TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

PASHCHIMANCHAL CAMPUS, POKHARA

Lamachaur, Pokhara-16

[Subject Code::CT 707]

A MAJOR PROJECT PROPOSAL ON

"REAL TIME NEPALI HANDWRITING RECOGNITION"

Submitted by

Bikash Pandey	**[073BCT615]**
Biswash Tripathi	**[073BCT615]**
Netra Prasad Neupane	**[073BCT615]**
Upendra Dhamala	**[073BCT615]**

Submitted to

Department of Electronics and Computer Engineering

Pashchimanchal Campus

Poush,2076

REAL TIME NEPALI HANDWRITING RECOGNITION

Submitted by:

Bikash Pandey	**[073BCT615]**
Biswash Tripathi	**[073BCT617]**
Netra Prasad Neupane	**[073BCT626]**
Upendra Dhamala	**[073BCT648]**

Submitted to:

Department of Electronics and Computer Engineering

Pashchimanchal Campus

Lamachaur,Pokhara-16

Poush,2076

Abstract

The purpose of the Real Time Nepali Optical Character Recognition(NOCR) or Real Time Handwriting and printed text recognition project is the recognition of Nepali Language. It uses OCR approach for language recognition. For this work, we have to acquire the huge knowledge of literature of Nepali language and the challenges on its implementation. The segmentation and recognition of Nepali text image is relatively more difficult than Latin text image document due to its features like cursiveness, modifiers and header line. We have planned to apply hybrid approach (combination of holistic approach and dissection approach guided by recognition) in this NOCR project. NOCR is trained to recognize whole words, basic characters as well as compound one. This hybrid approach for NOCR possesses two phases. In 1st phase, segmentation of input text images into words followed by recognition of words is done using holistic approach on the basis of confidence of classification. If the confidence was found less than threshold, we have to go for phase 2. Phase 2 is for the poorly classified words in phase 1. In this phase, words are segmented into characters using projection profile which results into characters and compound characters. Then dissection or merging of segments may be applied if necessary for further recognition process. With these two recognition models the error in word segmentation is expected to be (TRIPATHI???) % and the error in word recognition is about (TRIPATHI ??) %.

Key Words:
Image Segmentation, Feature Extraction, CNN

TABLE OF CONTENTS

ABSTRACT……………………………………………………………………iii

LIST OF FIGURES…………………………………………………………vi

LIST OF TABLES…………………………………………………………..vi

CHAPTER 1: INTRODUCTION

1.1 Motivation……………………………………………………………..1

1.2 Overview...1

1.3 Problem Statement………………………………………………………2

1.4 Objectives…………………………………………………………………2

1.5 Feasibility Analysis

1.5.1 Technical Feasibility…………………………………………………3

1.5.2Economic Feasibility………………………………………………....3

1.5.3Social Feasibility……………………………………………………….3

1.6 System Requirements

1.6.1 Software Requirements………………………………………………3

1.6.2 Hardware Requirements……………………………………………..4

CHAPTER 2: LITERATURE REVIEW…………………………………5

CHAPTER 3: METHODOLOGY

3.1 Architecture……………………………………………………………7

3.1.1 Pre-processing………………………………………………………….8

3.1.2 Segmentation…………………………………………………………...8

3.1.3 Feature Extraction Techniques…………………………………………9

3.1.4 Recognition of characters using Neural Network………………………10

3.2 Implementation…………………………………………………………....10

3.2.1 Dataset collection and generation……………………………………....10

3.2.2 Algorithms for classification and recognition……………………………………10

3.2.3 Programming Language, Frameworks and Development Environment…………...11

CHAPTER 4: EPILOGUE

4.1 Expected Output…………………………………………………………………...13

4.2 Cost Estimation……………………………………………………………………13

4.3 Project Scheduling…………………………………………………………………14

REFERNCES……………………………………………………………………..15

LIST OF FIGURES

Figures **Page**

Figure 1……………………………………………………………………......... 7

Figure 2…………………………………………………………………….....8

Figure 3…………………………………………………………………….....9

Figure 4…………………………………………………………………….....10

LIST OF TABLES

Table **Page**

Table 1………………………………………………………………………….13

Table ….………………………………………………………………..................14

CHAPTER 1: INTRODUCTION

1.1 Motivation

We have come across many softwares and applications that do the task of scanning and detecting the texts in images and handwritten texts in English language, though there are not any significant tasks done in the devanagari scripts for Optical Character Recognition(OCR).

The task of manually entering the handwritten texts into the system is quite slow and cumbersome. In many places, specially the administrative ones, we see that the manpowers are assigned for entering tasks specially for Nepali texts. A lot of time is wasted when it comes to typing the texts specially the Devanagari texts. So, there is the utmost need of a software or an application which can automatically scan and detect the texts written in an image as well as the handwritten texts. Our project focuses on the development of a software tool that can efficiently do the task of devanagari script recognition with high accuracy.

1.2 Overview

Devanagari is the script used for writing many official languages in Nepal and India. The official language of Nepal is Nepali is written in Devanagari script. Other languages in Nepal like Tharu, Tamang, Gurung, Raute, Limbu, Rai etc. also use this script. In India languages such as Hindi, Marathi, Sindhi, Sanskrit, Konkani, Gujarati, Punjabi, and Bengali etc. use this script. More than 300 million people use Devanagari script for documentation in Nepal and central and northern parts of India [1].

The script has a complex composition of its constituent symbols. Devanagari has 13 vowels and 34 consonants along with 14 modifiers of vowels and of “rakar,” symbols. Apart from the vowels and consonants, there are compound (composite) characters in Devanagari script, which are formed by combining two or more basic characters. The shape of a compound (composite) character is usually more complex than its constituent characters. A vowel following a consonant may take a modified shape, which depending on the vowel is placed to the left, right, top, or bottom of the consonant, and are called modifiers or “matras.” Text, characters, and digits are written from left to right in Devanagari. There is no concept of upper or lowercase characters. It is a phonetic and syllabic script. As Devanagari is phonetic, words are written exactly as they are pronounced;syllabic means that text is written using consonants and vowels that together form syllables. The vowels in can be either independent or dependent.

The script uses modifiers for “nasalization” or aspiration of a vowel or a consonant. Every devanagari script has its own specified composition rules for combining vowels, consonants,and modifiers. Some of them can be combined with their type. A modifier can be attached to a vowel or to a consonant. Consonants may have a half form when they are combined with other consonants. Except for some characters, the half forms of consonants are the left part of original consonants with the right part removed. Some special combinations are where a new character or the half forms of consonants may appear in the lower half of the new composite forms. Another distinctive feature of Devanagari is the presence of a horizontal line on the top of all characters.This line is known as header line or “shirorekha”.The words can typically be divided into three strips: top, core, and bottom. The header line separates the top and core strips and a virtual base lines eparates the core and lowerstrips.The topstrip generally contains the top modifiers, and bottom strip contains lower modifiers. When two or more characters appear side by side to form a word in Devanagari, the header lines touch and generate a bigger header line.

The translation of handwritten or printed text image to editable digital text is defined as Optical Character Recognition (OCR). OCR, using computers, has been a research area for many decades[2]. OCR is a subfield of pattern recognition. It has practical applications in many fields such as searching, information retrieval, document storage, document library etc. It can be used for either offline/online handwritten text recognition or printed text recognition [3]. The work on automatic recognition of printed Devanagari script started in early 1970s. The efforts then were initiated by Sinha[4], at Indian Institute of Technology, Kanpur. In Nepal, efforts on the research and development of the Nepali OCR system is quite new which is as recent as 2006 initiated by Madan Puraskar Pustakalaya (MPP) under the PAN Localization Project.

1.3 Problem statement

Much research has already been done for developing OCR applicable to languages that follow the Devanagari script and especially for Hindi claiming to have as much as 93% performance accuracy rate in the character level recognition [5]. Similar achievements have been reported for the Bangla script and language which is similar to the Devanagari script in many respects [6, 7, 8, 9]. In this respect, the current work on the Nepali OCR system is focused towards making use of the already available methods and techniques of OCR for the Devanagari making slight modifications wherever necessary so as to recognize characters with as much accuracy as possible.

1.4 Objectives

The main objectives of the project are:

a) To convert the handwritten and text-printed devanagari characters from an image into digital form.
b) To convert the handwritten and text-printed devanagari characters from an real time scanning document into digital form.
c) To automate the task of manually entering devanagari texts into the system from a simple scan-image technique.

1.5 Feasibility Analysis

1.5.1 Technical Feasibility

Devanagari OCR system is a challenging task to bring out into reality. It's not as easy as that of English character recognition which has few number of characters and a dataset smaller as compared to that of Devanagari character dataset. The most challenging task is the preparation of dataset. As the required and complete dataset is not available, it is mandatory for us to prepare the dataset by ourselves. Apart from the dataset all other resources required are:

1. Python
2. CNN
3. Google Colab
4. ANN
5. HMM

All these resources are freely available and required technical skills are manageable.

1.5.2 Economic Feasibility

There is initial cost of training in the high capacity cloud but after that there are not such significance costs. So the project will be completed within limited budget and will be type of freemium type of product.

1.5.3 Social Feasibility

Development of Devanagari OCR system is really a positive part for every Nepali people. No longer we need to manually type everything by hand in our system. What we need to do is a simple scan to the image and we get our required Nepali text on our device. It will save both time and manpower.

1.6 System Requirements

For our project, following are the hardware and software requisites:

1.6.1 Software Requirements

i. Operating System
ii. Programming Language

- ❖ Python 3.7 with libraries

iii. TensorFlow

iv. Git and GitHub
v. Development Environment

1.6.2 Hardware Requirements

For the implementation and proper execution of our system, following hardware requirements are necessary:

- **Central Processing Unit(CPU)**-intel i7 8th Gen
- **RAM**-8 GB minimum,16 GB or higher is recommended
- **Graphics Processing Unit(GPU)**-NVIDIA GeForce GTX 1050 or higher

CHAPTER 2: LITERATURE REVIEW

According to the research articles major work has been done in offline character recognition rather than online. Due to its cursive nature and features like modifiers and header line Devanagari script has been always remained open and popular area of research. Moreover, Nepali Optical Character Recognition is a field where significant research hasn't been carried out. The first research report on Devanagari recognition was published in 1977[10].

The paper "Improving Nepali OCR Performance by Using Hybrid Recognition Approaches" by Nirajan Pant and Bal Krishna Bal[11] uses hybrid approach (holistic approach & dissection approach) where recognition is performed in two phase. In phase-one, text image document is segmented into words and recognition is performed. In phase-two, character level segmentation is performed over poorly recognized words by phase-one. And then character level recognition is performed. It uses large scale of dataset i.e. 150,000 words are extracted from approximately 2000 articles. It uses Random Forest(RF) learning algorithm.Feature extraction is done using HOG.Histogram Oriented Gradients (HOG)descriptors are used to define a feature vector of a words or characters. Line and word level segmentation is performed using Blob detection method. It presents compound characters and ligatures as single characters. In average,the accuracy of the word recognition module is 81.44%.

The paper "Deep Learning Based Large Scale Handwritten Devanagari Character Recognition" by S. Acharya, A. K. Pant and P. K. Gyawali [12] presents Devanagari character recognition approach using deep CNN(Convolutional Neural Network). Their dataset consists of 92 thousand images of 46 different classes of characters of Devanagari script segmented from handwritten documents. RELU(Rectified Linear Unit) is used as activation function. This paper also proposed different techniques to prevent overfitting such as Dataset increment & Dropout. The highest value of Testing Accuracy obtained is 0.982681 in 50 epoch of training.

The paper "Online Devanagari Isolated Character Recognition for the iPhone Using Hidden Markov Models"by Abhimanyu Kumar and Samit Bhattacharya [13], presents a novel scheme, which is implemented on the iPhone, for the recognition of online handwritten basic isolated characters of the Devanagari scriptHe manual study of various characters was done and 42 stroke classes were created. In this paper, A stroke based recognition approach has been designed where strokes are recognized using Hidden Markov Models (HMM). One HMM is constructed for each stroke class. A second stage of classification has been designed and is used for recognition of characters using stroke classification results along with look up tables.

The paper "Optimized and Efficient Feature Extraction Method for Devanagari Handwritten Character Recognition" by S. Ansari and U. Sutar, presents three main tasks of general handwritten recognition such as Image segmentation, Feature Extraction and Classification [14]. In this paper different feature extraction methods are discussed and presented related with Devanagari script and proposed efficient and optimized extraction method with their comparative analysis. It mainly focuses on accuracy of the recognition system which is based on feature extraction phase, type of features and size of features. The hybrid efficient, faster and optimized feature vector is used which is a combination of geometrical features, regional features, distance transform and gradient features.

The paper "Combining Multiple FeatureExtraction Techniques for Handwritten Devanagari Character Recognition" by Sandhya Arora, Debotosh Bhattacharjee, described an OCR for Handwritten Devnagari Characters. They used four feature extraction methods like, intersection, shadow feature, chain code histogram and straight line fitting features. Shadow features are calculated for character image while intersection features, chain code histogram features and line fitting features are calculated by dividing the character image into various segments. A training set of 3332 samples and test set of 1568 samples are formed. They used a dataset of 4900 samples & achieved overall recognition rate as 92.80% [15].

In [16], Authors S Prajapati, SR Joshi, A Maharjan, B Balami describe the performance of Nepali Script OCR using Tesseract and ANN. A dataset of 69 Nepali fonts with the 2,484 character samples of consonants was used in the study. With Tesseract, the overall accuracy of 96% was obtained in the training phase and 69% in the testing phase. Similarly, with ANN, an accuracy of 98% was obtained in training phase and 81% in testing.

A paper by Sharma and Bhattarai in 2017 has shown a high character recognition accuracy using Convolutional Neural Networks. However, upon analysis of their confusion matrix, we found that they represented the character ‘?’ as ‘?.’ (a combination of two characters ‘?’ and ‘.’), which resulted in a high rate of error for that character, especially since 70% of their dataset was generated artificially[17].

‘

CHAPTER 3: METHODOLOGY

3.1 Architecture

The process of optically recognizing a character of a text usually begins with an image of the document which is generally obtained by scanning or via camera captures. The images should be preprocessed at first. Preprocessing includes steps like noise removal, binarization and skew detection. At binarization, image is converted into a gray scaled image and then to binary image (has only two possible values for each pixel) from a colored image. Then process of correcting skewness is done. Document may have various blocks of text, images and tables together, hence finding these blocks is necessary. Further the words are segmented into characters to be processed, recognized and classified.

A classifier is used over these extracted features which classify them into different output labels. We can use SVM if features can be linearly separated. Neural Networks, Deep Learning, Random Forest etc. if features are not linearly separable and are complex. The classifiers convert the input into UNICODE labels.

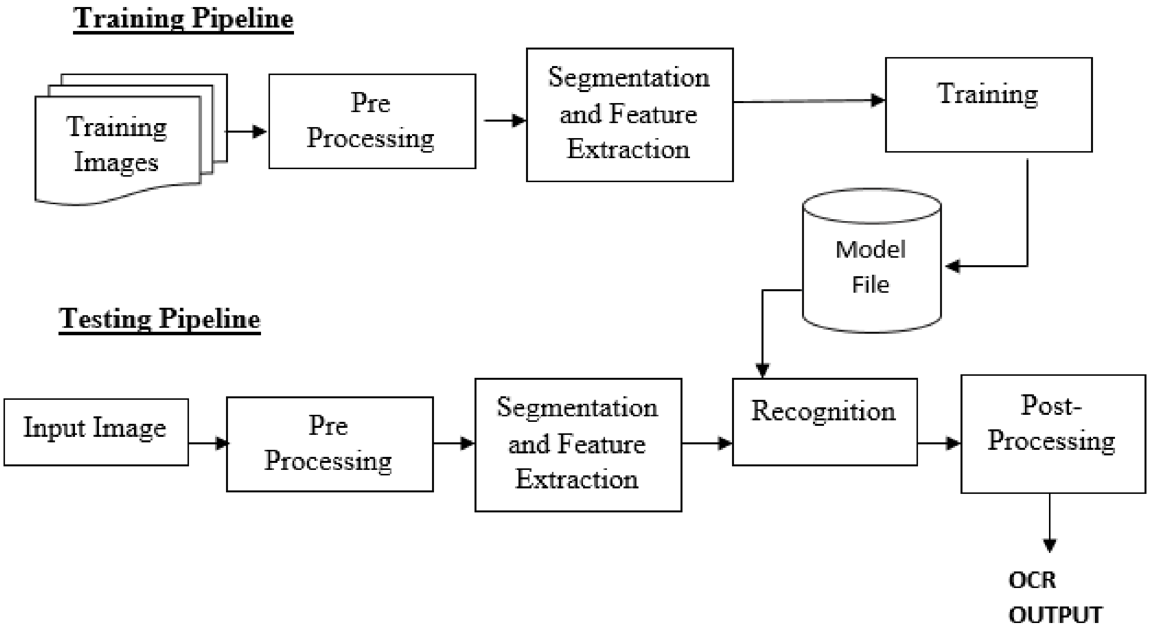


Figure 1: Training and Testing pipeline of a CNN Model [25]

In Figure 1, A post processing unit is generally used in OCR to improve the errors in classifying by the classifier.

3.1.1 Pre-processing:

Pre-processing involves modification of the input image in such a way that it is suitable for the next step. I.K Sethi and B.Chatterjee [10] use structural approach where they tried to find the relative positions of the horizontal segment,vertical segment and slants of that character. To overcome the issue of skew error in recognition problem, Chaudhari and Pal [18]] introduced a header line detection technique. To complement this, a mathematical morphology based fast and script independent technique was proposed by Chanda and Das [19]. In this technique, the skew is removed followed by the separation of the word into individual characters.

3.1.2 Segmentation

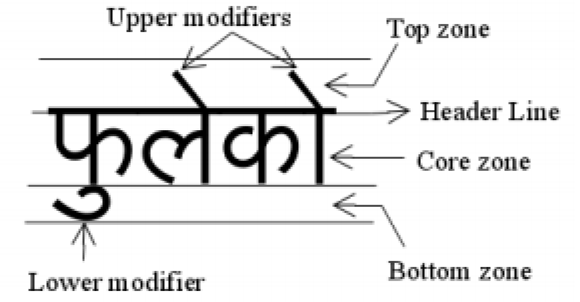
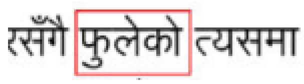


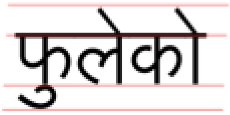
Figure2: Different zones of Nepali word [20]

Many approaches for character segmentation have been tried. Here we have tried to implement.. Most OCR systems are per-character OCR. Nepali words are tied together by headerline (dika). So it is difficult to segment and for that Nazly Sabbour and Faisal Shafait have proposed to use compound characters and ligatures as a single characters in Arabic and Urdu[21].That can be implemented for our language. Illustration of segmentation process is done by:
a. Algorithm to split multiple lines text
b. Algorithm to split physically disconnected words in a line.
c. Algorithm to remove top and bottom modifiers
d. Algorithm to split each character

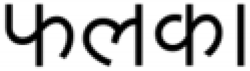
These algorithms are applied to binarized image of Nepali text.Illustratively it can be showns as:



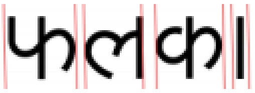
a. A line is separated from the multiple lines



b. A word is selected and its top zone, bottom zone and core zone is differentiated



c. Top and bottom modifiers are removed.Also header line is removed.

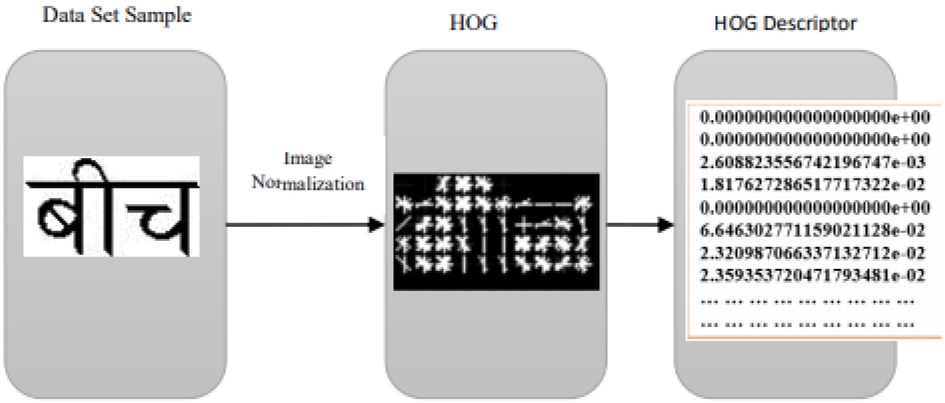


d. Individual characters are separated from it

Word segmentation using Blob detection has also been experimented[11]. S.Kompalli,S.Setlur,V.Govindaraju have proposed to segment along linear boundaries and use of classifiers to obtain preliminary hypothesis for each segment of the word[22].

3.1.3 Feature extraction Techniques

Ms.Seema A.Dongare and co. have used grid based approach which is the combination of image centroid zone and grid centroid zone of individual character or numerical image.It provides us with the feature vector of size 2X16 features[23].
HOG(Histogram of oriented gradients) descriptor is also good way which counts occurrences of gradient orientation in localized portions of an image.The HOG descriptor has a few key advantages over other descriptors. Since it operates on local cells, it is invariant to geometric and photometric transformations, except for object orientation[24].For HOG descriptor 'Normalization' has to be done at first which classifies data in certain size or in the ratio of sample width and height.



```
0.000000000000000000e+00
0.000000000000000000e+00
2.608823556742196747e-03
1.817627286517717322e-02
0.000000000000000000e+00
6.646302771159021128e-02
2.320987066337132712e-02
2.359353720471793481e-02
... ... ... ... ... ... ... ... ...
... ... ... ... ... ... ... ... ...
```

Figure3: Feature extraction using HOG descriptor

3.1.4 Recognition of characters using Neural Network

A variety of classifiers and techniques like ANN, HMM(Hidden Markov Model),fuzzy rules[27],etc had been used by researchers for the purpose of classification and recognition in the past. But here we have planned to use CNN (Convolutional Neural Network) which is more effective learning algorithm. CNN it is transfer learning algorithm and has generalization ability on unseen data. It learns more features about data in the image classification and has ability to handle overfitting. It is more accurate in comparison to Logistic regression, Decision trees, Support Vector Machine. Convolutional networks were inspired by biological processes and uses relatively little pre-processing compared to other image classification algorithms**.**

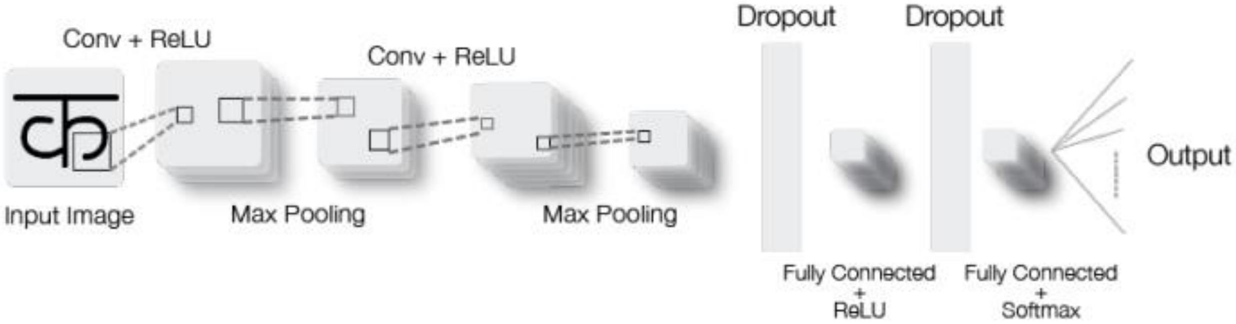


Figure4: Working Principle of CNN

3.2 Implementation

3.2.1 Dataset collection and generation

Here Nepali images of characters, words,compound characters images dataset is required. Up to this point, there is not well organized dataset that is available at public domain to use in our research works. We have found some datasets but some of they are incomplete or not labeled properly.So first of all we need training dataset and testing dataset [26].Apart from manually collecting datasets over a long period of time. So Aadash Neupane has created Nepali character dataset using semi-clustering approach which minimizes effort and time[20]. We have planned to ask Nepali handwriting text from our friends as well as method of artificial data synthesis as mentioned above.Madan Puraskar Pustakalaya, Nepal has been collaborating to develop Nepali character dataset.

Devanagari Handwritten Character Dataset (DHCD) is available, which was created as a part of research at a University in Nepal. There are three different categories in the dataset and the samples in the dataset are collected from 40 people. They are vowels, consonants and numerals.

3.2.2 Algorithms for classification and recognition

There are many machine learning classification algorithms that are used for recognition and classification. Here are some of the algorithms will use these algorithms to check the performance.

- **KNeighborsClassifier:**
  Supervised and unsupervised neighbor based learning method functionality is provided by sklearn.neighbors.Supervised and unsupervised neighbor based learning method functionality is provided by sklearn.neighbors.The idea behind the nearest neighbor algorithm is to search and find a preset number of samples having less distance from the new point and using these to predict the label. Here the number of samples, that is k, is user-defined value.

  Using this algorithm on a sample dataset from each class, we will calculate the test score, train score and the time taken by the algorithm to compute the results

.

- **RandomForestClassifier:**
  In this algorithm, each tree is built from a sample drawn with replacement from the training set. The split selected is the best one among a random subset of features. Due to this, the bias of the forest usually increases but its variance also decreases because of averaging and hence almost compensating for the increase in bias.
- **DecisionTreeClassifier:**
  Decision Trees are a method of doing supervised learning, generally used for classification and regression. The aim of this algorithm is to predict the target value by learning decision rules inferred from the data features. Trees are easy to understand, interpret and visualize. The data preparation is not that much as compared to another method, which needs their input data to be normalized, blanks removed, etc.
- **Gaussian Naïve Bayes (GaussianNB):**
  NB is a supervised learning method. It is based on using Bayes theorem with the assumption of naïve of conditional independence.

3.2.3 Programming Language, Frameworks and Development Environment

3.2.3.1 Python

We are planning to use python as our main programming language. The program components consist of preprocessing of the data, creating a classification model and a neural network. Python is really a powerful tool that has support for a lot of libraries and extensions. For example, one of the libraries used in this project is Numpy which provides support for high-level math functions on matrices and multidimensional arrays.

3.2.3.2 Library

Fastai,Numpy,Tensorflow,Scikit-learn,keras are some of the libraries we will be using throughout the project.

- Keras:It is a python API used for the high-level neural network. It uses Tensorflow as its base and is written in python. It enables fast experimentation. It is a user-friendly and modular library that allows models to be understood as a graph or a sequence. It helps in minimizing the number of actions required by the user for use cases that are common. The error feedback mechanism provided by Keras is an added plus.

- Tensorflow: It is an open source framework that consists of tools integrated into a flexible ecosystem, libraries, and other resources which are useful for developers to easily create and deploy machine learning based applications. It provides abstraction at multiple levels and allows the creation and training of models using high-level API Keras.
- Tensorflow: It is an open source framework that consists of tools integrated into a flexible ecosystem, libraries, and other resources which are useful for developers to easily create and deploy machine learning based applications. It provides abstraction at multiple levels and allows the creation and training of models using high-level API Keras.
- Fastai:The fastai library simplifies training fast and accurate neural nets using modern best practices.The library is based on research into deep learning best practices and includes "out of the box" support for vision, text, tabular, and collab (collaborative filtering) models.

3.2.3.3 Development Environments

- Google Colab :Google Colab, also known as Colaboratory, is a Jupyter environment provided and supported by Google with the option to work with CPUs, GPUs and even TPUs. It's like any other Jupyter notebook where we can code in Python and write descriptions as markdown along with all the other Jupyter features and a lot more.Google Colab comes with collaboration backed in the product It also runs on Google servers and you don't need to install anything. Moreover, the notebooks are saved to your Google Drive account.
- Jupyter Notebook :Jupyter Notebook. Jupyter Notebook (formerly IPython Notebooks) is a web-based interactive computational environment for creating Jupyter notebook documents.JupyterLab is flexible: configure and arrange the user interface to support a wide range of workflows in data science, scientific computing, and machine learning.

CHAPTER 4: EPILOGUE

4.1 Expected Output

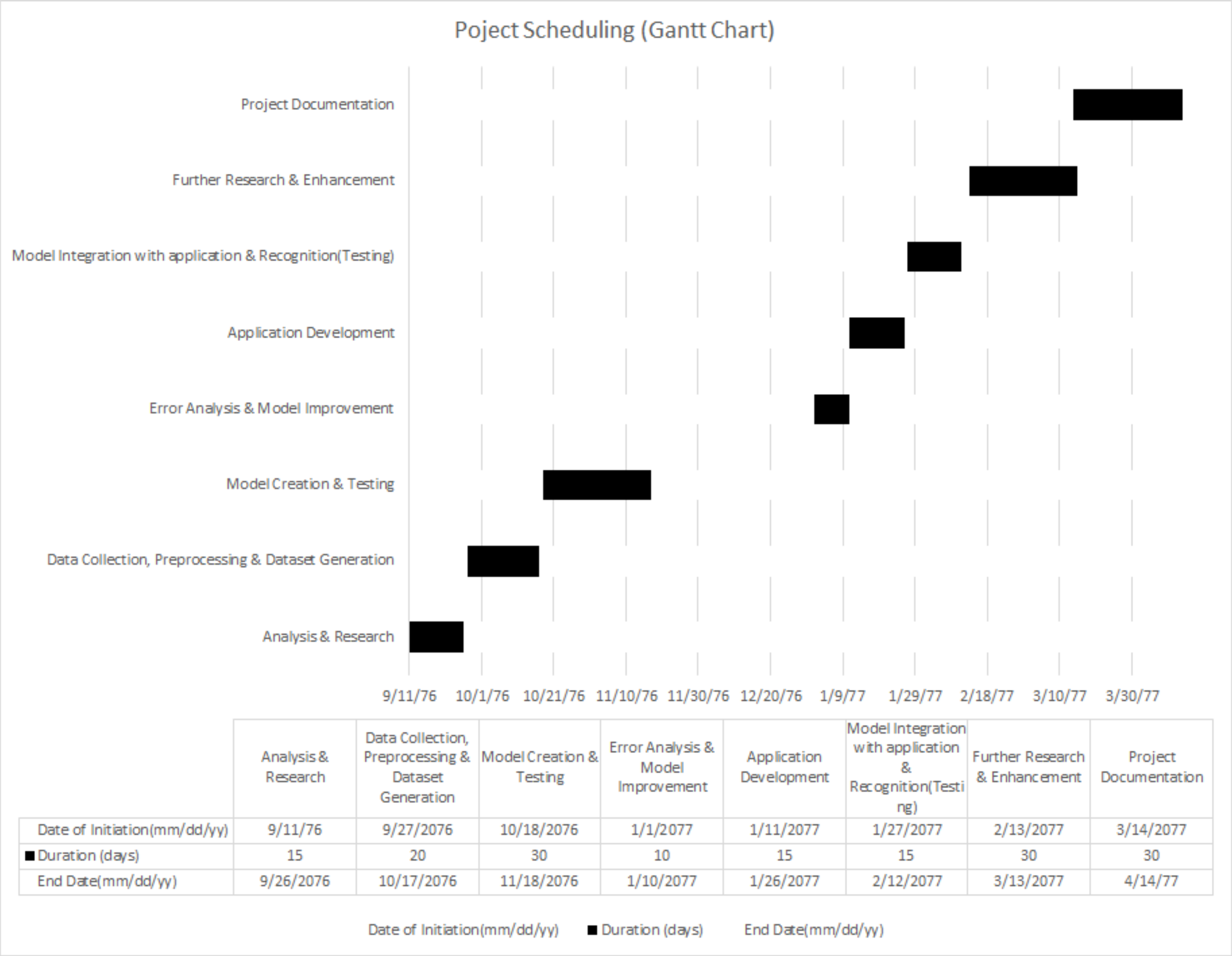
By the end of the project, we would be able to develop an application that would scan handwritten and printed text in stored images and scanning image and output the texts in the digital and editable format on user interface.

4.2 Cost Estimation

SN	Item	Cost
1	Research	2,500
2	Training on Cloud	3,500
3	Dataset generation	5,000
4	Miscellaneous	4,000
Total		15,000

Table 1: Approximate budget for the project

4.3 Project Scheduling



	Analysis & Research	Data Collection, Preprocessing & Dataset Generation	Model Creation & Testing	Error Analysis & Model Improvement	Application Development	Model Integration with application & Recognition(Testing)	Further Research & Enhancement	Project Documentation
Date of Initiation(mm/dd/yy)	9/11/76	9/27/2076	10/18/2076	1/1/2077	1/11/2077	1/27/2077	2/13/2077	3/14/2077
■ Duration (days)	15	20	30	10	15	15	30	30
End Date(mm/dd/yy)	9/26/2076	10/17/2076	11/18/2076	1/10/2077	1/26/2077	2/12/2077	3/13/2077	4/14/77

Table 2: Gantt Chart

REFERNCES

[1] R. M. K. Sinha, “A journey from Indian scripts processing to Indian language processing,” IEEE Ann. Hist. Comput., vol. 31, no. 1, pp. 8– 31, Jan./Mar. 2009.

[2]S. Prajapati, S.R. Joshi, A. Maharjan, B. Balami, “Evaluating Performance of Nepali Script OCR using Tesseract and Artificial Neural Network”, 2018 IEEE 3rd International Conference on Computing, Communication and Security(ICCCS), Kathmandu, pp. 1, 2018

[3]Y. Lu, “Machine Printed Character Segmentation – an Overview,” vol. 28, Pattern Recognition, 1995, pp. 67-80.

[4] R. M. K. Sinha, “A Syntactic pattern analysis system and its application to Devanagari script recognition,” Ph.D. Thesis, Dept. Elect. Eng., Indian Institute of Technology, Kanpur, India, 1973.

[5] V. Bansal and R. Sinha, "A complete OCR for printed Hindi text in Devanagari script," in Document Analysis and Recognition, 2001. Proceedings. Sixth International Conference, Seattle, WA, USA, 2001, pp. 800-804.

[6] S. Habib, H. Md. Abdul, and M. Khan, "A high performance domain specific ocr for Bangla script," Center for Research on Bangla Language Processing, Department of Computer Science and Engineering, BRAC University, 66 Mohakhali, Dhaka,Bangladesh, Working Paper 2007.

[7] B. Chaudhuri and U. Pal, "An OCR system to read two Indian language scripts: Bangla and Devanagari (Hindi)," in Proceedings of the Fourth International Conference on Document Analysis and Recognition, 1997, pp. 1011-1015.

[8] B. B. Chaudhuri, Digital Document Processing: Major Directions and Recent Advances (Advances in Pattern Recognition), P. S. Singh, Ed. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[9] U. Gerain and B. Chaudhuri, "Segmentation of touching characters in printed Devanagari and Bangla scripts using fuzzy multifactorial analysis," in Proceedings of the sixth international conference on Document Analysis and Recognition, Seattle,USA, 2001, pp. 805-809.

[10] I.K. Sethi and B.Chatterjee, “Machine Recognition of Constrained Hand printed Devanagari”, Pattern Recognition, Vol.9, 1977.

[11] Pant, Nirajan & Bal, Bal Krishna. (2016). Improving Nepali OCR performance by using hybrid recognition approaches. 1-6. 10.1109/IISA.2016.7785384.

[12] S. Acharya, A. K. Pant and P. K. Gyawali, "Deep learning based large scale handwritten Devanagari character recognition," *2015 9th International Conference on Software, Knowledge, Information Management and Applications (SKIMA)*, Kathmandu, 2015, pp. 1-6.

[13] Abhimanyu Kumar and Samit Bhattacharya, "Online Devanagari isolated character recognition for the iPhone using Hidden Markov Models," *2010 IEEE Students Technology Symposium (TechSym)*, Kharagpur, 2010, pp. 300-304.

[14] S. Ansari and U. Sutar, "Optimized and efficient feature extraction method for devanagari handwritten character recognition," *2015 International Conference on Information Processing (ICIP)*, Pune, 2015, pp. 11-15.

[15] Sandhya Arora, Debotosh Bhattacharjee "Combining Multiple FeatureExtraction Techniques for Handwritten Devanagari Character Recognition" 10 Colloquium and the Third ICIIS Kharagpur, INDIA December 8-10, 978-1-4244-2806-9/08 2008 IEEE

[16] S Prajapati, SR Joshi, A Maharjan, B Balami,"Evaluating Performance of Nepali Script OCR using Tesseract and Artificial Neural Network", 2018 IEEE 3rd International Conference on Computing …, 2018

[17] M. K. Sharma and B. Bhattarai, "Optical Character Recognition System for Nepali Language using ConvNet," Proceedings of the 9th International Conference on Machine Learning and Computing, pp. 184-189, 2017.

[18] B. B. Chaudhuri and U. Pal, "Skew angle detection of digitized Indian script documents," IEEE Trans. Pattern Anal. Mach. Intell., vol. 19, no. 2, pp. 182–186, Feb. 1997.

[19] A. K. Das and B. Chanda, "A fast algorithm for skew detection of document images using morphology," Int. J. Document Anal. Recognit., vol. 4, no. 2, pp. 109–114, 2001.

[20]Development of Nepali Character Database for Character Recognition based on Clustering,Aadesh Neupane.

[21]Sabbour, N., & Shafait, F. (2013, February). A segmentation-free approach to Arabic and Urdu OCR. In IS&T/SPIE Electronic Imaging (pp. 86580N86580N). International Society for Optics and Photonics.

[22]Kompalli, S., Setlur, S., & Govindaraju, V. (n.d.). Design and Comparison of Segmentation Driven and Recognition Driven Devanagari OCR. Second International Conference on Document Image Analysis for Libraries (DIAL'06). doi:10.1109/dial.2006.12.

[23]https://www.researchgate.net/publication/314373205_Handwritten_Devanagari_Character_Recognition_using_Neural_Network

[24] Dalal, N., & Triggs, B. (2005, June). Histograms of oriented gradients for human detection. In Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on (Vol. 1, pp. 886-893). IEEE.

[25]Kai Ding, Zhibin Liu, Lianwen Jin, Xinghua Zhu, "A Comparative study of GABOR feature and gradient feature for handwritten chinese character recognition", International Conference on Wavelet Analysis and Pattern Recognition, pp. 1182-1186, Beijing, China, 2- 4 Nov. 2007.

[26].https://www.researchgate.net/publication/291348945_Nepali_Optical_Character_Recognition_-_A_Hybrid_Approach

[27] V. Ganapathy and C. C. H. Lean, "Optical Character Recognition Program for Images of Printed Text using a Neural Network," 2006 IEEE International Conference on Industrial Technology, Mumbai, 2006, pp. 1171-1176.
doi: 10.1109/ICIT.2006.372591