the pipe to wait a while. How these conditions are handled depends on the type of device and the software.

Each packet is made up of a set of components called fields including the following:

• An eight bit "SYNC" synchronization field used by inputs to correct their timing for accepting data. Part of this field is a special symbol used to mark the start of a packet.

• The 8 bit Packet Identifier (PID) which uses 4 bits to determine the type, and hence format, of the packet data. The remaining 4 bits are a 1's complement of this, acting as check bits. Part of this field determines which of the four groups (token, data, handshake, and special) that the packet belongs to, and also specifies an input, output or setup instruction.

• An address field which gives the address of the function on the end of the pipe to be used.

• The 4 bit endpoint field, giving the appropriate endpoint which sends or receives the packet.

• A data field consisting of 0-1023 bytes.

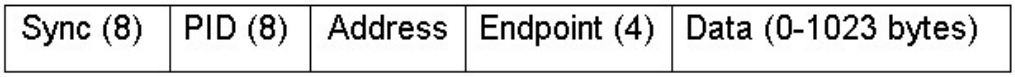


Figure 5: A Typical Data Packet

## 5.2 Software Requirements

### 5.2.1 Sublime Text

Sublime Text is a proprietary cross-platform source code editor with a Python application programming interface (API). It natively supports many programming languages and markup languages, and functions can be added by users with plugins, typically community-built and maintained under free-software licenses.

### 5.2.2 Jupyter Notebook

The Jupyter Notebook is an interactive computing environment that enables users to author notebook documents that includes live code, interactive widgets, plots, narrative text, equations, images, and video.

## 5.3 Recognition System

In this section, the prepared recognition system is described. A typical handwriting recognition system consists of preprocessing, segmentation, feature extraction, classification and recognition, and post processing stages. The schematic diagram of the proposed recognition system is shown as:

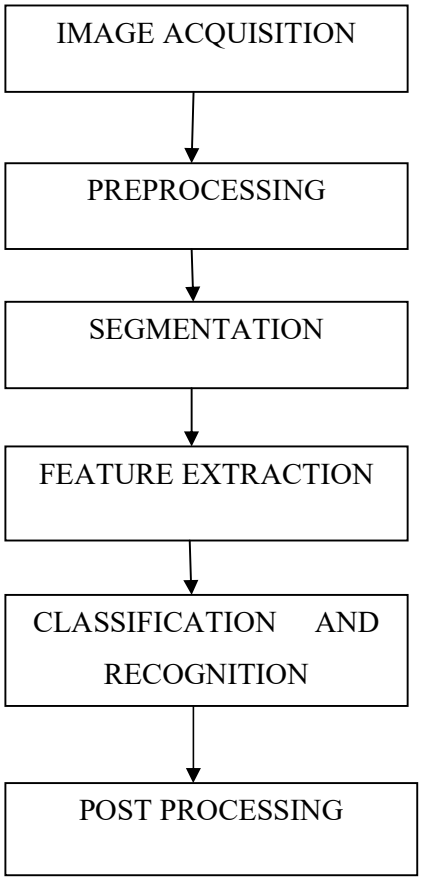


Figure 6: Recognition System

### 5.3.1 Image Acquisition

The recognition system acquires input image from the image drawn on a canvas program. This is image of specific format such as .jpeg, .png.

### 5.3.2 Image Preprocessing

The pre-processing is a series of operations performed on the scanned input image. Here scanning is referred to the process of acquisition of the character pixels from the input device. Pre-processing is the method essentially enhancing the image rendering it suitable for segmentation. The various tasks involved in image preprocessing are shown as:

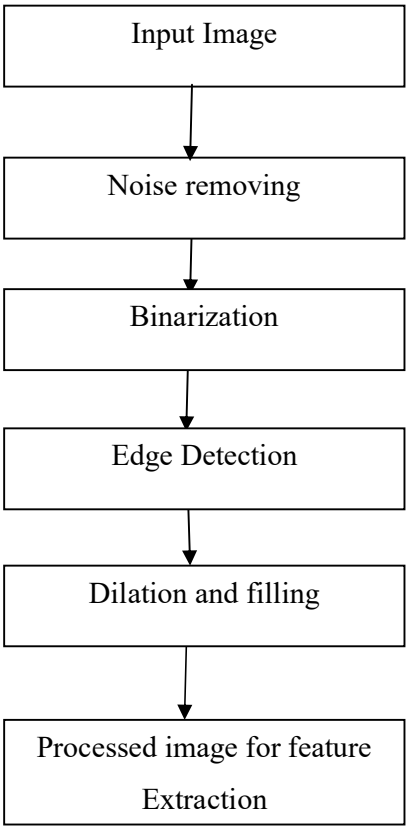


Figure 7: Image Preprocessing

### 5.3.3 Segmentation

In the segmentation stage, an image of sequence of characters is decomposed into sub-images of individual character. In the proposed system, 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 150x150 pixels for classification and recognition stage.

### 5.3.4 Feature Extraction Method

In this stage, the features of the characters that are crucial for classifying them at recognition stage are extracted. This is an important stage as its effective functioning improves the recognition rate and reduces the misclassification.

### 5.3.5 Classification and Recognition

The classification stage is the decision-making part of a recognition system and it uses the features extracted in the previous stage. In this stage, input is given to our model and based on the feature of the input image our model predicts the class of the input image. Every class in our model represents the distinct Devanagari letters. By matching the class, our model predicts the letter. As image may contain several letters within it, so image is first processed to distinguish whether it contain multiple letters or single letter. Based on the output, our model then forward image to two different sections. By using edge detecting algorithm image is divided into different images (equal to the number of letter) and each image is then converted into pixel size of 150*150 for further processing.

In order to fully understand about our model, let us know some basic structure how to it is build and what it is composed of. We are using CNNs as building block of our model. The reason for choosing CNN are several and the most important reason is that it uses much less memory than traditional FC network. CNNs are composed of 5 layers and they are Input layer, Convolutional layer, ReLU layer, pooling layer and fully connected layer. In FC networks, the inputs are depicted as vertical lines of neurons, basically vectors. Whether we process images or not we always have to tweak our data to switch to this configuration. However, when dealing with images, it helps to think about them as squares or neurons where a = each neuron represents a pixel value. Basically, CNNs keep images as they are and don't try to squeeze them in a vector. It can be illustrated by following diagram:

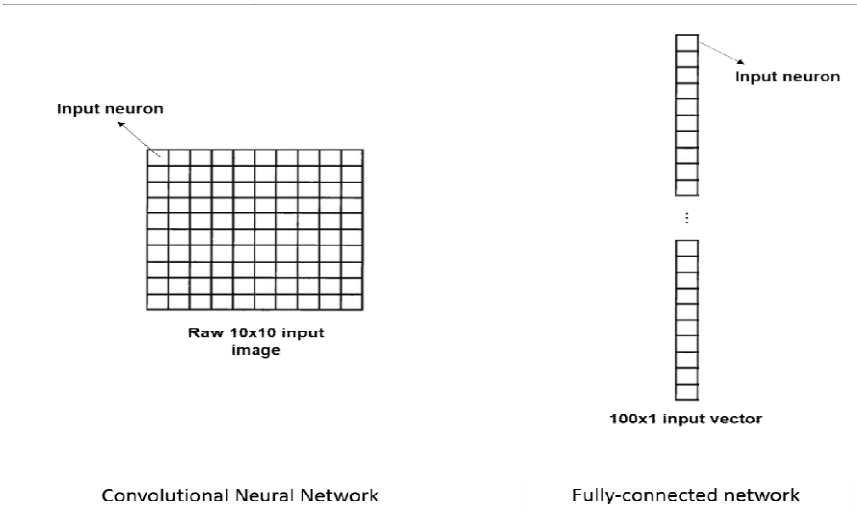


Figure 8: Convolutional Neural Network

Next comes the main component of a convent. In this layer, each pixel in the image is not connected with all neuron of hidden layer. It is rather connected to a patch (generally a small square region) of neurons in the previous layer as shown in figure below:

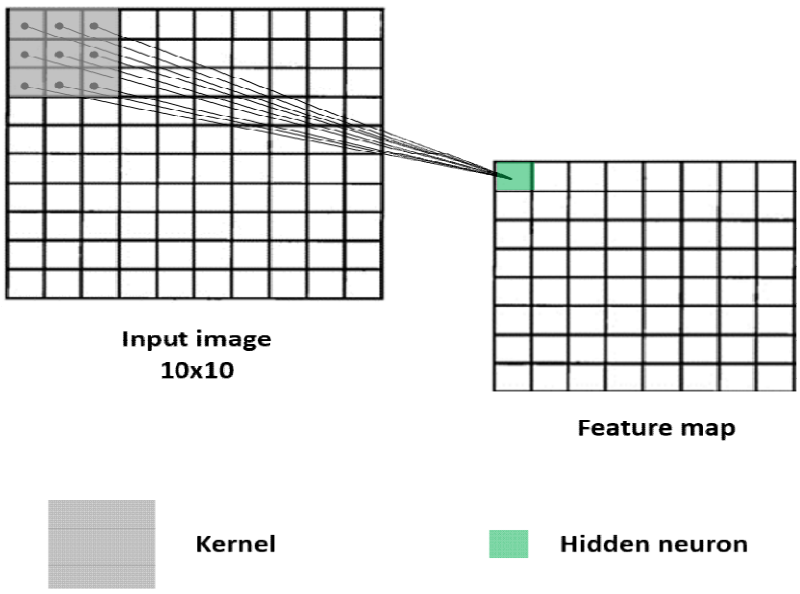


Figure 9: Feature Map

In this figure, the first neuron of the first hidden layer, which we also call a feature map, is connected to a patch of 3x3 pixels in the input. This hidden neuron depends

only on this small region and will ultimately, throughout the learning, capture its characteristics.

What does the value of this first hidden neuron represent? This is the result of a convolution between a weight matrix called the kernel (the little gray square) and a small region of same size in the image, called the receptive field.

The operation behind is very simple: it's an element-wise multiplication of two matrices: the 3x3 image region and the kernel of same size. The multiplications are then summed up into an output value. In this example, we have 9 multiplications that are summed into the first hidden neuron.

This neuron basically learns a visual pattern out of the receptive field. We can think of its value as a intensity that characterizes the presence or not of a feature in the image. And now other hidden layer is computed by shifting the kernel by a unit (or one stride) on the input image from left to right and applies the same convolution, again with the same filter and similarly other neurons are calculated by shifting the kernel by a unit. Now a question may arise that what does this CN operation really represent? The answer it generates a convolved output that responds to a visual element existing in the input. The output may be of reduced size, and it can be thought of as a condensed version of the input version. The kernel determines which feature the convolution is looking for. It plays the role of the feature detector. We can think of many filters that could detect edges, semi circles, corners, etc. In a typical CNN architecture, we won't have one single filter per convolution layer. Sometimes we'll have 10, 16 or 32. In this case we'll be performing as many convolutions per layer as the number of filters. The idea is to generate different feature maps, each locating a specific simple characteristic in the image. The more filters we have, the more intrinsic details we extract when training CNNs we won't be setting the filter weights manually. These values are learnt by the network automatically. In a typical fully connected network it learns via back propagation. Well, convolution networks do the same thing.

Instead of large weight matrices per layer, CNNs learn filter weights. In other words, this means that the network, when adjusting its weights (from random values) to

decrease the classification errors, comes up with the right filters that are suitable for characterizing the object we're interested in. This is a powerful idea that reverse-engineers the vision process.

Once the feature maps are extracted from the convolution layer, the next step is to move them to a ReLU layer. ReLU layers are usually coupled with convolutional layers. They usually work together. A ReLU layer applies an element wise ReLU function on the feature map. This basically sets all negative pixels to 0. The output of this operation is called a rectified feature map.

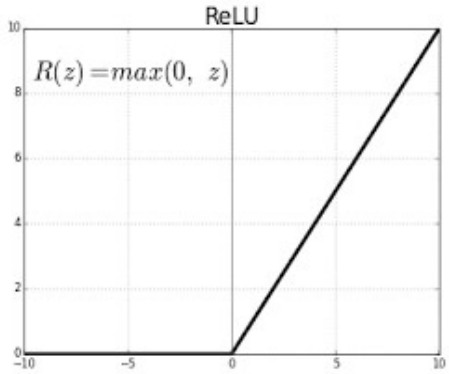


Figure 10: ReLU Function

The mainn advantages of ReLu is that they introduce a non-linearity in the network. In fact, all operations seen so far: convolutions, elementwise matrix multiplication and summation are linear. If we don't have a non linearity, we will end up with a linear model that will fail in the classification task.

The rectified feature maps now go throught a pooling layer. Pooling is a down-sampling operation that reduces the dimensionality of the feature map.

The most common pooling operation is max-pooling. It involves a small window of usally size 2x2 which slides by a stride of 2 over the rectified feature map and takes the largest element at each step.For example a 10x10 rectified feature map is converted to a 5x5 output.

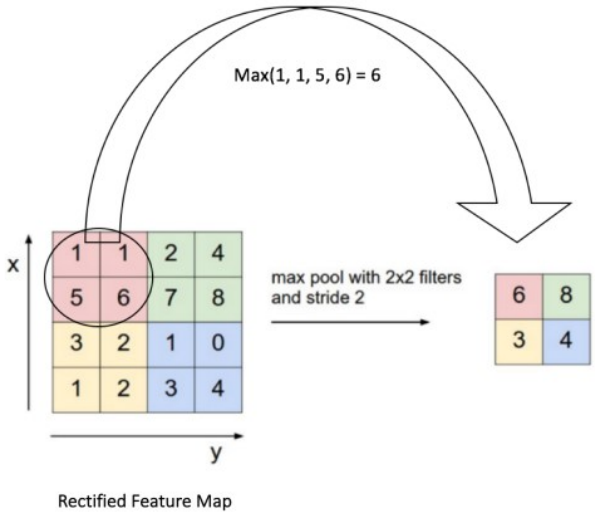


Figure 11: Max Pooling

It helps in reducing size of the rectified feature maps and the number of trainable parameters, thus controlling overfitting. It also condenses the feature maps by retaining the most important features and makes the network invariant to small transformations, distortions and translations in the input image (a small distortion in input will not change the output of Pooling – since we take the maximum value in a local neighborhood).

If a feature (an edge for instance) is detected, say on small portion of an image like the 2x2 red square above, we don't care what exact pixels made it appear. Instead, we pick from this portion the one with the largest value and assume that it's this pixel that summarizes the visual feature. This approach seems aggressive since it loses the spatial information. But the fact is, it's really efficient and works very well in practice. In fact you shouldn't have in mind a 4x4 image example like this. When max-pooling is applied to a relatively high resolution image, the main spatial information still remains. Only unimportant details vanish. This is why max pooling prevents overfitting. It makes the network concentrate on the most relevant information of the image.

CNNs also have a fully connected layer. The one we are used to see in typical FC nets. It usually comes at the end of the network where the last pooled layer is flattened into a vector that is then fully connected to the output layer which is the prediction

vector (its size is the number of classes). Fully connected layer(s) play the role of the classification task, whereas the previous layers act as the feature extractors. Fully connected layers take the condensed and localized results of convolutions, rectification and pooling and integrate them, combine them and perform the classification. Apart from classification, adding a fully-connected layer is also a way of learning non-linear combinations of these features. Most of the features from convolution and pooling layers may be good for the classification task, but combinations of those features might be even better.

# 6. RESULTS AND ANALYSIS

The theme requirement of any work is the reasonable output from it. So, to analyze whether the system has been developed as expected or not, various testing and analysis were performed which are described below.

## 6.1 Software Testing

The software for Nepali Character Recognition was tested at its various level of development process. The tests were performed at the image acquisition part(whether the correct image is being drawn as per the mouse gesture, at the image pre-processing part(whether the correct image is supplied to the prediction part), at the training phase(checking how well the accuracy of the training is?), and at the prediction part(checking the accuracy of prediction of drawn characters).

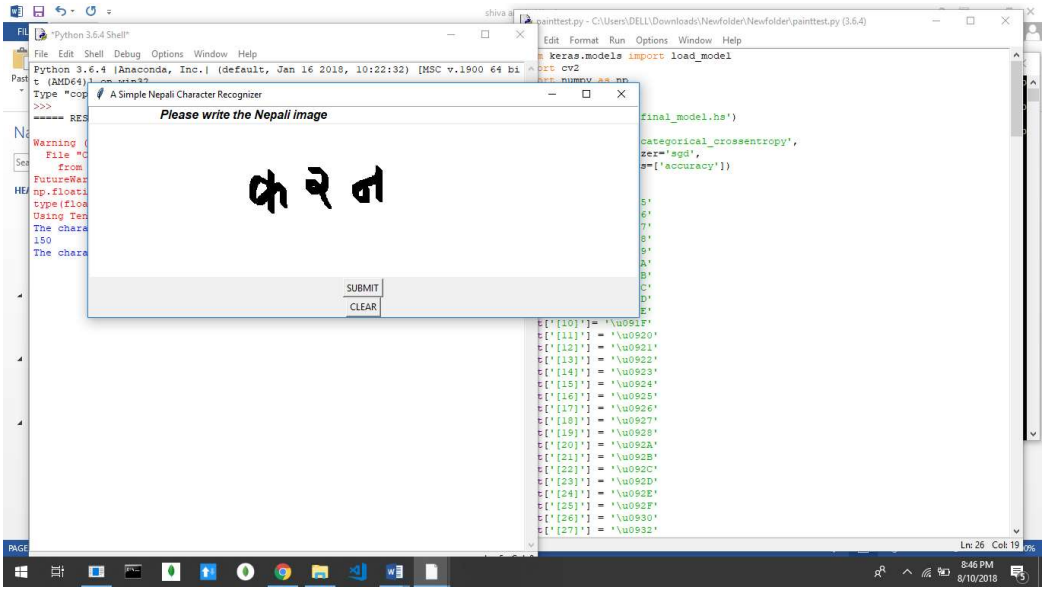


Figure 12: Desktop Application Receiving Input from User

## 6.2 Hardware Testing

Input to our system is provided from a mouse based sensor. The mouse pen is tested at different level of development process. The co-ordinates obtained from the mouse pen, is prime matter of our concern. The co-ordinates provided from the gestures were analyzed. The accuracy of mouse was tested via online application.

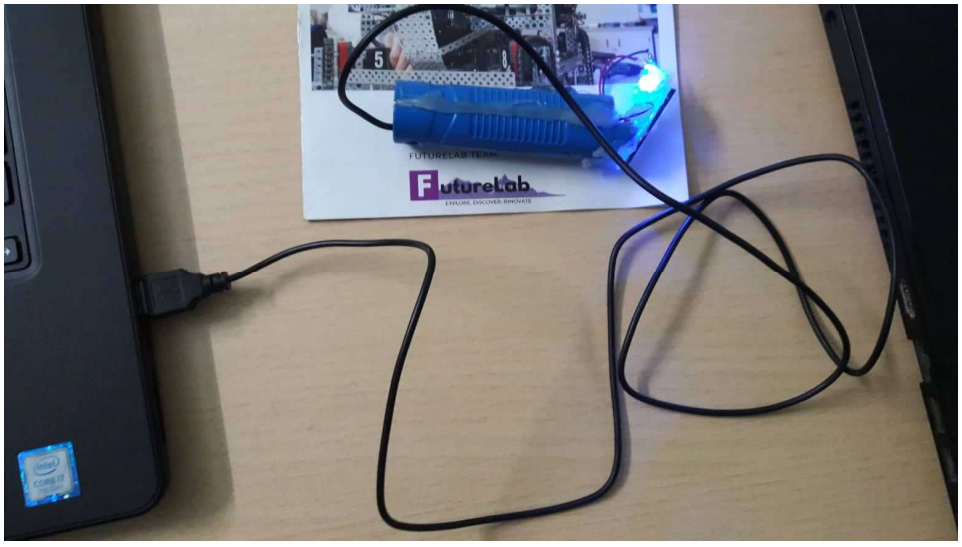


Figure 13: Digital Pen Testing

## 6.3 Output Inspection

After all the necessary integrations of hardware and different software modules, the proposed system is developed with proper accuracy and user friendly interface.

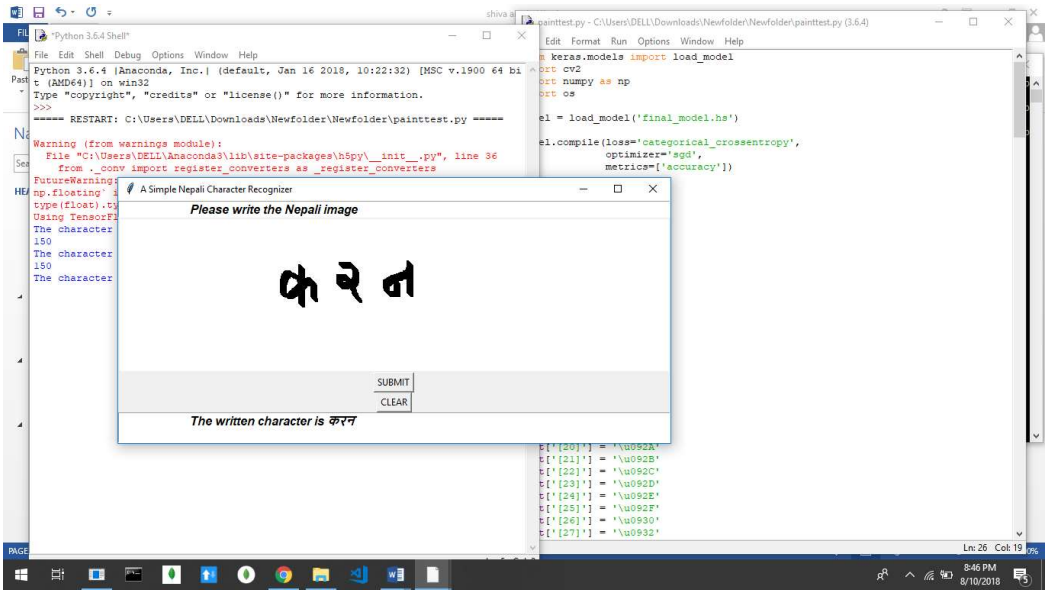


Figure 14: Desktop Application Predicting the written Characters

# 7. CONCLUSION AND FUTURE ENHANCEMENT

## 7.1 Conclusion

Input from the mouse like gestures will surely be a revolution in the modern world. Although, we haven't achieved best accuracy on Nepali character recognition yet, we are constantly optimizing the model for low loss and high accuracy. The accuracy of character recognition has reached to 91% which is still not good (till the date best accuracy that has been achieved is 99.4%) but it is sure to increase after some optimization of the training model.

## 7.2 Limitations

As nothing can be perfect, our system also has some limitation and are mentioned below:

- Due to the lack of enough variety of datasets, our system can recognize only letters of some definite shapes.
- Also because of lack of proper machine to train our Neural Network, we couldn't train all the letters (Nepali vowel and Nepali Barnamala) so our system can recognize only Nepali consonant and words made of them.
- We couldn't find the Unicode for the last 3 letters (ksha, tra , gya). So, we have given some alias value for them.

## 7.3 Future Enhancement

- We will train our model so that it will recognize all the Nepali words.
- We will improve the time require for the processing of input images.
- The pen for writing now is difficult to handle so we will improve the architect of the pen and make it easy for use.
- Also the GUI we prepared has less options and functions. So, it will also be updated to make it more user friendly.

## 8.REFERENCES


[1] Shodhganga.com, "Brief History Of Generations of Optical Character Recognition",[Online].Available: http://shodhganga.inflibnet.ac.in/bitstream/10603/4166/10/10_chapter%202.pdf

[2] http://tensorflow.org

[3] https://keras.io

[4] https://towardsdatascience.com/types-of-optimization-algorithms-used-in-neural-networks-and-ways-to-optimize-gradient-95ae5d39529f

[5] https://elitedatascience.com/keras-tutorial-deep-learning-in-python

[6] https://www.dlology.com/blog/quick-notes-on-how-to-choose-optimizer-in-keras/

[7] https://docs.python.org/3/library/tkinter.html

[8] https://en.wikipedia.org/wiki/Stochastic_gradient_descent

[9] https://medium.com/the-theory-of-everything/understanding-activation-functions-in-neural-networks-9491262884e0

[10] https://towardsdatascience.com/activation-functions-and-its-types-which-is-better-a9a5310cc8f

[11] https://keras.io/layers/pooling/

[12] http://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_watershed/py_water

shed.html

[13] https://www.researchgate.net/publication/29134_Segmentation_review

[14] https://www.mouseaccuracy.com