Sarvajanik College of Engineering andTechnology
Department of Computer Engineering

B.E.III Sem-VI

Subject: Deep Learning and Neural Network

Mini Project Report

Team Members:
1. Aayushi Kosambia-200420107096

Project Name : Hindi Character Recognition

Date :

Guided By :
Prof. Sarosh Dastoor

CERTIFICATE

This is to certify that the report of DEEP LEARNING AND NEURAL NETWORK entitled “Hindi Character Recognition” has been carried out by Aayushi Kosambia(200420107096) studying at “Sarvajanik college of engineering and technology” for partial fulfillment of the minor subject Deep learning and neural network at Sarvajanik college of Engineering and Technology affiliated to GujaratTechnological University during the academic year, 2022-2023.

Date:
Place: SCET, Surat

Prof.DR. Sarosh Dastoor
Faculty of Deep learning and neural network

ACKNOLEDGEMENT

We would like to express our special thanks of gratitude to our guide Prof.Dr Sarosh Dastoor aswell as our Head of department Dr.Pariza Kamboj who gave us the golden opportunity to do this wonderful project on the topic “Hindi Character Detection” which also helped us in doing a lot of Research and we came to know about so many new things. We are really thankful to them.

Secondly, we would also like to thank our parents and friends who helped us a lot in finalizing this project within the limited time frame. And lastly our gratitude to our college for maintaining such a library with lots of books and research papers and labs which made it lot easier to research on this topic.

Last but not the least, thanks to Sarvajanik college of engineering and technology for giving us the platform for representing this project.

Aayushi Kosambia
(200420107096)

ABSTRACT

In this report we present a handwritten Hindi character recognition system based on different Deep learning technique. Handwritten character recognition plays an important role and is currently getting the attention of researchers because of possible applications in assisting technology for blind and visually impaired users, human–robot interaction,automatic data entry for business documents, etc. In this work, we propose a technique to recognize handwritten Hindicharacters using deep learning approaches like Convolutional Neural Network (CNN) With Optimizer RMSprop (Root MeanSquare Propagation) , Adaptive Moment (Adam) Estimation and Keras Model. The proposedsystem has been trained on samples of large set of database images and tested on samples images from user defines data set and from this experiment we achieved very high recognition results. Experimental results are compared with other neural network based algorithm

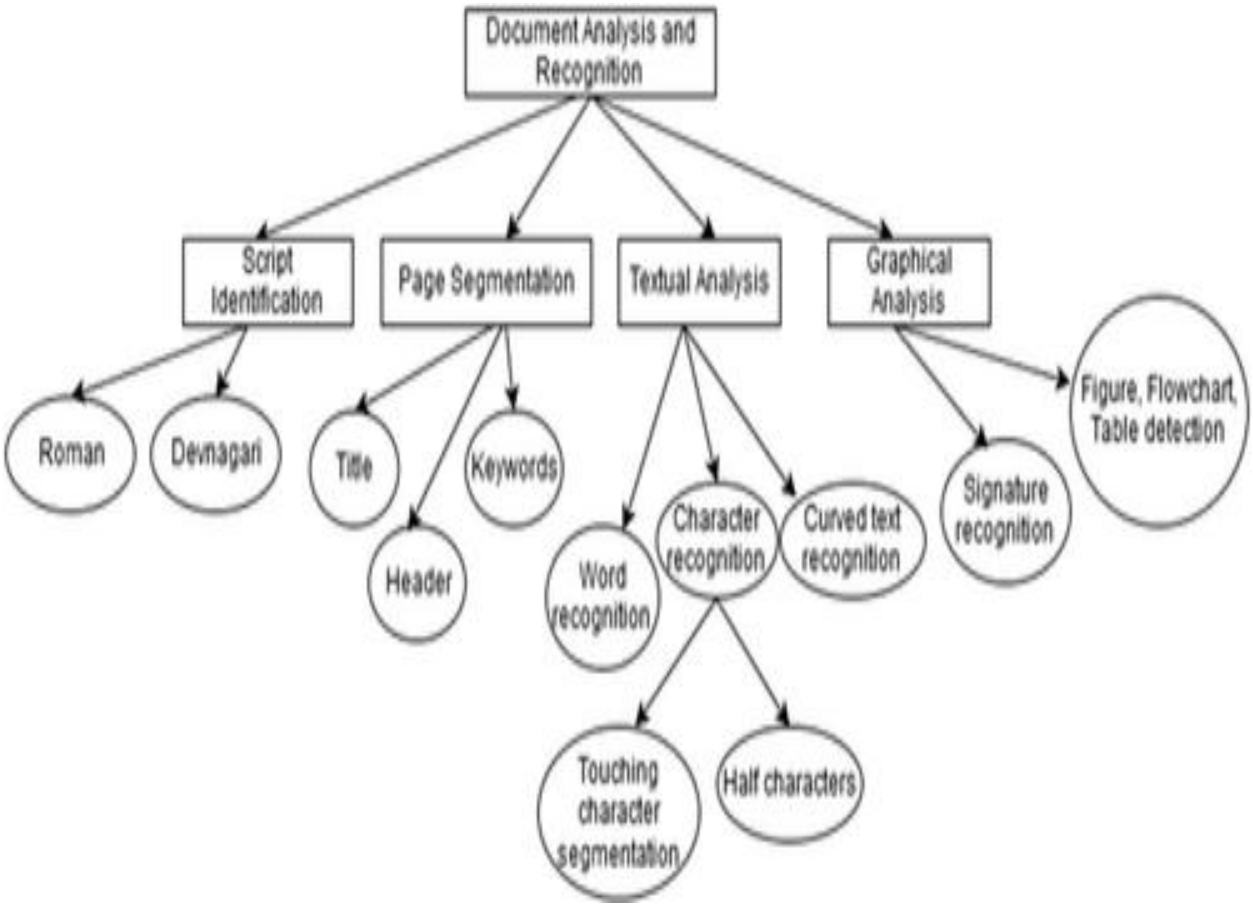
INDEX

1.INTRODUCTION TO DLNN

- Deep learning is a subset of machine learning, which involves the use of neural networks to train models. Neural networks are computational systems inspired by the structure and function of the human brain.
- Deep learning has recently emerged as one of the most powerful and widely used techniques for solving complex problems in a variety of domains, including computer vision, natural language processing, and robotics, among others.
- Neural networks are made up of multiple layers of interconnected nodes, or neurons, which process input data and generate output predictions. Each neuron performs a simple computation, such as a weighted sum of its inputs followed by a nonlinear activation function.
- The weights and biases of the neurons are learned during the training process by minimizing a cost function, which measures the error between the predicted output and the true output. This optimization process is typically performed using an algorithm called backpropagation, which calculates the gradient of the cost function with respect to the network parameters.
- One of the main advantages of deep learning is its ability to automatically learn features from raw data, without the need for manual feature engineering. This is achieved by stacking multiple layers of nonlinear transformations, which can capture increasingly abstract and complex patterns in the data.
- For example, in computer vision, a deep convolutional neural network can learn to recognize objects in an imageby extracting features such as edges, corners, and textures at different scales and orientations.
- Another key advantage of deep learning is its ability to generalize well to unseen data, even in the presence of noise and variability. This is achieved by using regularization techniques such as dropout, which randomly drops
- out neurons during training to prevent overfitting, and by using large datasets for training, which can help to capture the underlying structure of the data.
- In conclusion, deep learning and neural networks have revolutionized the field of artificial intelligence by enabling machines to learn from complex data and make accurate predictions.
- As the field continues to evolve, we can expect to see even more exciting applications of deep learning in areas such as healthcare, finance, and Transportation among others.

2.AIM

- Real time hindi characters recognition is an innovative project that aims to develop a system for recognizing handwritten characters in the Devanagari script.
- Handwritten character recognition has been a challenging problem in the field of computer vision and machine learning for many years.
- This project builds upon previous research in the field of handwritten character recognition and aims to develop a system specifically for recognizing Devanagari characters.



3.SOLUTION OF MODEL WITH EXPLANATION

1.1 Model-1 : Real-time language detection from images.

A. !pip install easyocr is a command to install the easyocr library using pip. The exclamation mark ! indicates that the command is executed in the command-line interface (CLI), not in Python code.

B. import matplotlib.pyplot as plt imports the pyplot module from the matplotlib library. It is typically used for visualizing and displaying images or plots.

C. import cv2 imports the cv2 module from the OpenCV library, which provides various computer vision and image processing functions.

D. import easyocr imports the easyocr module, allowing you to use its functionalities for optical character recognition.

E. from pylab import rcParams imports the rcParams class from the pylab module. rcParams is used for setting various plot parameters in matplotlib.

F. from IPython.display import Image imports the Image class from the IPython.display module. This class is used for displaying images in Jupyter Notebook or IPython environments.

G. The code rcParams['figure.figsize'] = 8, 16 sets the figure size for plots created using matplotlib to 8 inches in width and 16 inches in height. This line ensures that the resulting plots will have a specific size when displayed.

H. The next line reader = easyocr.Reader(['hi']) creates an instance of the easyocr.Reader class, which is the main class provided by the easyocr library for performing OCR tasks. The argument ['hi'] specifies the languages to be used for text recognition. In this case, it indicates that the OCR reader should be configured to recognize text in Hindi (the language code for Hindi is 'hi'). You can include multiple language codes in the list if you want to recognize text from multiple languages.

I. These file names represent the paths to specific image files on your system. Each variable is assigned a string value representing the file path to an image file in JPEG or PNG format.

J. Image(file_name0) displays the image specified by file_name0. It uses the Image class from the IPython.display module to show the image in Jupyter Notebook or IPython environments.

K. output = reader.readtext(file_name0) uses the readtext() method of the easyocr.Reader instance (reader) to perform OCR on the image specified by file_name0. It extracts the text from the image and assigns the result to the output variable.

L. output contains the OCR results, which are usually in the form of a list of text and its corresponding bounding box coordinates.

M. cord = output[3][0] selects the bounding box coordinates for the fourth text detected in the image. It assumes that the desired text and its coordinates are stored at index 3 in the output list.

N. x_min, y_min = [int(min(idx)) for idx in zip(*cord)] calculates the minimum x and y values from the bounding box coordinates using a list comprehension. It assumes that the cord variable is a list of tuples representing the coordinates.

O. x_max, y_max = [int(max(idx)) for idx in zip(*cord)] calculates the maximum x and y values from the bounding box coordinates using another list comprehension.

P. image = cv2.imread(file_name0) reads the image specified by file_name0 using the cv2.imread() function from the OpenCV library. The image is stored in the image variable.

Q. cv2.rectangle(image, (x_min, y_min), (x_max, y_max), (255, 0, 0), 2) draws a rectangle on the image using the minimum and maximum coordinates calculated earlier. The rectangle is drawn with blue color (BGR format) and a thickness of 2 pixels.

R. plt.imshow(cv2.cvtColor(image, cv2.COLOR_BGR2RGB)) displays the modified image with the rectangle using the imshow() function from matplotlib.pyplot. The cv2.cvtColor() function is used to convert the color space of the image from BGR to RGB format, which is expected by imshow().

3.2 Model-2 : Hindi-character recognition.

A. Importing necessary libraries: The code imports various libraries including numpy, pandas, matplotlib.pyplot, os, and the required modules from scikit-learn and Keras.

B. Inserting the dataset: It reads the data from the CSV file "/content/data.csv" using pandas and assigns it to the DataFrame named "train".

C. Creating dataframe of data: The code reassigns the "train" DataFrame to itself, which is redundant in this case.

D. Printing sum of Data: It calculates the sum of null values in each column of the "train" DataFrame using the isnull().sum() method and prints the result. It also prints the data types of each column using the dtypes attribute.

E. Extracting the last column from the table: It extracts the values from the "character" column of the "train" DataFrame and assigns them to the "Column_name" variable. Then, it deletes the "character" column from the DataFrame. Finally, it converts the "train" DataFrame to a NumPy array of integers.

F. Visualizing Images: It creates a dictionary to store images for visualization. It iterates over the "train" array in steps of 2000 and reshapes each image to a 32x32 array. It adds the reshaped image to the dictionary with an index as the key. It then plots the images from the dictionary using subplots and displays them.

G. Encoding: It performs label encoding and one-hot encoding on the "Column_name" array using the LabelEncoder and OneHotEncoder classes from scikit-learn. It also demonstrates how to invert the encoding for the first example.

H. Training and Testing Data: It splits the "train" array and the one-hot encoded labels into training and testing datasets using the train_test_split() function from scikit-learn. The test dataset size is set to 10% of the total data. It reshapes the training and testing data to match the input shape required by the CNN model. Additionally, it normalizes the pixel values to the range of 0 to 1.

I. Keras Model-CNN: It defines a CNN model using the Sequential API of Keras. The model consists of convolutional, pooling, flattening, dropout, and dense layers. The model is compiled with the Adam optimizer and categorical cross-entropy loss function. It displays the summary of the model.

J. Training the Model: It trains the CNN model using the fit() function. The training data, one-hot encoded labels, batch size, and number of epochs are provided as parameters. Validation data can be specified as well.

K. Evaluating the Model: It evaluates the trained model on the test data using the evaluate() method and prints the evaluation metrics such as loss and accuracy.

L. Plotting Model Accuracy and Loss: It plots the accuracy and loss values for both training and validation datasets across epochs using the history object obtained from the training process.

M. Visualizing Losses and Accuracy: It plots the training and validation losses and accuracies across epochs using the history object.

N. Saving the Model: It saves the trained model to a file named "my_model.h5" using the save() method of the model object.

O. Prediction and Confusion Matrix: It predicts the test set results using the trained model and calculates the confusion matrix to evaluate the model's performance.

4.DIAGRAMS

4.1 BASIC HINDI CHARACTERS

अ आ इ ई उ ऊ

ए ऐ ओ औ ऋ

क ख ग घ ङ

च छ ज झ ञ

ट ठ ड ढ ण

त थ द ध न

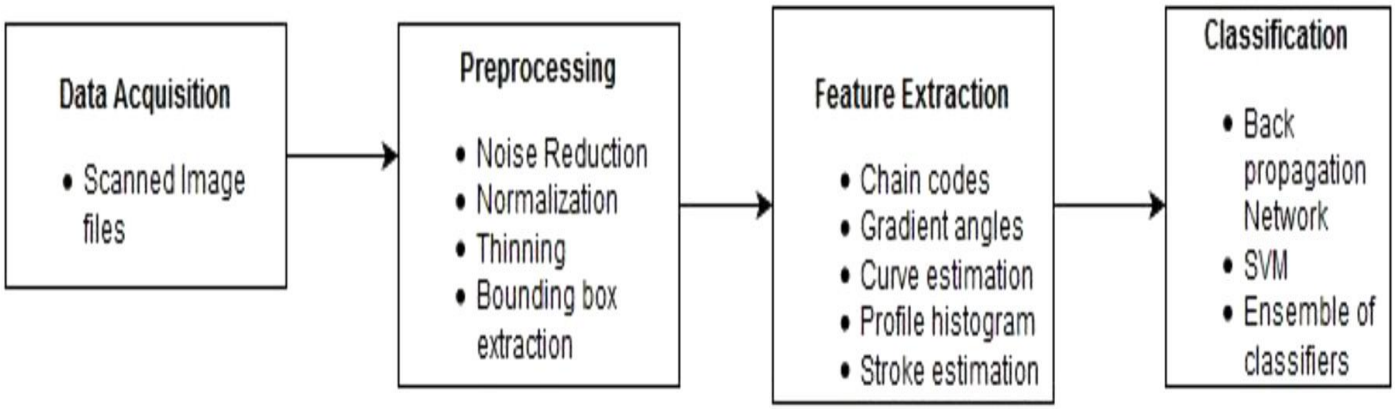
प फ ब भ म

य र ल व

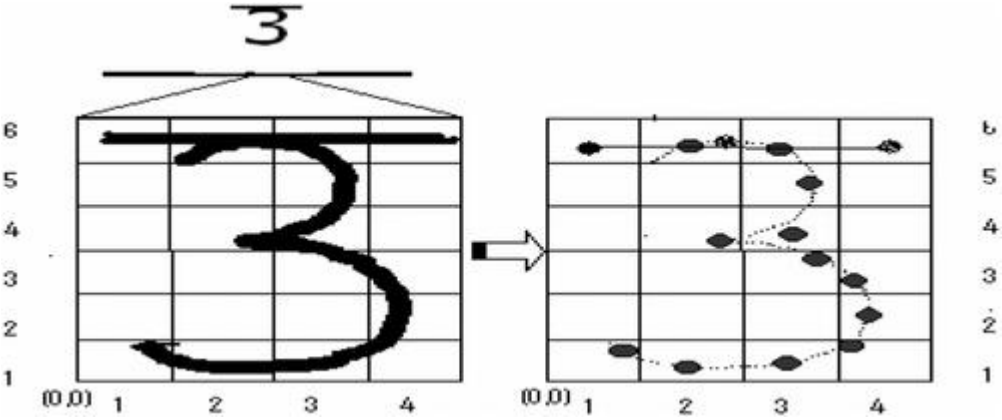
श ष स ह

क्ष त्र ज्ञ

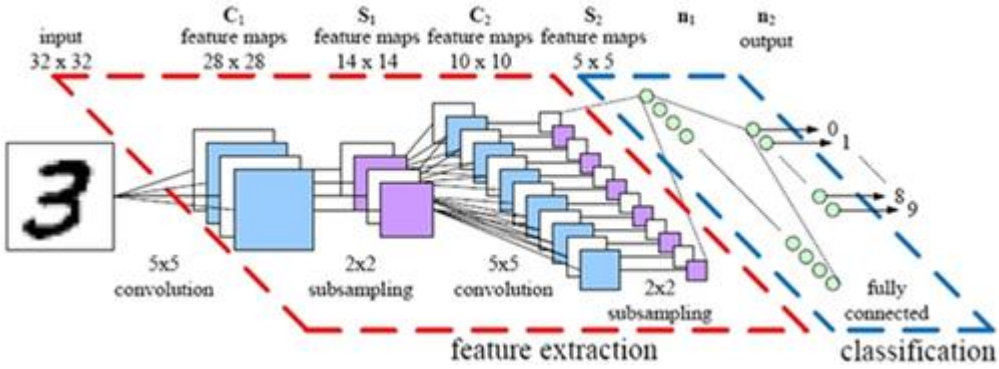
4.2 BASIC STAGES OF CHARACTER RECOGNITION

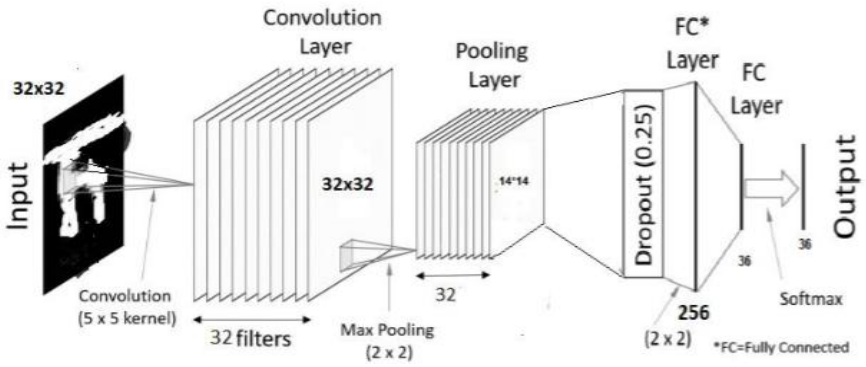


4.3 SPATIAL DIVISION OF HINDI CHARACTER BY BOX METHOD

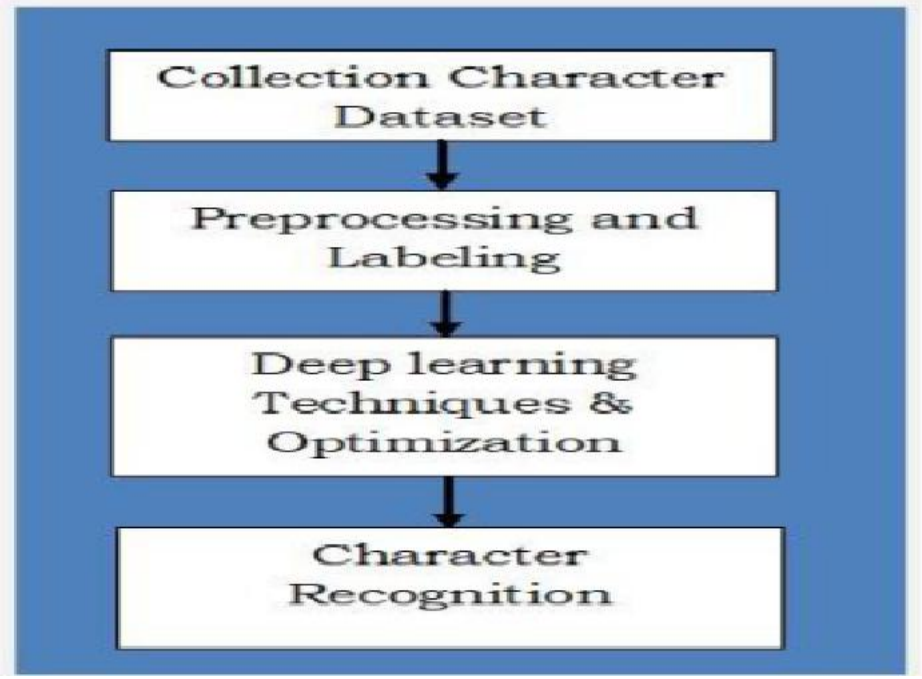


4.4 BASIC ARCHITECTURE OF CNN





4.5 BLOCK DIAGRAM PROPOSED HANDWRITTEN DIGIT RECOGNITION SYSTEM



5.DATASET FOR THE MODEL

pixel_0000	pixel_0001	pixel_0002	pixel_0003	pixel_0004	pixel_0005	pixel_0007	character
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka

0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_

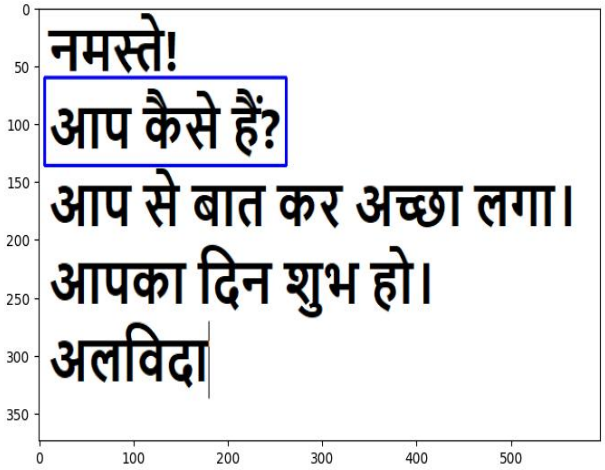
							ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka

0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka
0	0	0	0	0	0	0	character_01_ka

6.OUTCOME

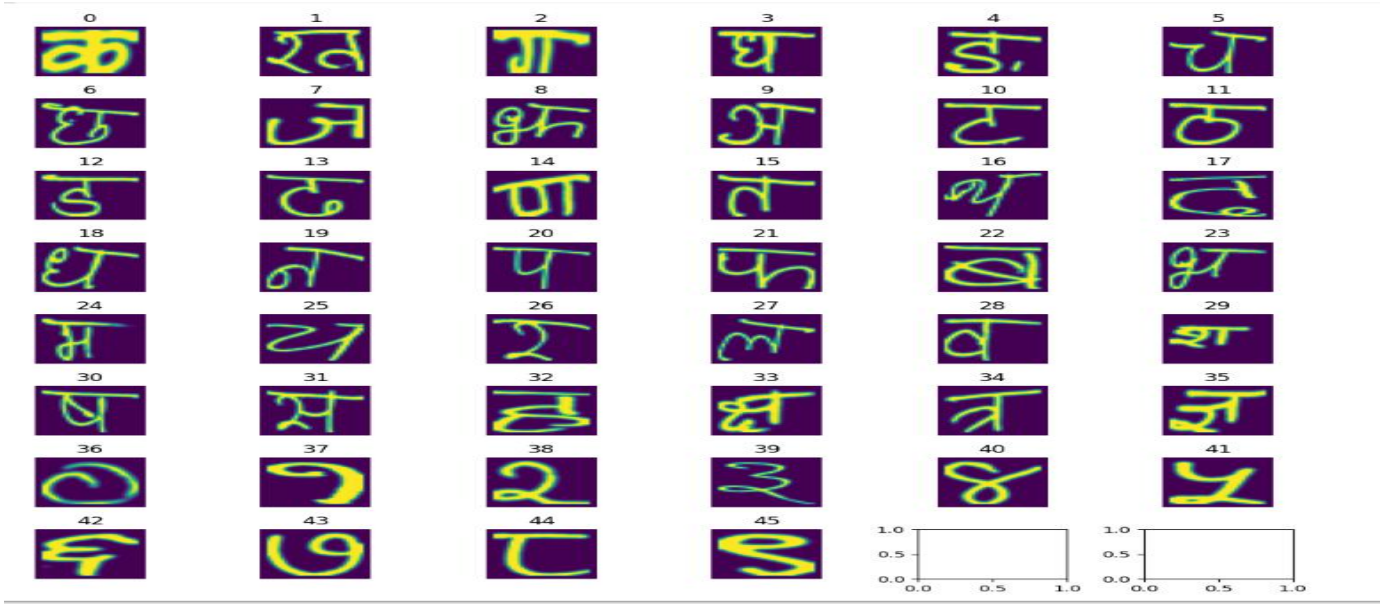
6.1 Model-1

- **Visualising Images**

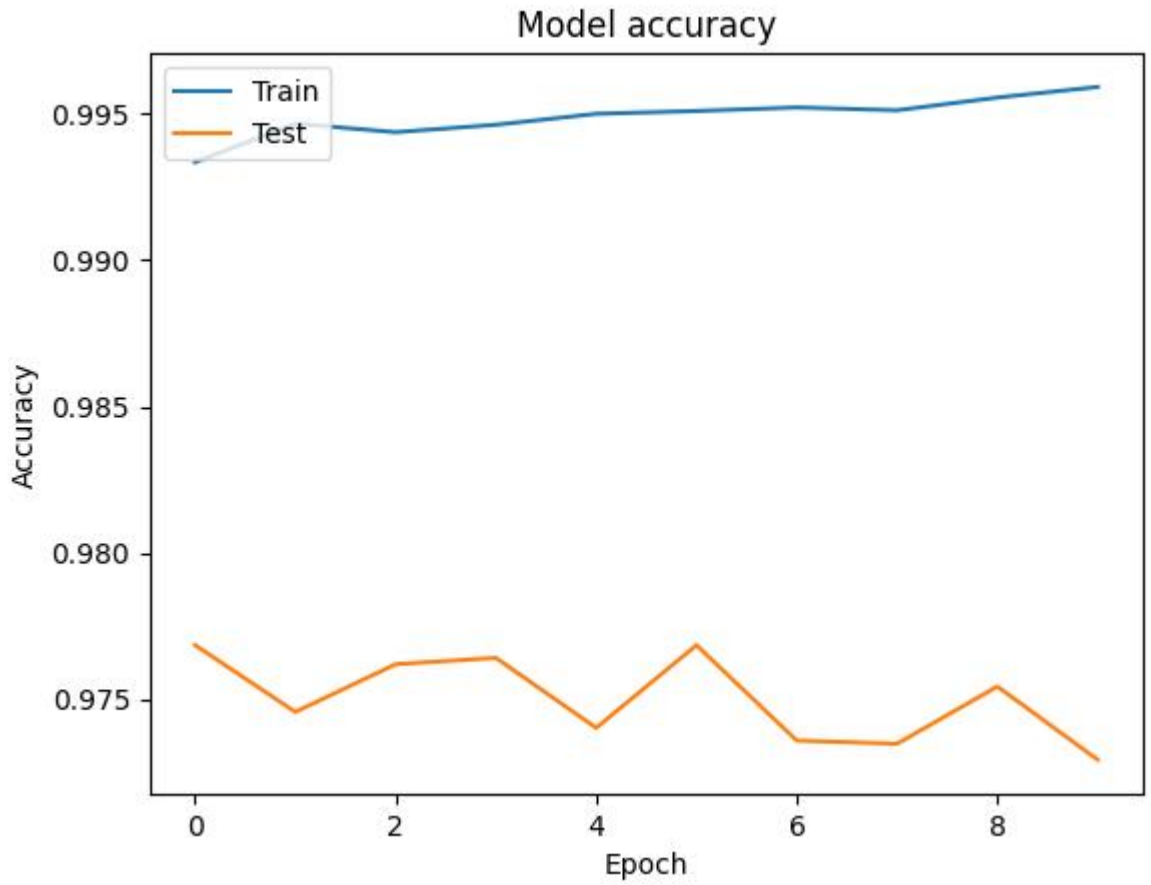


6.2 Model -2

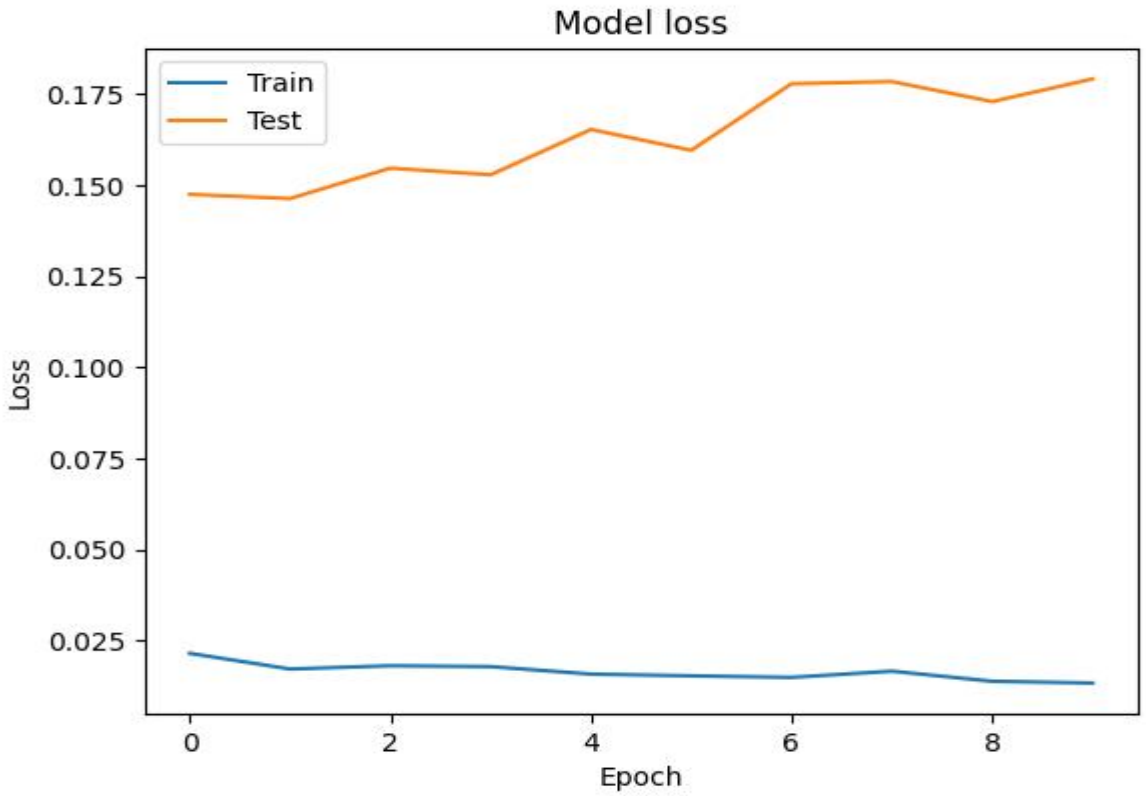
- **Visualising Images**



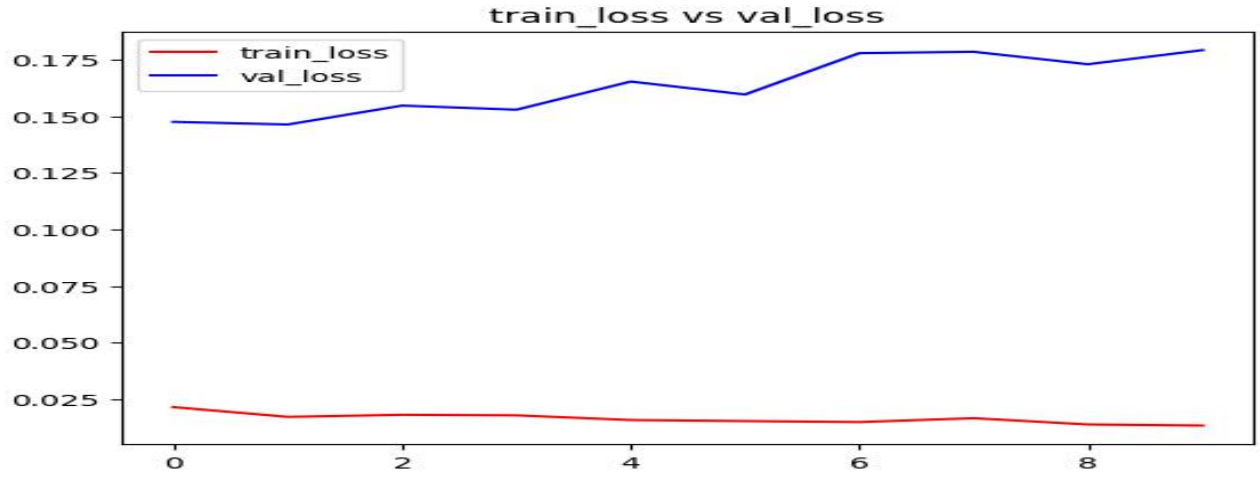
- **Model accuracy**

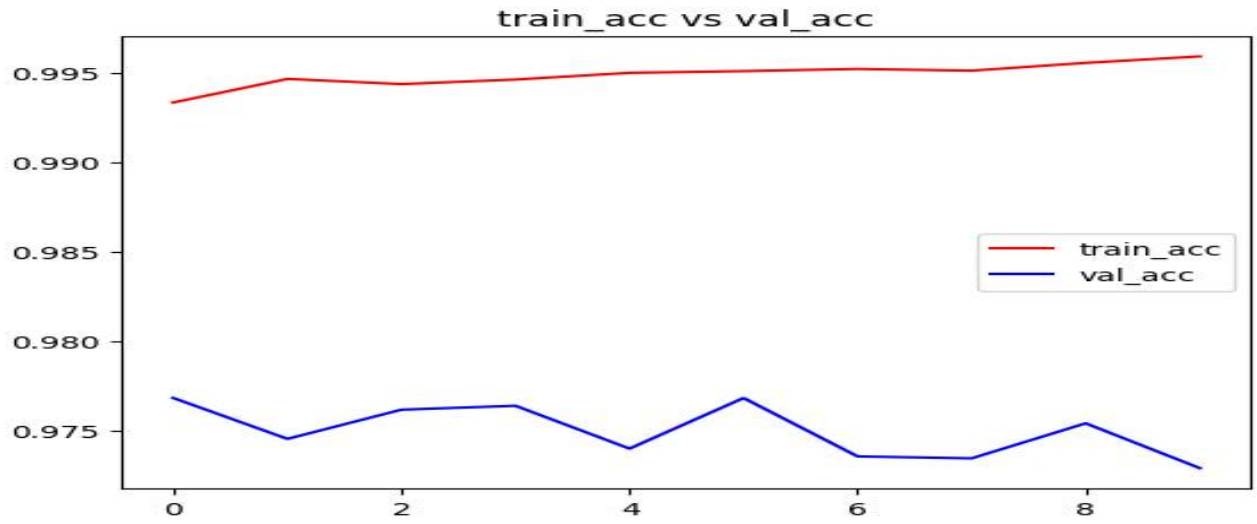


- **Plot training & validation loss values**



- **Visualizing losses and accuracy**





7.APPLICATION

A. The model built in this Python code can be used to predict the outcome of hindi language based on the image properties and font. This can be useful in a variety of day-to-day life, where predicting the outcome of hindi language accurately can save time and resources.

B. Digitizing Books.

C. Indexing of images in Search Engines.

D. Help for people who don't know how to operate a keyboard.

E. Recognizing addresses on envelopes in post offices.

F. Use of those who don't know Hindi.

G. Overall, this model can be used to detect Hindi characters and language outcomes.

8.CONCLUSION

A. In this project we proposed different neural network approach for Recognition of Handwritten Hindi characters.

B. We evaluated the performance using Convolutional Neural Network (CNNs) withoptimization techniques and Deep Feed Forward NeuralNetwork.

C. These techniques are train and test on a standard userdefine dataset which is collect from different users. From experimental results, it is observed that DFFNN, CCN-Adamand CNN-RMSprop yield the best accuracy for HandwrittenHindi characters compared to the alternative techniques.

D. We achieved promising results from proposed method with highaccuracy rate .

9. REFERENCES

- Python: https://www.python.org/
- Pandaslibrary: https://pandas.pydata.org/
- Scikit-learn library: https://scikit-learn.org/
- Devnagri Charachter Recognition | Kaggle
- R.M.K. Sinha and Veena Bansal, "On Devnagari Document Processing".
- Vishwanatha Kaushik and C.V. Jawahar, "Detection of Devnagiri Text in digital images using Connected Component Analysis", National Conference on Document Analysis and Recognition(NCDAR0,pp.41-48.
- (PDF) Handwritten Hindi Character Recognition using Deep Learning Techniques (researchgate.net)
- Ciregan, D.; Meier, U.; Schmidhuber, J, "Multi-column deep neural networks for image classification", In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI,USA, 16-21 June 2012.
- Devanagari Character Recognition (slideshare.net)
- Multiple language detection using python - YouTube