ANALYSIS OF HANDWRITTEN DEVANAGARI CHARACTER RECOGNITION USING NEURAL NETWORKS AND DIFFERENT CLASSIFIERS

DISSERTATION

Submitted in partial fulfillment of the requirements for the degree

of

MASTERS OF SCIENCE

in

COMPUTER SCIENCE

by

Himanshu Rana

DEPARTMENT OF COMPUTER SCIENCE
D.S.B. CAMPUS, KUMAUN UNIVERSITY
NAINITAL-263001 (INDIA)
JUNE – 2024

DECLARATION

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Himanshu Rana

Roll No. 220120670003

CERTIFICATE

I hereby certify that the project work embodied in the thesis entitled "***Analysis of Handwritten Devanagari Character Recognition using Neural Networks and different Classifiers***" submitted by me to the Department of Computer Science, D.S.B. Campus, Kumaun University, Nainital for the award of degree of Master of Science in Computer Science, is a bonafide research work and authentic record of my own work carried out during the year 2024 under the supervision of Professor Ashish Mehta, H.O.D, Department of Computer Science, Kumaun University, Nainital. The matter presented in this thesis has not been submitted by me for the award of any other degree or diploma to this or any other University/Institution.

Himanshu Rana
M.Sc. Computer Science (Semester IV)
Roll No. 220120670003

Professor Ashish Mehta
(Supervisor)
D.S.B. Campus
Kumaun University
Nainital-263001

ACKNOWLEDGEMENT

First and foremost, I would like to express my deepest gratitude to my supervisor, Professor Ashish Mehta for his support, insightful guidance, and encouragement throughout the duration of this research. Their expertise and constructive feedback have been invaluable to the completion of this dissertation.

I am also profoundly grateful to the faculty and staff of the Department of Computer Science, DSB Campus, Kumaun University for providing a stimulating academic environment and the necessary resources that facilitated my research on the **Analysis of Handwritten Devanagari Character Recognition using Neural Networks and different Classifiers**.

I would like to extend my heartfelt thanks to my colleagues and friends for their camaraderie and for creating a collaborative atmosphere that was conducive to my research endeavors. Their moral support and intellectual discussions have enriched my academic journey.

I am deeply indebted to my family for their unwavering love, patience, and encouragement. Their belief in me has been a constant source of strength and motivation.

Thank you all for your invaluable contributions to this work.

Himanshu Rana

June, 2024

Kumaun University, Nainital

ABSTRACT

The project, "Analysis of Handwritten Devanagari Character Recognition using Neural Networks and different Classifiers", presents an analysis of using different classifiers like Random Forest Classifier, Support Vector Machine Classifier, etc., to classify and predict the Handwritten Devanagari Characters from images.

Although significant research has been made in full character recognition of Handwritten Devanagari Characters using Convolution neural networks for both feature extraction and classification, this report experiments different classifiers for classifying and predicting the handwritten characters while using CNN and DNN for feature extraction. The primary objective of this report is to see if the model can predict more accurately if trained on both full images and cropped images of handwritten devanagari characters.

The methodology involves a collection of full handwritten devanagari character images, and cropped handwritten devanagari character images. These images will undergo preprocessing and then using neural networks we will perform feature extraction. Then we will use those features for classification purpose using classifiers like Random Forest Classifier, Multi-layer Perceptron (MLP) Classifier, K-Nearest Neighbor(KNN) Classifier and Support Vector Machine (SVM) Classifier.

The outcomes of this project will be the predictions and comparisons of accuracies of different classifiers when the model is trained with full images of handwritten devanagari characters, cropped images of handwritten devanagari characters and mixed images of both, full images and cropped images of handwritten devanagari characters.

KEYWORDS: Handwritten Devanagari Character Recognition, Image processing, Feature extraction, Neural Networks, Classifiers, MLP, Random Forest, SVM, K-Nearest Neighbor.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1:

Chapter 4:

Chapter 5:

LIST OF TABLES

Chapter 5:

Chapter 1 - Overview

1.1 Introduction

Devanagari is a Northern Brahmic script related to many other South Asian scripts including Gujarati, Bengali, and Gurmukhi, and, more distantly, to a number of South-East Asian scripts including Thai, Balinese, and Baybayin. The script is used for over 120 spoken Indo-Aryan languages, including Hindi, Nepali, Marathi, Maithili, Awadhi, Newari and Bhojpuri. It is also used for writing Classical Sanskrit texts. Generally the orthography of the script reflects the pronunciation of the language [1].

The script is written from left to right. Letters hang from a headstroke, which is generally continuous throughout the length of the word, except when writing the letters jha, tha, dha, bha, a and ā, which all have a break in the headstroke. In handwriting, the headstroke is sometimes omitted [1].

Most of the Indian mythology is written in this script. Handwritten Devanagari character recognition has gained popularity over the years due to such importance of the script.

Handwriting character recognition is an art of identifying characters from handwritten images. Recognition of handwritten images is getting more and more attention due to its wide range of applications. Conversion of handwritten characters is significant for preserving several historical documents related to our history such as manuscripts, into machine editable form. This script (Devanagari) has various characteristics like complex shape, presence of modifiers, similarity between characters which makes recognition of Devanagari characters, a difficult task. Hence, this topic is one of the fascinating topics in the field of image processing and pattern recognition.

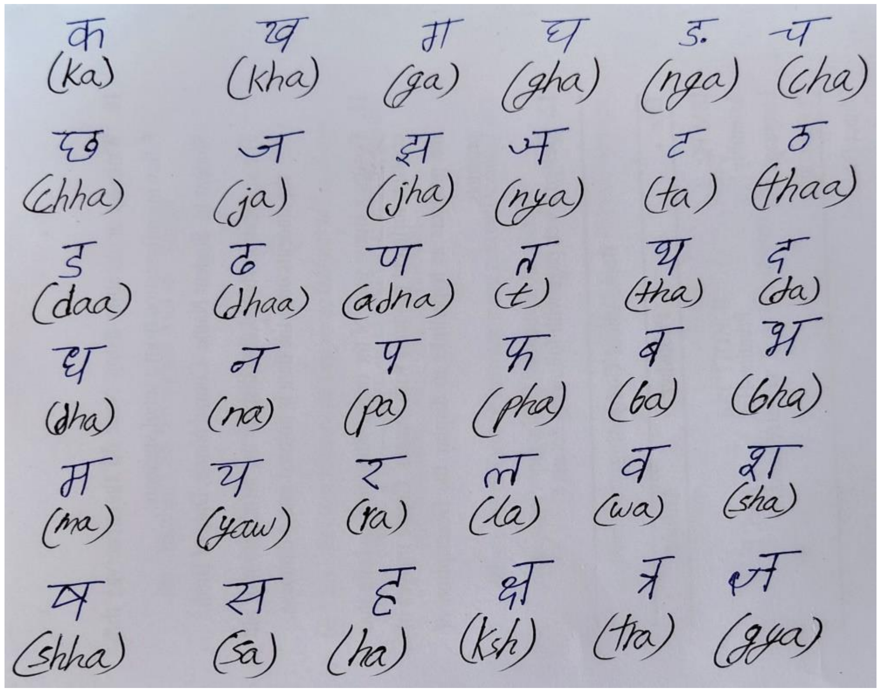

Fig. 1.1: Handwritten Devanagari Characters

Character recognition techniques associate a symbolic identity with the image of a character. These character images are preprocessed and then features are extracted from them. Features extracted from character encode the structural characteristics of character shape [2].

This field can be broadly divided into two parts [2]:

1) Online Character recognition:

   In online character recognition, characters are recognized at real time as soon as it is written. Online systems perform better than offline recognition as they have timing information and since they avoid the initial search step of locating the character. Online systems obtain the position of the pen as a function of time directly from the interface. This is usually done through pen-based interfaces where the writer writes with a special pen on an electronic tablet [2].

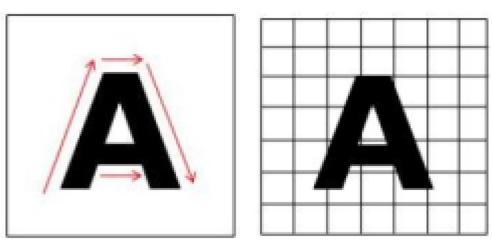

Fig. 1.2: (a) Online Character Recognition (b) Offline Character Recognition [2]

2) Offline character recognition:

   Offline character recognition can be classified further into following:

1. Printed characters recognition.
2. Handwritten character recognition.

Classification of handwritten character recognition is more difficult due to variation of handwriting styles, character shapes, thickness of character lines, image quality and many more.

Offline handwritten Character recognition is basically used for reading signatures on documents like cheques, verification documents, contract documents, digitalization of handwritten scrips or notes etc. Handwritten character recognition plays very important role in signature verification and for recognition of written texts filled by user in any document.

Some of the main reasons for difficulty in recognition of Handwritten Devanagari Characters are:-

- Many devanagari characters are similar looking like ङ(nga) and ड(daa), घ(gha) and ध(dha).
- Characters written by same writer can look different depending upon the type or size of pen or pencil used by them.
- The characters can be written anywhere on the paper. Can be slant or straight.
- Cropped images can cause more ambiguity in recognition.

1.2 Motivation

Till now a great amount of research has been done in the field of recognition of handwritten devanagari characters and recently an accuracy of approx. 99.64% [3] has been achieved using Convolution neural networks. However, all the research aims at predicting the character given the full image of the character. A few research also has been done on make a model to learn and predict the Devanagari character given the cropped images of the character [4]. The motivation here is to make a model that is trained on both full and cropped handwritten devanagari character images and see how well it can predict both a full handwritten devanagari character image and a cropped handwritten devanagari character image.

Chapter 2 - Review of Literature

Devanagari script, the foundation for languages like Hindi, Marathi, and Nepali, is crucial for information processing and communication in a vast region. Devanagari Character Recognition (DCR) technology plays a vital role in bridging the gap between physical Devanagari text and digital information. This review of literature explores the advancements in DCR, particularly focusing on the application of neural networks and various classifiers for achieving optimal recognition accuracy.

Handwritten Devanagari character recognition has seen significant advancements over the years, driven by the application of various techniques and methodologies. Pioneering research in Handwritten Devanagari Character Recognition (HDCR) during the 1970s primarily focused on analyzing the unique properties of the Devanagari script to achieve recognition.

In 2006, [5] discuss the characteristics of the some classification methods that have been successfully applied to handwritten Devanagari character recognition and results of SVM and ANNs classification method, applied on Handwritten Devanagari characters. After preprocessing the character image, they extracted shadow features, chain code histogram features, view based features and longest run features. These features are then fed to neural classifier and in support vector machine for classification. This paper concluded that, the result obtained for recognition of Devanagari characters show that reliable classification is possible using SVMs. The applied SVMs and ANNs classifiers on same feature data namely Shadow based, Chain code Histogram, Longest Run, and View based features. The SVM-based method described here for offline Devanagari can be easily extended to other Indian scripts and Handwritten Devanagari numerals also.

In 2012, [6] discuss the handwritten Devanagari script recognition system using neural network. Diagonal based feature extraction is used for extracting features of the handwritten Devanagari script. After that these feature of each character image is converted into chromosome bit string of length 378. In this paper, it is attempted to use the power of genetic algorithm to recognize the character. In recognition step using fitness function in which find the Chromosome difference between unknown character and Chromosome which are store in data base.

In 2012, [7] paper a recognition system for isolated Handwritten Devanagari Numerals has been proposed. The proposed system is based on the division of sample image into sub-blocks and then in each sub-block Strength of Gradient is accumulated in 8 standard directions in which Gradient Direction is decomposed resulting in a feature vector with dimensionality of 200. Support Vector Machine (SVM) is used for classification. Accuracy of 99.60% has been obtained by using standard dataset provided by ISI (Indian Statistical Institute) Kolkata.

In 2012, [8] paper the main aim of this research is to prepare a recognition system which can be used for the recognition of offline handwritten Hindi characters. For this proposed system Support Vector Machine is used as classifier and Diagonal feature extraction approach is used to extract features.

In 2013, [9] paper describes the development and implementation of a system comprising combination of several stages. Mainly Artificial Neural Network technique is used to designed to preprocess, segment and recognize devanagari characters. The system was designed, implemented, trained and found to exhibit an accuracy of 75.6% on noisy characters.

Compound character recognition of Devanagari script is one of the challenging tasks since the characters are complex in structure and can be modified by writing combination of two or more characters. These compound characters occurs 12 to15% in the Devanagari Script. The moment based techniques are being successfully applied to several image processing problems and represents a fundamental tool to generate feature descriptors where the Zernike moment technique has a rotation invariance property which found to be desirable for handwritten character recognition. In 2014, [10] paper discusses extraction of features from handwritten compound characters using Zernike moment feature descriptor and proposes SVM and k-NN based classification system. The overall recognition rate of proposed system using SVM and k-NN classifier is up to 98.37%, and 95.82% respectively.

In order to rapidly build an automatic and precise system for image recognition and categorization, deep learning is a vital technology. Handwritten character classification also gaining more attention due to its major contribution in automation and specially to develop applications for helping visually impaired people. In 2020, [11] the proposed work highlighting on fine-tuning approach and analysis of state-of-the-art Deep Convolutional Neural Network (DCNN) designed for Devanagari Handwritten characters classification. A two-stage VGG16 deep learning model is implemented to recognize the characters using two advanced adaptive gradient methods. A two-stage approach of deep learning is developed to enhance overall success of the proposed Devanagari Handwritten Character Recognition System (DHCRS).

Manuscripts serve as a wealth of knowledge for future generations and are a useful source of information for locating material from the Middle Ages. Ancient manuscripts can be found in handwritten form, thus they must be translated into digital form so that computing equipment can access them and additional indexing and search operations can be performed with ease. In 2023, [12] the Devanagari characters from the manuscripts is recognised using a CapsNet-based method. 33 fundamental characters, 3 conjuncts, and 12 modifiers make up the Devanagari alphabet. Due to spatial relationship, CapsNet is used to recognize the handwritten characters. The authors observed the best recognition accuracy of 94.6% was achieved to recognize the Devanagari characters using CapsNet.

In 2024, [3] paper, explore the use of Convolutional Neural Network for handwriting character recognition. Recent improvements in CNN technology have produced significant advances in Handwritten Character Recognition by learning discriminating qualities from enormous volumes of raw data. The CNN has a substantial benefit over traditional pattern recognition algorithms in terms that it can extract features, decrease data dimensionality, and classify all in one network structure. The final model for Devanagari consonants (and numerals) and vowels which have been integrated into this intended project achieve an accuracy of 99.54% and 99.64% respectively. These results show that the suggested CNN has superior classification performance, proving that it is a viable real-time solution for Handwritten Character Recognition.

Chapter 3 - Backgrounds

Background study of the proposed work by various authors has been done to study and understand the work done and methodology used by them so far on Handwritten Devanagari Characters Recognition. The methods, models and classification techniques used for classifying and predicting the images have been understood and then future scope and experiments of the project is decided based on the outcomes of these papers.

Following are the research papers used for the study:

1. Deepa Kumari, Yogita Borse, Hardika Gawde, Asra Masrat, “Handwritten Devanagari Character Recognition Using Convolutional Neural Network,” (2024) [3]:

   This paper explores the use of Convolutional Neural Network for handwriting character recognition. It also talks about the recent improvements in CNN technology that have produced significant advances in Handwritten Character Recognition by learning discriminating qualities from enormous volumes of raw data. The results in this paper show that the suggested CNN has superior classification performance, proving that it is a viable real-time solution for Handwritten Character Recognition.

2. Priyanka Sawant, Siddhant Gokule, “Handwritten Devanagari Character Recognition using Neural Networks,” (2018) [4]:

   In this paper, the report experiments different classifiers for classifying and predicting the handwritten characters while using CNN and DNN for feature extraction. The scope of this report has been widened by making the model to predict partial Devanagari characters while been trained on full characters and vice versa.

3. Aradhana A Malanker, Prof. Mitul M Patel, “Handwritten Devanagari Script Recognition: A Survey,” (2014) [2]:

   This paper summarizes the Character Recognition detail and surveys the various research works done in this field and also addresses the most important results reported so far and it is also tried to highlight the beneficial directions of the research till 2014.

4. Aditi Moudgil, Saravjeet Singh, Vinay Gautam, Shalli Rani, Syed Hassan Shah, "Handwritten devanagari manuscript characters recognition using CapsNet," (2023) [12]:

   In this study, the Devanagari characters from the manuscripts is recognised using a CapsNet-based method. Due to spatial relationship, CapsNet is used to recognize the handwritten characters. The authors observed the best recognition accuracy of 94.6% was achieved to recognize the Devanagari characters using CapsNet. This method was studied and tried however not implemented in this report.

Chapter 4 - Methodology

4.1. Problem Statement

Handwritten Devanagari Character Recognition using Convolutional neural networks and Dense neural networks for feature extraction and various multiclass classifiers classifying the characters. The dataset consisting of 36 unique Devanagari characters with 1000 images of each character is being used for training and testing the model. Predicting full or cropped handwritten devanagari images accurately when trained on both full images and cropped images of handwritten devanagari characters.

Following are the highlights of the report:

1) Training the model on full character image and making it predict the Devanagari character given full image of the character.
2) Training the model on cropped character image and making it predict the Devanagari character given full image of the character.
3) Training the model on full character image and making it predict the Devanagari character given cropped image of the character.
4) Training the model on both full images and cropped images of handwritten devanagari characters and making it predict the handwritten devanagari character given cropped and full image of the character.

4.2 Data Collection

The dataset used for this dissertation was downloaded from the UCI Machine Learning Repository

[https://archive.ics.uci.edu/dataset/389/devanagari+handwritten+character+dataset],

a publicly available collection of machine learning datasets.

The UCI dataset for characters is in folder Train_Data_Full. The cropped version of the images in Train_Data_Full is stored in Cropped_Images_For_Training folder. It also contains cropped images used for prediction in the Cropped_Images For_Testing folder which are predicted by the model after being trained on full character images. The folder Full_Cropped_Data_Train contains both full and cropped images of each character respectively.

4.3 Data preprocessing

Data preprocessing is the process of transforming raw data into a format that is suitable for analysis, machine learning, or other data processing procedures. It involves cleaning, transforming, and integrating data to improve its quality, accuracy, and consistency, making it ready for further processing or analysis.

“Data preprocessing is an important step in the data mining process. It refers to the cleaning, transforming, and integrating of data in order to make it ready for analysis. The goal of data preprocessing is to improve the quality of the data and to make it more suitable for the specific data mining task.” [13].

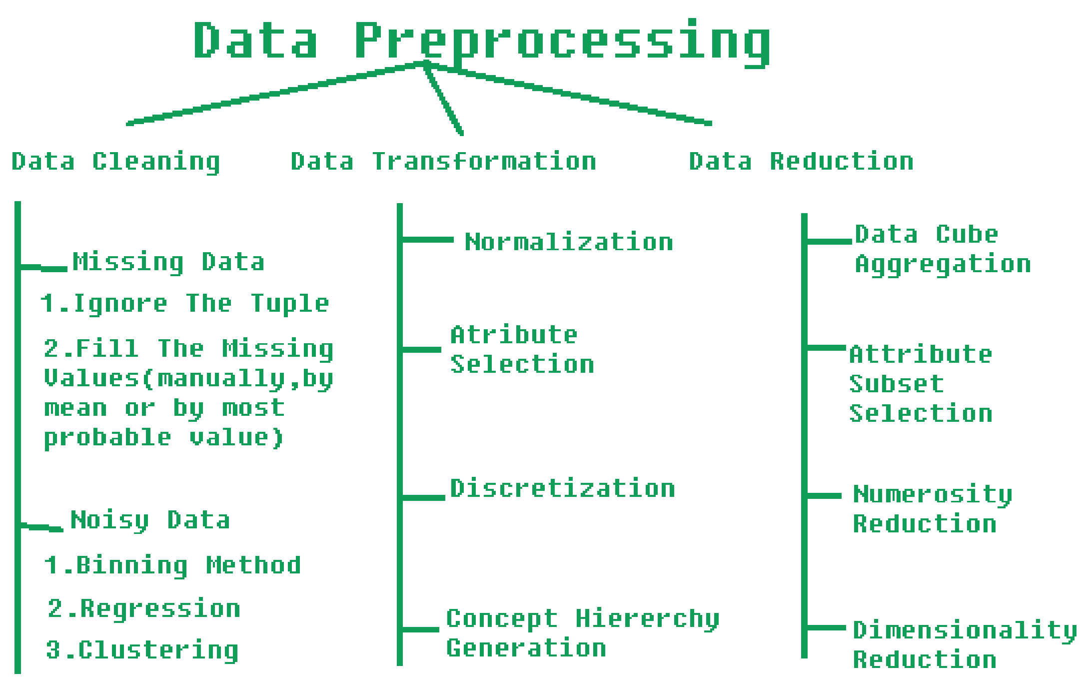

Fig. 4.1: Data Pre-Processing Steps [13]

Preprocessing is essential because in real-world data there may be inconsistencies, typos, missing entries, or even outliers that can skew the results. By preprocessing your data, you ensure a smooth and efficient analysis.

The dataset used from UCI contains training and test data, each having 36 Devanagari characters. For each character a separate directory is created having the name of the character in English_Hindi. Each directory contains 1000 images of the respective character. The target labels is not given separately.
Thus, data is preprocessed by extracting the character name from the directory name and storing it into a label array which is further used for training the model.

Each image is a 32x32 gray scale image which is firstly converted into an array and then flattened and stored in an image matrix to train the model.

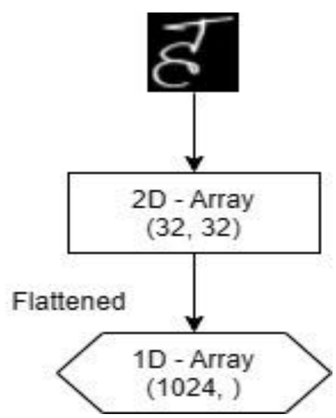

Fig. 4.2: Image Pre-processing

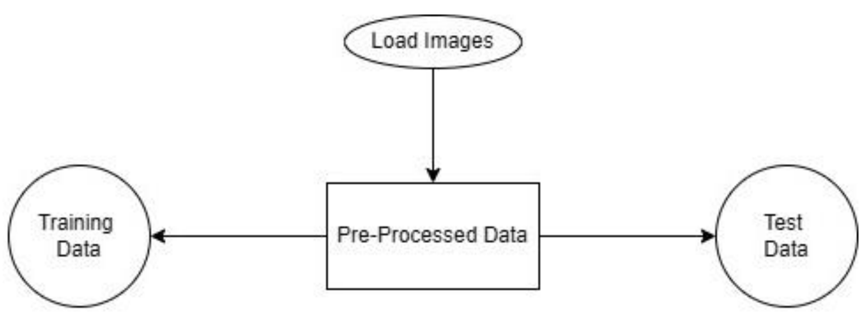

Fig. 4.3: Pre-Processing and Splitting data

4.4 Feature Extraction

Feature extraction is a crucial step in machine learning that involves transforming raw data into a format that machine learning algorithms can understand and process effectively.

Imagine a pile of unlabeled pen, pencil, and eraser — this is essentially raw data. Machine learning algorithms struggle to make sense of this jumble. Feature extraction acts like a sorting process, identifying and extracting the key characteristics or features of each pen, pencil, and eraser — their size, type, and material. This creates a more organized and meaningful dataset that the machine learning algorithm can then use to learn and make predictions.

Feature extraction is a process in machine learning and data analysis that involves identifying and extracting relevant features from raw data. These features are later used to create a more informative dataset, which can be further utilized for various tasks such as Classification, Prediction, Clustering, etc. [14].

Feature extraction aims to reduce data complexity (often known as "data dimensionality") while retaining as much relevant information as possible. This helps to improve the performance and efficiency of machine learning algorithms and simplify the analysis process. Feature extraction may involve the creation of new features ("feature engineering") and data manipulation to separate and simplify the use of meaningful features from irrelevant ones [14].

The features should be selected in such a way that it reduces the intra-class variability and increases the inter-class discriminability in the feature space [15].

Convolution Neural Networks (CNN) has been the best feature extraction neural network used so far by various authors. The CNN model used here is mainly composed of combination of Convolutional layers, Max-Pooling layers and Dense Neural networks layers (or Fully Connected Layers). "RELU" function is used as activation function for initial and hidden layers and "SOFTMAX" activation function is used as activation function for the output layer. The Adam optimizer is used in this model for computing the adaptive learning rates at each epoch.

Given below is the neural network architecture:

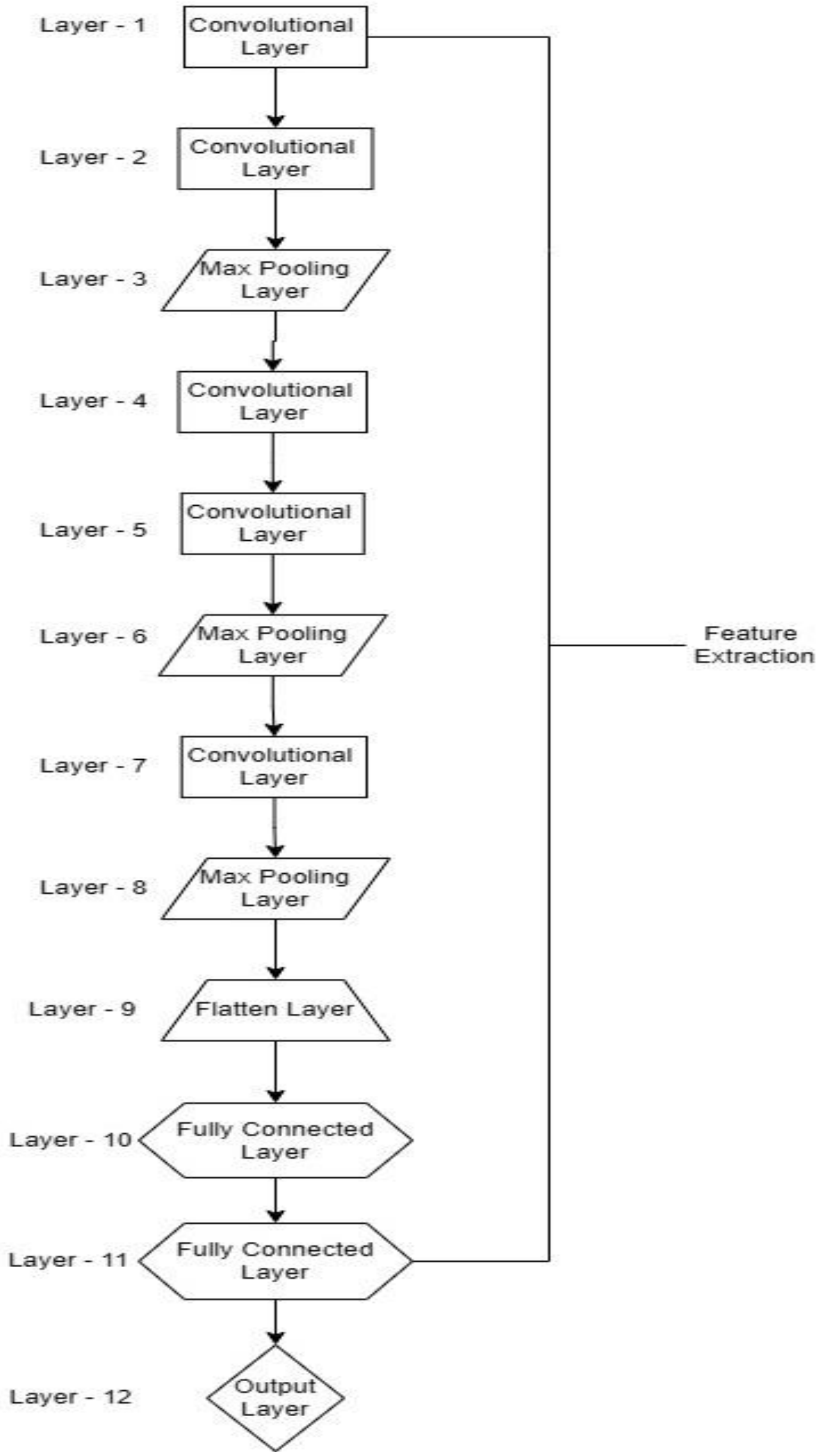

Fig. 4.4: Neural Network Architecture Flow Chart

```
Model: "sequential_5"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 L1Conv (Conv2D)             (None, 30, 30, 32)        320

 L2Conv (Conv2D)             (None, 28, 28, 32)        9248

 L3MaxPool (MaxPooling2D)    (None, 14, 14, 32)        0

 L4Conv (Conv2D)             (None, 12, 12, 64)        18496

 L5Conv (Conv2D)             (None, 10, 10, 64)        36928

 L6MaxPool (MaxPooling2D)    (None, 5, 5, 64)          0

 L7Conv (Conv2D)             (None, 3, 3, 128)         73856

 L8MaxPool (MaxPooling2D)    (None, 2, 2, 128)         0

 flatten_4 (Flatten)         (None, 512)               0

 L10Dense (Dense)            (None, 256)               131328

 L11Dense (Dense)            (None, 128)               32896

 dense_4 (Dense)             (None, 36)                4644

=================================================================
Total params: 307716 (1.17 MB)
Trainable params: 307716 (1.17 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Fig. 4.5: Neural Network Architecture

The features extracted from the dense layers are then passed to the classification model.

4.5 Classification

Classification is a fundamental task in machine learning that deals with assigning data points to one of a set of predefined categories. It’s like sorting emails into folders like "inbox," "spam," or "important." Classification algorithms learn to identify patterns in data that allow them to make these categorizations.

In its simplest terms, the goal is to predict a class label, which is a choice from a predefined list of possibilities [16]. Classification is sometimes separated into binary classification, which is the special case of distinguishing between exactly two classes, and multiclass classification, which is classification between more than two classes [16].

The main challenge in multi-class classification is to develop models that can effectively distinguish between these multiple classes and assign the most appropriate label to each new data point.

In our model:

● The features extracted from the Neural Network layer are passed to different classifiers for purpose of classification.

● The classifiers are then used to predict the target labels based on the extracted features.

● Since the target class contain 36 different labels the problem is a multiclass classification problem.

4.5.1 Random Forest Classifier

Random forests are one way to address the main downside of decision trees, that is even with the use of pre-pruning, they tend to over-fit and provide poor generalization performance. A random forest is essentially a collection of decision trees, where each tree is slightly different from the others. The idea behind random forests is that each tree might do a relatively good job of predicting, but will likely over-fit on part of the data. If we build many trees, all of which work well and over-fit in different ways, we can reduce the amount of over-fitting by averaging their results [16].

Whenever a new data point is arrived, it's passed through all the trees in the forest. Each tree makes a prediction, and the final classification is decided by a majority vote among the trees. This approach reduces the risk of over-fitting and leads to more robust and accurate predictions compared to a single decision tree.

● For model trained on all full 36 character images of handwritten devanagari characters, the accuracy of Random Forest Classifier was found to be in a range of 98.5% to 99.7%.

● For a model trained and validated on cropped character images of handwritten devanagari characters, the accuracy was around 72.4% to 80.32%.

● For a model trained and validated on both full and cropped character images of handwritten devanagari characters, the accuracy was around 80% to 90%.

4.5.2 Multi-Layer Perceptron Classifier

Multilayer perceptrons (MLPs) are also known as (vanilla) feed-forward neural networks, or sometimes just neural networks. MLPs can be viewed as generalizations of linear models that perform multiple stages of processing to come to a decision. In an MLP the process of computing weighted sums is repeated multiple times, first computing hidden units that represent an intermediate processing step, which are again combined using weighted sums to yield the final result [16].

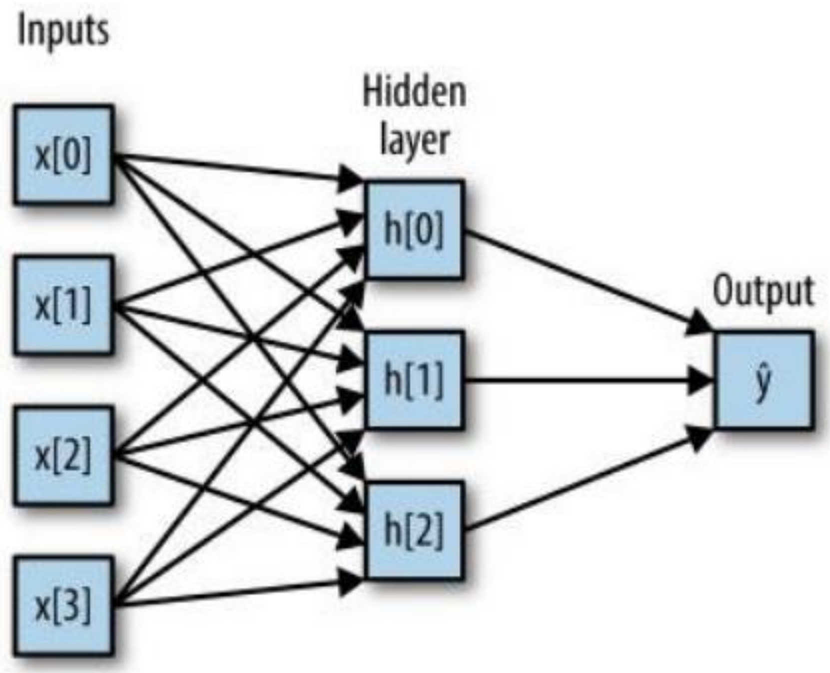

Fig. 4.6: Illustration of a multilayer perceptron with a single hidden layer [16]

This model has a lot more coefficients (also called weights) to learn: there is one between every input and every hidden unit (which make up the hidden layer), and one between every unit in the hidden layer and the output [16].

● For model trained on all full 36 character images of handwritten devanagari characters, the accuracy of MLP Classifier was found to be in a range of 97.8% to 99.8%.

● For a model trained and validated on cropped character images of handwritten devanagari characters, the accuracy was around 87% to 90.46%.

● For a model trained and validated on both full and cropped character images of handwritten devanagari characters, the accuracy was around 85% to 94.81%.

4.5.3 K-Nearest Neighbor Classifier

K-Nearest Neighbors (KNN) is a popular and straightforward machine learning technique used for both classification and regression tasks. It assumes that data points that are close together or have less distance between them in a feature space are likely to have similar characteristics. In simpler terms, similar things tend to behave similarly.

The k in k-nearest neighbors signifies that instead of using only the closest neighbor to the new data point, we can consider any fixed number k of neighbors in the training (for example, the closest three or five neighbors). Then, we can make a prediction using the majority class among these neighbors [16].

Building this model only consists of storing the training set. To make a prediction for a new data point, the algorithm finds the k points in the training set that are closest to the new point. Then it assigns the label using the majority class among these neighbors to the new data point [16].

● For model trained on all full 36 character images of handwritten devanagari characters, the accuracy of KNN Classifier was found to be in a range of 97.9% to 99.83%.

● For a model trained and validated on cropped character images of handwritten devanagari characters, the accuracy was around 70% to 75.87%.

● For a model trained and validated on both full and cropped character images of handwritten devanagari characters, the accuracy was around 80% to 88.69%.

4.5.4 Support Vector Machine (SVM) Classifier

A support vector machine (SVM) is a supervised machine learning algorithm that classifies data by finding an optimal line or hyperplane that maximizes the distance between each class in an N-dimensional space [17].

SVMs were developed in the 1990s by Vladimir Vapnik, Steven E. Golowich, Alex Smola and they published this work in a paper titled "Support Vector Method for Function Approximation, Regression Estimation, and Signal Processing" in 1995 [18].

SVMs are commonly used within classification problems. They distinguish between two classes by finding the optimal hyperplane that maximizes the margin between the closest data points of opposite classes. The number of features in the input data determines if the hyperplane is a line in a 2-D space or a plane in an n-dimensional space. Since multiple hyperplanes can be found to differentiate classes, maximizing the margin between points enables the algorithm to find the best decision boundary between classes. This, in turn, enables it to generalize well to new data and make accurate classification predictions. The lines that are adjacent to the optimal hyperplane are known as support vectors as these vectors run through the data points that determine the maximal margin [17].

- For model trained on all full 36 character images of handwritten devanagari characters, the accuracy of SVM Classifier was found to be in a range of 98% to 99.87%.
- For a model trained and validated on cropped character images of handwritten devanagari characters, the accuracy was around 82.4% to 86.45%.
- For a model trained and validated on both full and cropped character images of handwritten devanagari characters, the accuracy was around 85% to 91.74%.

4.6 SUMMARY OF PROPOSED APPROACH

Recognizing Handwritten Devanagari Character images given full and cropped images of handwritten devanagari characters is proposed in this report. From all the proposed works seen in the papers cited in the reference section, handwritten devanagari character recognition has been done with respect to the context of full character image and only some work is done with respect to the context of partial character image. As Devanagari script has been used in ancient times for document and text writing, it is very likely that we can have scripts or documents where cropped or partial characters of handwritten devanagari characters are present.

This report proposes to predict such cropped or partial handwritten devanagari characters images more accurately. For this we are using CNN model which is mainly composed of combination of Convolutional layers, Max-Pooling layers and Dense Neural networks layers (or Fully Connected Layers) for feature extraction and then pass those features to the classifiers like Random Forest Classifier, Multilayer Perceptron, K-NN, SVM Classifier.

4.7 Prediction

Prediction of Devanagari characters have following variations:

1. Predict the given full image of the handwritten Devanagari character when model trained on full character images.

   The model has been trained on full handwritten character images of the Devanagari characters. It then predicts the full handwritten devanagari character images with high accuracy.

2. Predict the given full image of the handwritten Devanagari character when model trained on cropped character images.

   The model has been trained on cropped handwritten character images of the Devanagari characters. It then predicts the full handwritten devanagari character images.

3. Predict the given cropped image of the handwritten Devanagari character when model trained on full character images.

   The model has been trained on full handwritten character images of the Devanagari characters. It then predicts the cropped handwritten devanagari character images. However, as the characters in Devanagari are similar their cropped images also look similar. Hence much accuracy cannot be obtained in this case.

Given below are three characters “kha”, “ra” and “Wa”. If “kha” is cut vertically, the left half will look like “ra” and the right side will look like “wa” and hence the model gets confused.

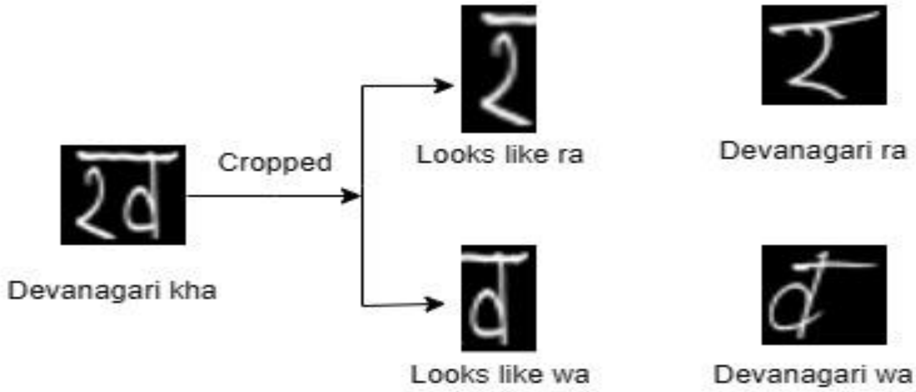

Fig. 4.7: “kha” and cropped “kha” which looks like “ra” and “wa”

4. Predict the given cropped and full image of the handwritten Devanagari characters when model trained on both full and cropped character images.

   The model has been trained on both full and cropped handwritten character images of the Devanagari characters. It then predicts the full and cropped handwritten devanagari character images.

Chapter 5 - Experiments

Various experiments were performed to analyze the accuracy when Neural Networks were used for extracting features and then using different Classifiers to predict the target label using those features. Each of the experiment performed has been explained below:

1) Training the model on full handwritten devanagari character images and making it predict the full handwritten devanagari character images.

- In this experiment, Neural Network were configured to extract the features from the images.

- Full handwritten devanagari character training images were then fed to this network consisting of multiple convolution, max-pooling and fully connected layers.

- The output of the second last layer (Layer 11 – Fully Connected Layer) from the Neural Networks consisting of the features was extracted after passing the inputs to the model.

- This output was then divided as train set and test set to evaluate classifiers performance.

- The classifiers were trained using these extracted features and their corresponding labels from the train set.

- The test set was used to assess the performance of the classifiers.

- Performance of each classifier showed variations depending on the type of data used to train and evaluate the model.

KEY FINDINGS:

- A little amount of variations were observed in the accuracies of different classifiers.
- The accuracy of the CNN model, if was used for classification, was around the around 94.8% to 96.99% for 25 epochs.

● When the classifiers were fed with features extracted from this neural network, then their accuracies was in the range of 97.8% to 99.87%.

● Thus, it could be seen that using the classifiers led to a small increase in the accuracy.

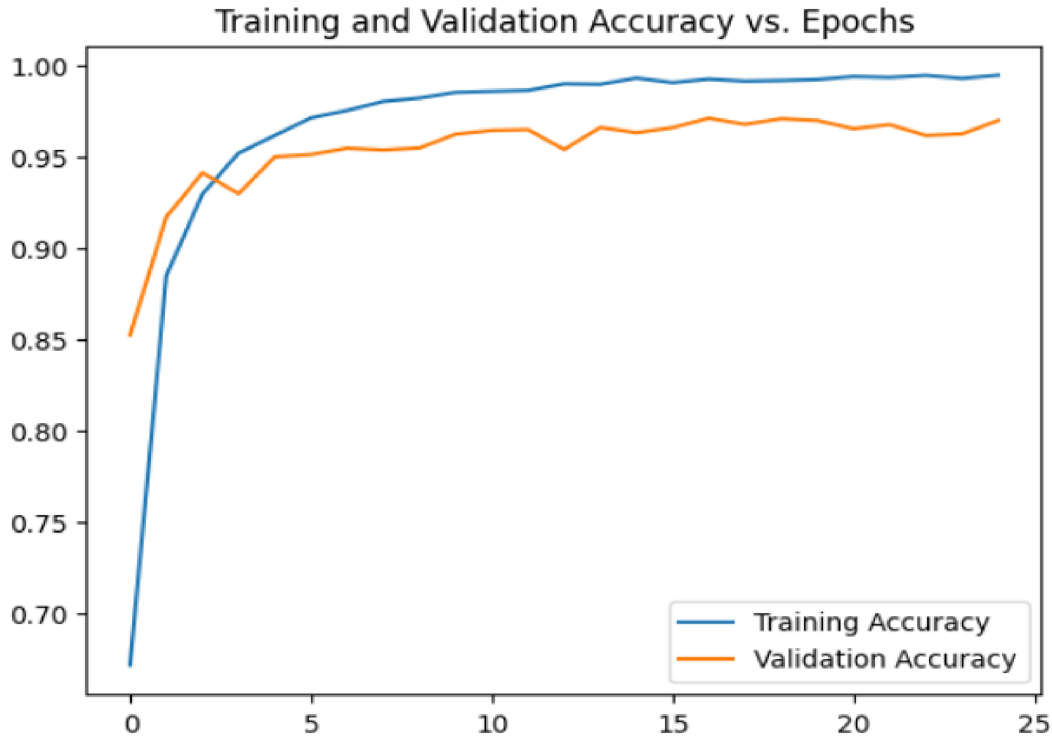

Fig. 5.1: Model accuracy over increasing epochs when trained with full character images

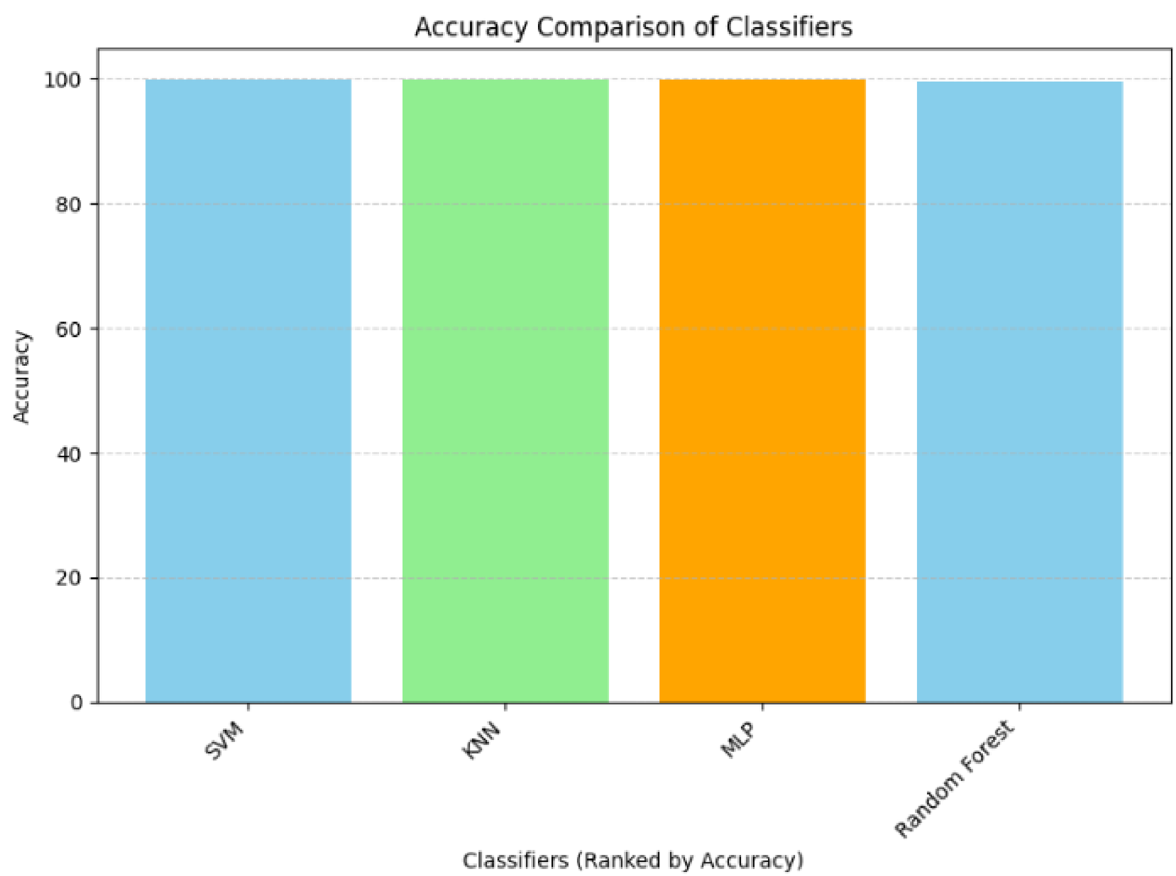

Fig. 5.2: Accuracy comparison of Classifiers when trained with full character images

Table 5.1: Prediction on Test images (300 of each character) of full handwritten devanagari character when the model is trained with full handwritten devanagari character images

	Random Forest	MLP	KNN	SVM
ka-क	93.67	96.0	93.0	96.0
kha-ख	94.34	95.34	98.67	97.0
ga-ग	97.67	97.34	81.34	99.67
gha-घ	91.34	94.0	11.0	92.67
nga-ङ	92.34	95.34	92.67	97.0
cha-च	97.0	97.34	88.34	98.0
chha-छ	91.67	96.67	90.34	96.0
ja-ज	94.67	98.34	89.67	98.34
jha-झ	95.0	98.34	97.67	99.0
nya-ञ	96.34	98.0	95.67	98.34
ta-ट	95.34	95.67	87.0	97.34
thaa-ठ	96.67	98.34	97.34	98.0
daa-ड	97.0	96.67	90.34	96.34
dhaa-ढ	97.34	97.34	96.34	96.34
adna-ण	99.0	99.34	99.67	99.0
t-त	98.34	98.67	98.0	99.0
tha-थ	78.34	97.34	10.0	94.0
da-द	84.34	94.0	78.34	92.0
dha-ध	97.34	93.34	100	98.0
na-न	97.0	98.67	93.67	98.67
pa-प	97.0	97.67	97.67	97.34
pha-फ	97.67	97.67	97.67	98.67
ba-ब	94.34	93.67	90.67	94.34
bha-भ	94.0	97.67	86.34	94.34
ma-म	90.34	96.67	97.34	94.34
yaw-य	81.34	95.0	87.0	89.67
ra-र	98.34	98.34	91.34	98.34
la-ल	98.0	97.67	97.67	98.67
wa-व	80.34	93.67	81.67	94.34
sha-श	97.67	98.0	99.34	98.0

shha-ष	96.0	98.0	97.67	97.34
sa-स	87.0	97.0	49.0	93.0
ha-ह	96.34	97.67	98.0	96.34
ksh-क्ष	96.34	97.34	96.67	97.0
tra-त्र	96.67	97.34	97.34	98.0
gya-ज्ञ	96.67	95.34	96.34	96.67

2) Training the model on cropped handwritten devanagari character images and making it predict the full handwritten devanagari character images.

● In real world cases, it is possible that the available images may contain cropped/partial characters due to poor data quality.

● An experiment was performed to see the effect of training the model using the cropped handwritten devanagari character images and then using it to predict the full handwritten devanagari character images.

● Only 500 images of each characters were used for training the model and for validation due to unavailability of existing cropped images of handwritten devanagari characters.

● Full handwritten devanagari character images were passed to the classifiers to predict those images.

KEY FINDINGS:

● This experiment yielded some interesting results and hypothesis.

● It was observed that when the cropped images of handwritten devanagari characters were used to train the model, the model was able to learn some of the features of certain images.

● Since only cropped handwritten devanagari images were used for training and validation the accuracy of the model was around 58% to 60.69%, as expected.

● However, it was observed that when full handwritten devanagari character images were passed to different classifiers that were trained on the features extracted by the CNN model above, for prediction, some of the classifiers were able to correctly predicted the labels of some characters. It was because the classifiers had an intimation of the features from the cropped devanagari character images which they could locate in the full images and correctly identify them leading to better accuracy, For more detail of which classifier was able to predict which characters (see Table 2).

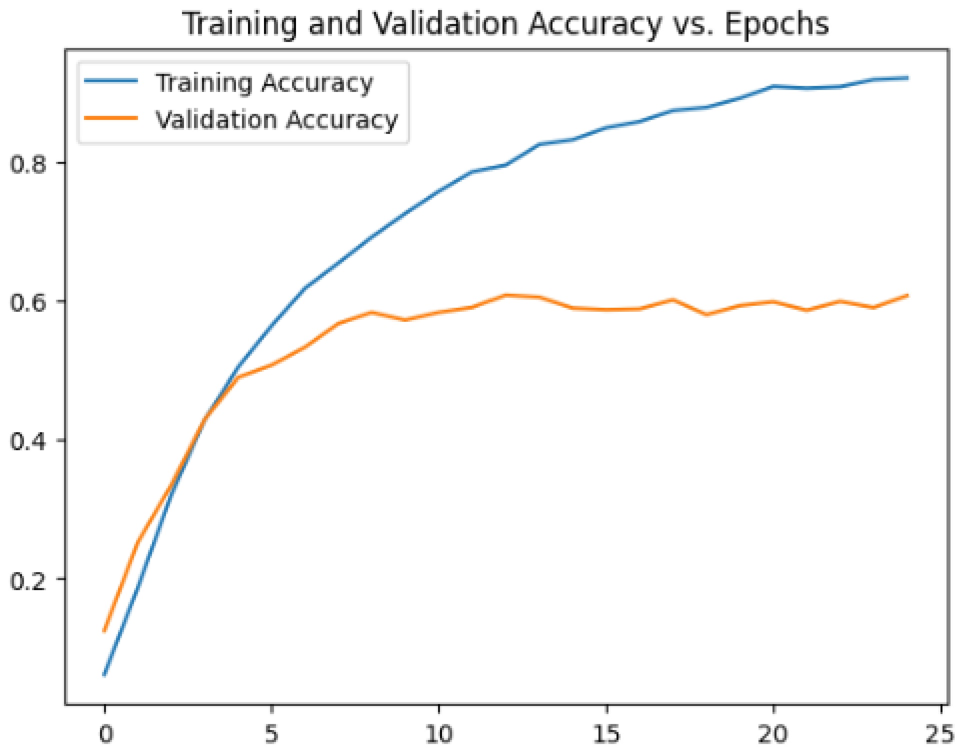

Fig. 5.3: Model accuracy over increasing epochs when trained with cropped character images

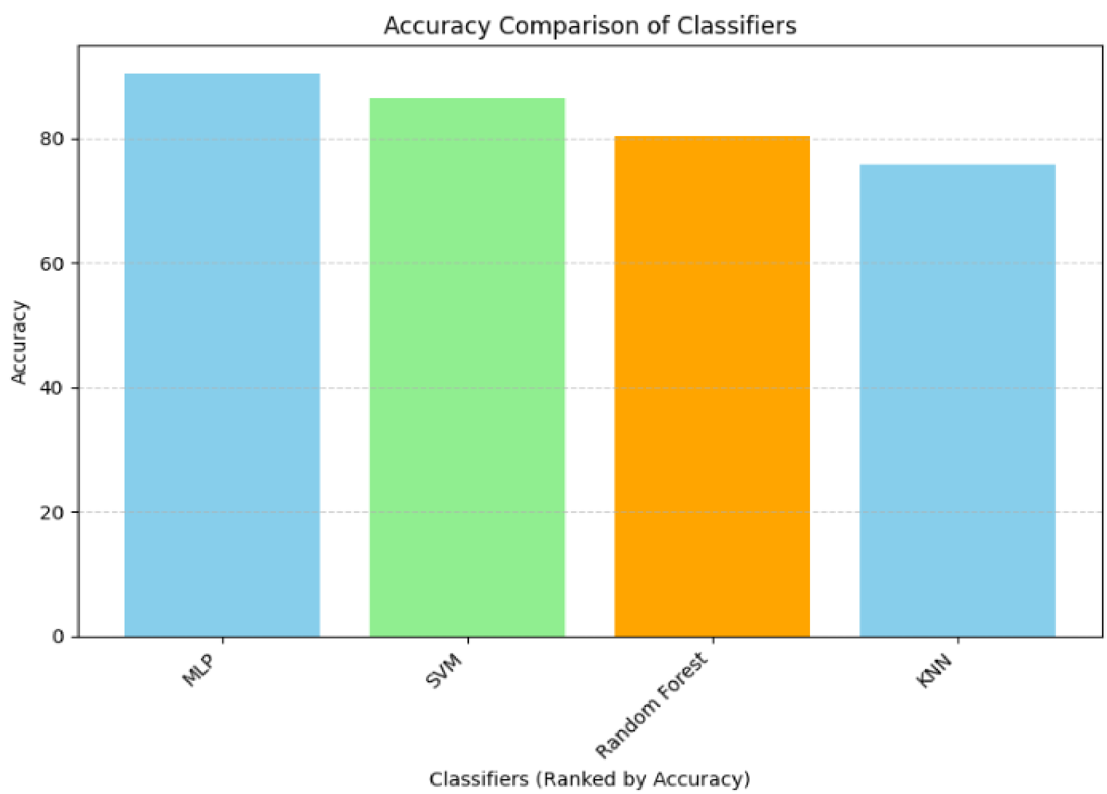

Fig. 5.4: Accuracy comparison of Classifiers when trained with cropped character images

Table 5.2: Prediction on Test images (300 of each character) of full handwritten devanagari character when the model is trained with cropped handwritten devanagari character images

	Random Forest	MLP	KNN	SVM
ka-क	4.67	0.67	5.67	4.0
kha-ख	45.34	24.36	47.67	70.0
ga-ग	4.67	8.0	1.0	14.67
gha-घ	8.0	7.34	2.33	7.34
nga-ङ	13.34	19.0	0.33	31.66
cha-च	8.0	9.67	5.0	2.67
chha-छ	21.0	33.0	9.34	27.66
ja-ज	16.66	5.67	36.0	11.33
jha-झ	19.0	12.66	23.66	26.33
nya-ञ	3.0	1.0	3.34	3.67
ta-ट	29.34	29.34	30.0	42.67
thaa-ठ	7.34	12.34	11.33	16.0
daa-ड	44.33	24.67	47.34	28.67
dhaa-ढ	16.33	21.0	6.67	20.0
adna-ण	31.33	24.67	35.67	38.66
t-त	2.34	5.0	1.0	5.0
tha-थ	10.0	9.34	12.0	13.0
da-द	5.67	19.33	2.0	16.33
dha-ध	11.67	10.67	11.0	4.34
na-न	25.0	27.67	3.34	27.0
pa-प	3.66	5.34	1.0	3.66
pha-फ	11.0	4.67	8.67	7.67
ba-ब	21.67	28.67	17.34	12.0
bha-भ	9.0	8.0	3.34	3.0
ma-म	5.34	8.0	1.67	4.0
yaw-य	0.67	15.0	2.34	5.67
ra-र	16.67	18.67	12.0	5.0
la-ल	34.0	13.0	40.34	8.67
wa-व	1.0	5.34	1.0	3.67
sha-श	9.0	7.67	1.67	6.67

shha-ष	6.0	15.0	5.34	14.34
sa-स	13.67	18.0	9.0	7.01
ha-ह	19.0	30.34	32.66	18.33
ksh-क्ष	19.0	22.33	1.67	9.67
tra-त्र	7.34	9.0	0.67	3.0
gya-ज्ञ	27.0	28.99	17.34	14.0

3) Training the model on full handwritten devanagari character images and making it predict the cropped handwritten devanagari character images.

● Following the reverse methodology implemented above in experiment no. 2, an experiment was performed to train the model on full handwritten devanagari character images and then predict cropped handwritten devanagari character images.

● The primary goal in this experiment was to analyze how the classifiers will perform on receiving a cropped handwritten devanagari character image for prediction.

KEY FINDINGS:

● Unlike the above experiment no. 2, this experiment provided with some poor results.

● In this experiment, the model was trained on full handwritten devanagari character images and then the features were extracted and used to train the classifiers.

● Such an arrangement led to the very poor performance of the classifiers due to the ambiguity introduced by cropping of the handwritten devanagari character images, as shown in Fig. 4.7.

● The model yielded good accuracy when trained and validated on the full handwritten devanagari character images.

● But when the cropped handwritten devanagari images were provided to classifiers for label prediction, this led to misclassification..

● Since Devanagari script contains so many such similar looking characters, even a small modification can lead to a complete change of character.

● For e.g. if character “kha”(ख) is cut vertically then the left part of “kha” (ख) will look like “ra”(र) and the right part of “kha” (ख) will look like “wa”(व).

● This lead to mediocre performance by the classifiers in the arrangement provided above.

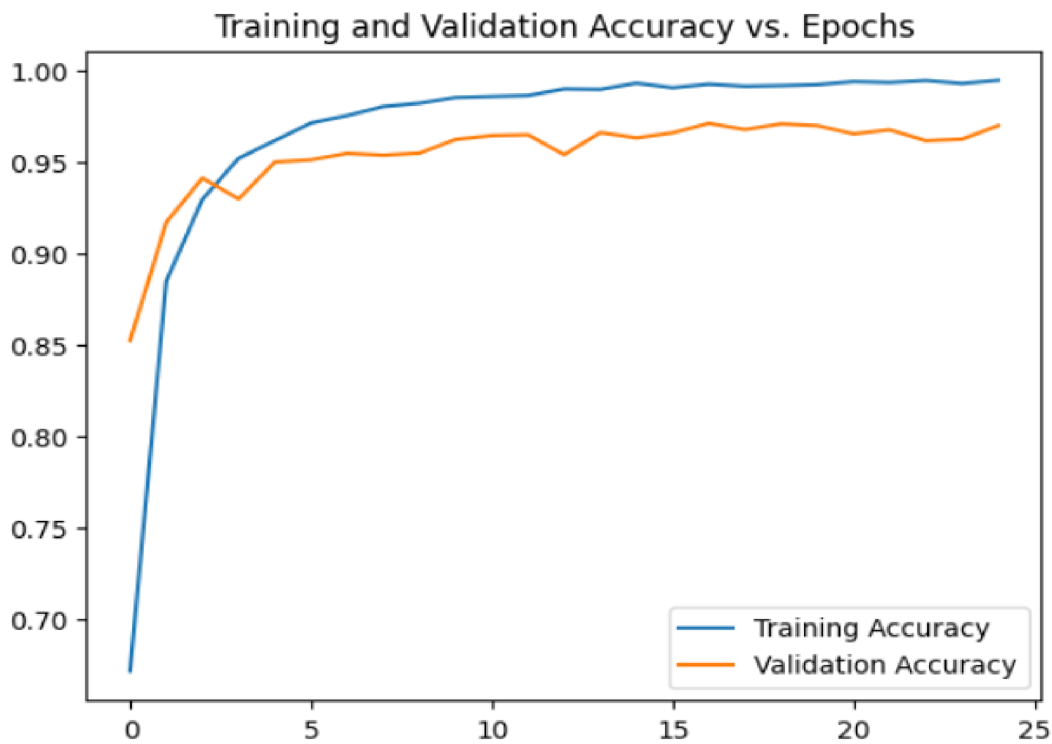

Fig. 5.5: Model accuracy over increasing epochs when trained with full character images

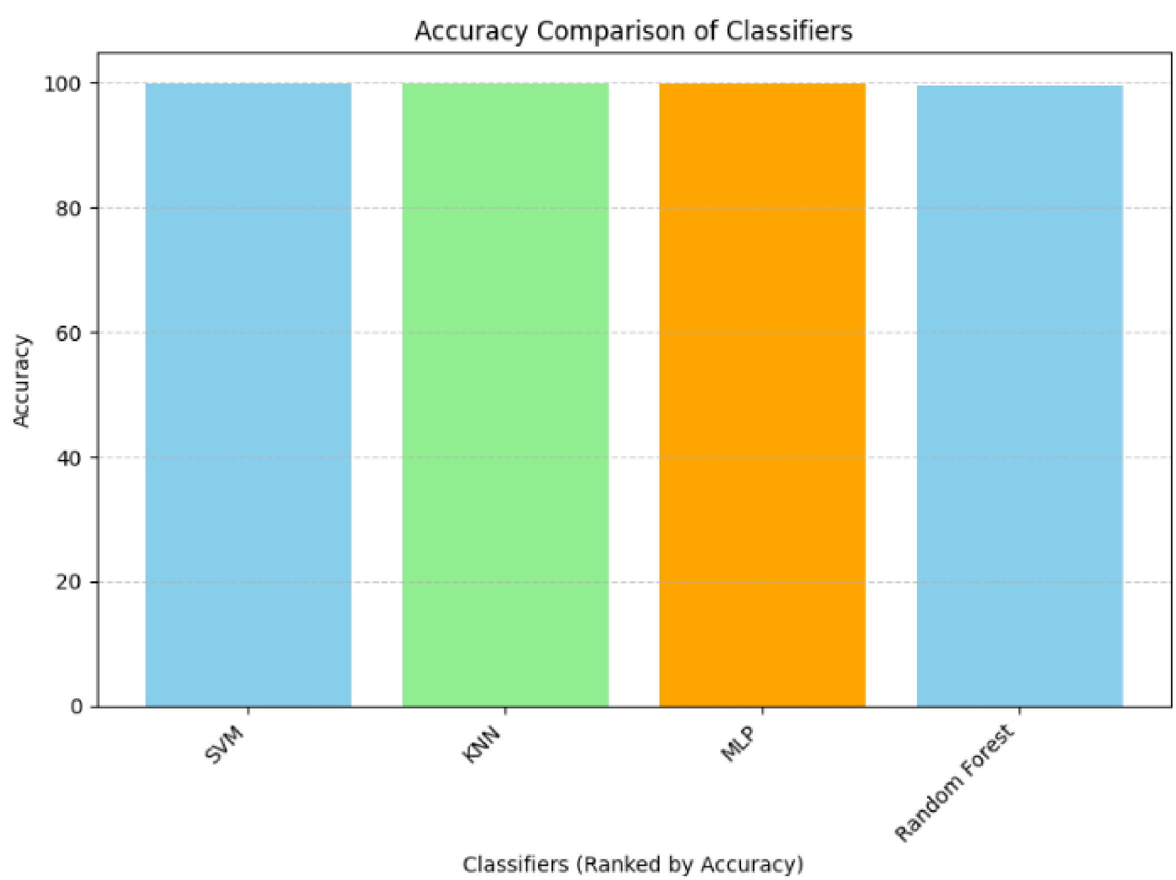

Fig. 5.6: Accuracy comparison of Classifiers when trained with full character images

Table 5.3: Prediction on Test images (200 of each character) of cropped handwritten devanagari character when the model is trained with full handwritten devanagari character images

	Random Forest	MLP	KNN	SVM
ka-क	0.5	1.0	1.0	4.5
kha-ख	11.5	3.50	14.0	12.0
ga-ग	22.0	18.0	5.0	45.5
gha-घ	2.5	1.5	0.0	1.5
nga-ङ	13.5	11.0	6.0	15.5
cha-च	12.0	8.0	9.5	7.5
chha-छ	11.0	15.5	13.0	14.0
ja-ज	7.0	4.5	2.5	5.0
jha-झ	4.0	6.5	6.0	7.0
nya-ञ	2.5	2.0	2.0	1.5
ta-ट	11.5	29.5	2.0	18.5
thaa-ठ	4.0	4.0	4.0	4.5
daa-ड	27.0	24.0	14.49	32.0
dhaa-ढ	8.5	9.5	3.0	4.5
adna-ण	30.5	32.0	35.5	29.5
t-त	6.0	12.0	6.5	24.5
tha-थ	5.0	9.5	0.0	2.0
da-द	9.5	12.5	4.5	10.5
dha-ध	16.0	4.0	23.0	11.5
na-न	7.5	9.0	5.5	6.5
pa-प	10.5	17.0	15.0	13.0
pha-फ	5.0	7.5	8.0	3.0
ba-ब	9.5	15.0	8.0	13.5
bha-भ	6.5	10.0	2.0	4.5
ma-म	5.0	5.5	5.0	5.5
yaw-य	2.5	6.0	2.5	3.0
ra-र	27.5	25.0	14.49	28.99
la-ल	9.5	8.5	7.5	8.5
wa-व	2.0	9.0	3.0	8.0

sha-श	14.0	12.5	27.0	16.0
shha-ष	6.5	24.5	14.49	13.0
sa-स	1.0	1.0	0.0	0.0
ha-ह	24.5	24.0	33.0	23.0
ksh-क्ष	9.5	10.5	21.0	12.5
tra-त्र	13.5	25.5	15.0	19.0
gya-ज्ञ	10.0	4.5	7.5	7.5

4) Training the model on both full images and cropped images of handwritten devanagari characters and making it predict the handwritten devanagari character given cropped and full image of the character.

● In this experiment, we train our model with both full and cropped handwritten devanagari images.

● We then perform the feature extraction on the second last layer of the cnn model.

● Since both the full and cropped handwritten devanagari images were used for training and validation the accuracy of the model was around 72.8% to 76.74%.

● We then use the extracted features to train our classifiers in order for them to be able to predict both the full and cropped handwritten devanagari images with a better accuracy then what is achieved in the earlier experiments mentioned above.

● We then passed the full and cropped handwritten devanagari images separately to each classifiers.

KEY FINDINGS:

● When the full handwritten devanagari character images were used for label prediction, it was observed that the accuracy of the classifiers were decreased as compared to the accuracy of these classifiers in experiment 1.

● It could be caused due to the presence of some cropped images where the pen or pencil stokes are not visible or are of poor quality, in the training data. Because of which the features selected by the CNN model might not be that good.

● Whereas, when the cropped handwritten devanagari character images were used for label prediction, it was observed that the accuracy of the classifiers were significantly increased as compared to the accuracy of these classifiers in experiment 3.

● It could be caused due to the use of cropped handwritten devanagari character images along with full handwritten devanagari character images, which allowed the CNN model to select more relevant features to classify the cropped handwritten devanagari character images.

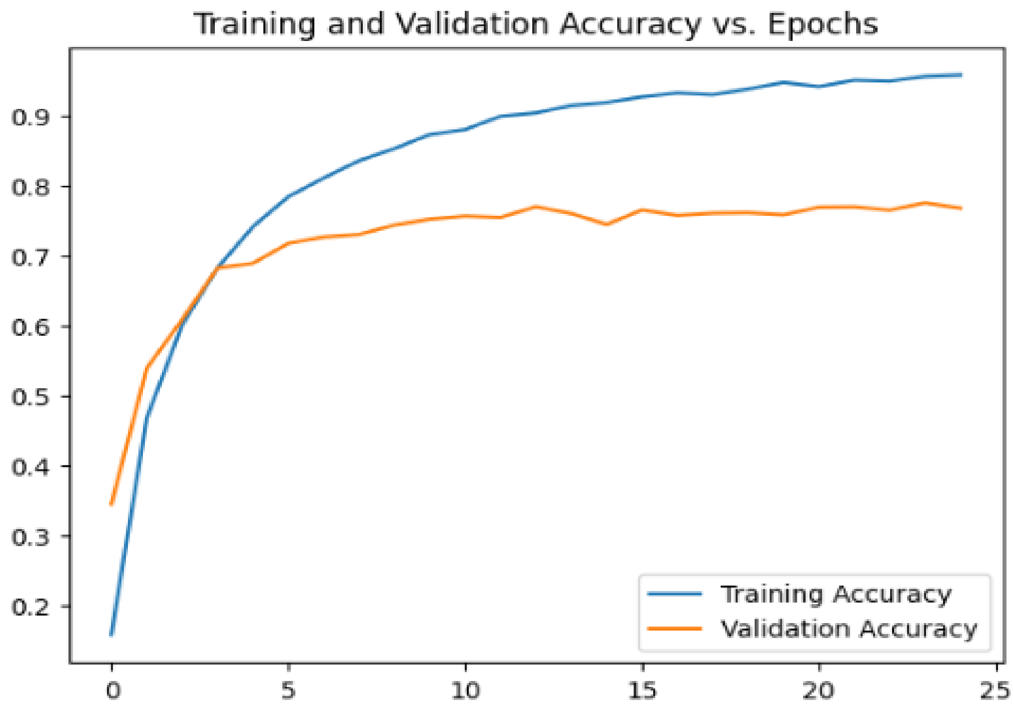

Fig. 5.7: Model accuracy over increasing epochs when trained with both full and cropped character images

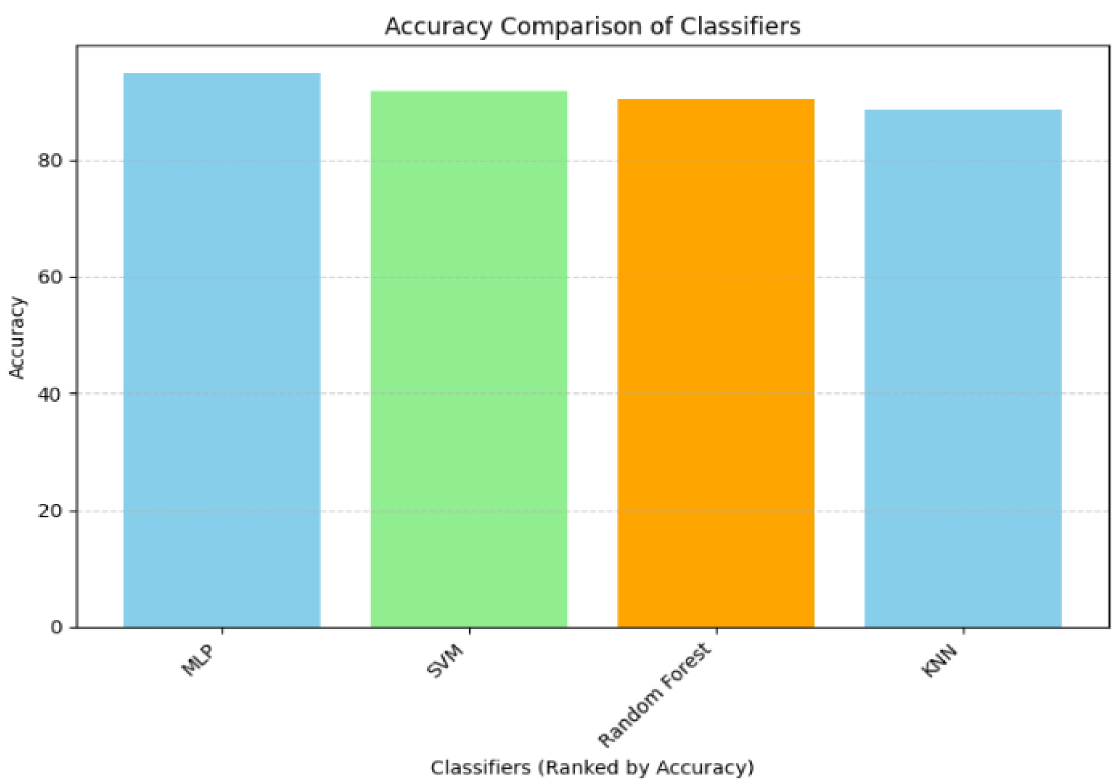

Fig. 5.8: Accuracy comparison of Classifiers when trained with both full and cropped character images

Table 5.4: Prediction on Test images (300 of each character) of full handwritten devanagari character when the model is trained with both full and cropped handwritten devanagari character images

	Random Forest	MLP	KNN	SVM
ka-क	85.34	83.3	85.67	88.34
kha-ख	81.34	90.34	14.002	95.34
ga-ग	81.34	84.34	67.0	93.67
gha-घ	53.0	71.34	35.33	60.0
nga-ङ	84.67	83.67	86.34	83.67
cha-च	65.34	79.67	53.33	81.0
chha-छ	54.33	83.0	40.66	83.34
ja-ज	85.0	77.67	36.33	67.67
jha-झ	76.34	78.34	85.34	84.67
nya-ञ	84.0	89.0	75.0	93.67
ta-ट	89.67	83.67	64.0	84.34
thaa-ठ	83.34	84.34	60.33	86.0
daa-ड	70.34	78.67	78.0	66.67
dhaa-ढ	86.67	91.34	77.34	91.0
adna-ण	92.0	94.67	96.34	74.67
t-त	68.34	79.34	4.0	71.0
tha-थ	71.34	72.34	72.67	63.0
da-द	28.96	74.34	16.33	52.64
dha-ध	82.0	82.67	83.34	86.67
na-न	81.34	80.67	75.67	73.67
pa-प	76.67	78.67	25.0	57.33
pha-फ	91.67	94.67	90.67	94.67
ba-ब	79.0	82.0	60.67	73.0
bha-भ	64.67	86.0	44.0	81.34
ma-म	80.0	86.34	65.67	70.0
yaw-य	57.33	77.67	51.34	70.34
ra-र	76.67	85.67	43.33	90.34
la-ल	69.67	83.67	67.34	81.67
wa-व	46.64	76.0	6.0	57.33

sha-श	83.34	86.34	32.0	80.34
shha-ष	89.0	91.67	88.67	87.67
sa-स	68.34	78.34	24.33	80.67
ha-ह	64.67	64.67	61.67	50.34
ksh-क्ष	91.0	93.34	85.34	77.34
tra-त्र	82.67	81.0	52.34	67.0
gya-ज्ञ	82.0	73.0	87.34	67.0

Table 5.5: Prediction on Test images (200 of each character) of cropped handwritten devanagari character when the model is trained with both full and cropped handwritten devanagari character images

	Random Forest	MLP	KNN	SVM
ka-क	28.0	26.0	22.0	53.5
kha-ख	33.0	27.5	10.5	47.0
ga-ग	20.5	44.5	1.5	60.0
gha-घ	9.5	25.5	2.5	18.5
nga-ङ	37.5	61.5	33.0	58.5
cha-च	30.0	50.0	11.5	56.99
chha-छ	38.0	55.0	16.0	45.5
ja-ज	39.5	47.5	7.0	28.49
jha-झ	42.0	68.5	55.0	75.5
nya-ञ	52.5	59.5	16.0	77.5
ta-ट	46.0	46.0	17.5	49.5
thaa-ठ	58.5	59.0	26.5	66.5
daa-ड	62.5	63.5	73.0	40.0
dhaa-ढ	26.5	46.5	18.5	57.49
adna-ण	61.0	61.5	55.0	61.5
t-त	23.0	33.0	1.0	20.0
tha-थ	40.0	57.49	45.0	45.0
da-द	7.5	34.0	7.0	20.0
dha-ध	41.0	46.5	24.0	55.5
na-न	28.9	39.5	30.5	30.5

pa-प	31.0	35.5	6.5	10.5
pha-फ	31.5	38.0	39.0	44.5
ba-ब	39.0	44.5	22.0	30.5
bha-भ	16.5	42.0	18.0	24.5
ma-म	19.0	35.5	12.0	20.5
yaw-य	14.49	37.0	12.5	17.0
ra-र	28.49	41.0	3.0	48.5
la-ल	41.0	55.5	23.5	36.0
wa-व	13.5	27.0	0.0	16.0
sha-श	31.0	59.0	13.5	59.5
shha-ष	37.0	45.5	45.0	37.0
sa-स	18.5	36.0	3.0	28.49
ha-ह	37.0	41.0	46.0	22.5
ksh-क्ष	41.0	58.5	61.0	33.5
tra-त्र	32.0	41.5	6.0	19.0
gya-ज्ञ	42.0	40.5	61.0	23.5

Chapter 6 - Conclusion

Devanagari script is the foundation of many languages. Huge amount of ancient text, documents, and religious books are written using devanagari script. It is very important to digitalize these for better access, sharing, management, organization, etc.., and also to preserve them as there are many handwritten documents in devanagari script.

Handwritten devanagari character recognition is a very complex task as many devanagari characters are similar looking. With the use of Neural Network model, which was mainly composed of combination of Convolutional layers, Max-Pooling layers and Dense Neural networks layers (or Fully Connected Layers) for extracting significant features of the handwritten devanagari character in the images which are used to train different classifiers to predict labels.

Moreover, experimenting with full and cropped handwritten devanagari character images used for trainings and predictions in the experiments showed that when model was trained with both full and cropped handwritten devanagari character images, the accuracy of predicting labels for cropped images was increased but the accuracy of predicting labels for full images was slightly decreased. But the accuracy achieved could be further improved by using a good quality of cropped handwritten devanagari character images with full handwritten devanagari character images for training the above model.

And among all the classifiers used in the experiments for classification the SVM Classifier showed a better overall accuracy as compared to other classifiers.

To conclude, Handwritten Devanagari Character Recognition, Image Pre-processing, Neural Networks, CNN, DNN, Feature extraction, Classifiers are the various popular fields for research and the insights of these topics can be obtained from this report.

References

[1] Devanagari (Nagari), Script Features and Description, United States: SIL International, 2013, archived from the original on 2 July 2017

[2] Aradhana A Malanker, Prof. Mitul M Patel,"Handwritten Devanagari Script Recognition: A Survey," EC Department, Gujarat Technological University, India, IOSR-JEEE, Volume 9, Issue 2 Ver. II (Mar – Apr. 2014), PP 80-87

[3] Deepa Kumari , Yogita Borse, Hardika Gawde, Asra Masrat, "Handwritten Devanagari Character Recognition Using Convolutional Neural Network," Dept. of Information Technology, K.J Somaiya College of Engineering, Mumbai, India, International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181 Published by : IJERTV13IS040083 Vol. 13 Issue 4, April 2024

[4] Priyanka Sawant, Siddhant Gokule, "Handwritten Devanagari Character Recognition using Neural Networks," 2018

[5] Sandhya Arora, Debotosh Bhattacharjee, Mita Nasipuri, L. Malik, M. Kundu and D. K. Basu, "Performance Comparison of SVM and ANN for Handwritten Devanagari Character Recognition," Dept. of CSE & IT, Meghnad Saha Institute of Technology, Kolkata; Department of Computer Science and Engineering, Jadavpur University, Kolkata; AICTE Emeritus Fellow, Department of Computer Science and Engineering, Jadavpur University, Kolkata; Dept. of Computer science. G.H. Raisoni college of Engineering, Nagpur, India, (06-05-2006)

[6] Ved Prakash Agnihotri, "Offline Handwritten Devanagari Script Recognition" I.J. Information Technology and Computer Science, July 2012

[7] Ashutosh Aggarwal, Mrs.Rajneesh Rani, Prof. Renu Dhir,“Recognition of Devanagari Handwritten Numerals using Gradient Features and SVM,” Department of Computer Science and Engineering, Dr B.R. Ambedkar National Institute of Technology, Jalandhar- 144011, Punjab (India), International Journal of Computer Applications (0975 – 888) Volume 48– No.8, June 2012

[8] Sonika Dogra, Chandra Prakash, “PEHCHAAN: Hindi Handwritten Character Recognition System Based On SVM,” Dapartment of CSE, Lovely Professional University Jalandhar (INDIA), 2012

[9] Neha Sahu, Nitin Kali Raman, "An efficient handwritten Devanagari character recognition system using neural network," Computer Science Department, ITM University, Gurgaon, Haryana, India, 2013

[10] Karbhari V. Kale, Prapti D. Deshmukh, Shriniwas V. Chavan, Majharoddin M. Kazi, Yogesh S. Rode, "Zernike Moment Feature Extraction for HandwrittenDevanagari (Marathi) Compound CharacterRecognition," International Journal of Advanced Research in Artificial Intelligence, Vol. 3, No.1, 2014

[11] Shalaka Prasad Deore and Albert Pravin, "Devanagari Handwritten Character Recognition using fine-tuned Deep Convolutional Neural Network on trivial dataset," Department of Computer Science and Engineering, Sathyabama Institute of Science and Technology, Chennai,India; Department of Computer Engineering, M.E.S. College of Engineering, Pune, S.P Pune University, Pune, India, 2020

[12] Aditi Moudgil, Saravjeet Singh, Vinay Gautam, Shalli Rani, Syed Hassan Shah, "Handwritten devanagari manuscript characters recognition using capsnet," Chitkara University Institute of Engineering and Technology, Chitkara University, Punjab, India; State California University, California, USA, 2023

[13] GeeksforGeeks, “Data Preprocessing in Data Mining,” (06 May, 2023)

[14] Domino, "Feature Extraction,"

[15] Jie Shi, Zhengyu Li, Hong Zhao, "Feature selection via maximizing inter-class independence and minimizing intra-class redundancy for hierarchical classification," Information Sciences, Volume 626, May 2023, Pages 1-18

[16] Andreas C. Müller and Sarah Guido, "Introduction to Machine Learning with Python, A Guide for Data Scientists," Published by O’Reilly Media, 2017, pg. no. 25, 83, 104-106, pg. no. 106.Fig. 2-45, pg. no. 20

[17] IBM, "What are support vector machines (SVMs)?," (07/12/2023)

[18] Vladimir Vapnik, Steven E. Golowich, Alex Smola, "Support Vector Method for Function Approximation, Regression Estimation, and Signal Processing, " AT&T Research 101 Crawfords Corner Holmdel, N J 07733; Bell Laboratories 700 Mountain Ave. Murray Hill, NJ 07974; GMD First Rudower Shausee 5 12489 Berlin; (1995)