**Optical Character Recognition**

**Contributors**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.no | Name | Roll number | Registration Number | Section |
| 1 | **ROHIT LALWANI** | **16** | **11801301** | **K18KK** |
| 2 | **VUTUKURI SMS MANIDEEP** | **15** | **11801361** | **K18KK** |

**Abstract**

Optical character recognition project is all about to convert a handwritten letter to a printed letter format. This is developed using Conventional Neural Network (CNN) technique of deep learning. The library used to implement CNN is TensorFlow. We used the Mnist’s characters and number data set to train the model such that it will be classifying handwritten characters or numbers accurately. It also includes paint API to take the input of handwritten characters and the output is displayed using tkinter interface window. This will have a wide range of application in the world like processing cheques automatically without human interpretation, image to text converter in software domain, etc.

**Related Work**

Bansal et al (2010)-This paper elaborates the segmentation of various irregular text words written in Gurumukhi script. This paper deals with the segmentation of words containing skewed, broken, irregular headline, touching and overlapped characters. Some of the new techniques like counter tracing methods are used along with horizontal and vertical projections.

Garg N. et.al (2011)-Character recognition is an important stage of any text recognition system. In Optical Character Recognition (OCR) system, the presence of half characters decreases the recognition rate. Due to touching of half character with full characters, the determination of presence of half character is very challenging task. In this paper, they have proposed new algorithm based on structural properties of text to segment the half characters in handwritten Hindi text. The results are shown for both handwritten Hindi text as well as for printed Hindi text. The proposed algorithm achieves the segmentation accuracy as 83.02% for half characters in handwritten text and 87.5% in printed text

Kumar et al. (2014)-This paper presents the segmentation of handwritten Gurumukhi characters is carried out defining the whole process for segmentation including digitization process and pre-processed techniques. Water Reservoir method is applied for identification and segmentation of touching characters.

**Implementation**

To implement the program of handwritten character recognition, we need to first install the requirements and we can do so by the following commands.

**For Windows:**

We need to have python3 installed along with pip. Once we have them, we can continue to install the modules required using the following command in Command Prompt in the project folder.

**pip3 install -r requirements.txt**

Once installed, we need to install the emnist dataset, so that we can train our model. To install the emnist dataset, run the “emnist-0.0-py3-none-any.whl” file provided in the project folder. Then we are ready to train our model by executing the following command.

**python3 train.py**

Now, we have trained our model and saved it to our disk. To scan the handwritten characters and convert them into 28x28 numpy array, we need to install Ghostscript and we can execute the file “ghostscript32.exe” or “ghostscript64.exe” based on the system architecture to install the same. Now we are ready to scan the handwritten characters and identify them. We can do that by executing the “main.py” python script as follows.

**python3 main.py**

**For Linux:**

Again, we need to have python3 installed along with pip. Then we can continue to install the other requirements by using the following command in Terminal in the project folder.

**pip3 install -r requirements.txt**

After the installation of these modules, we need to install the emnist dataset to train our model. To install the same, execute the following set of commands in Terminal.

**tar -xvzf emnist-0.0.tar.gz**

**cd emnist-0.0**

**sudo python3 setup.py install**

**cd ..**

After it is installed, we are ready to train our model by executing the “train.py” script as follows.

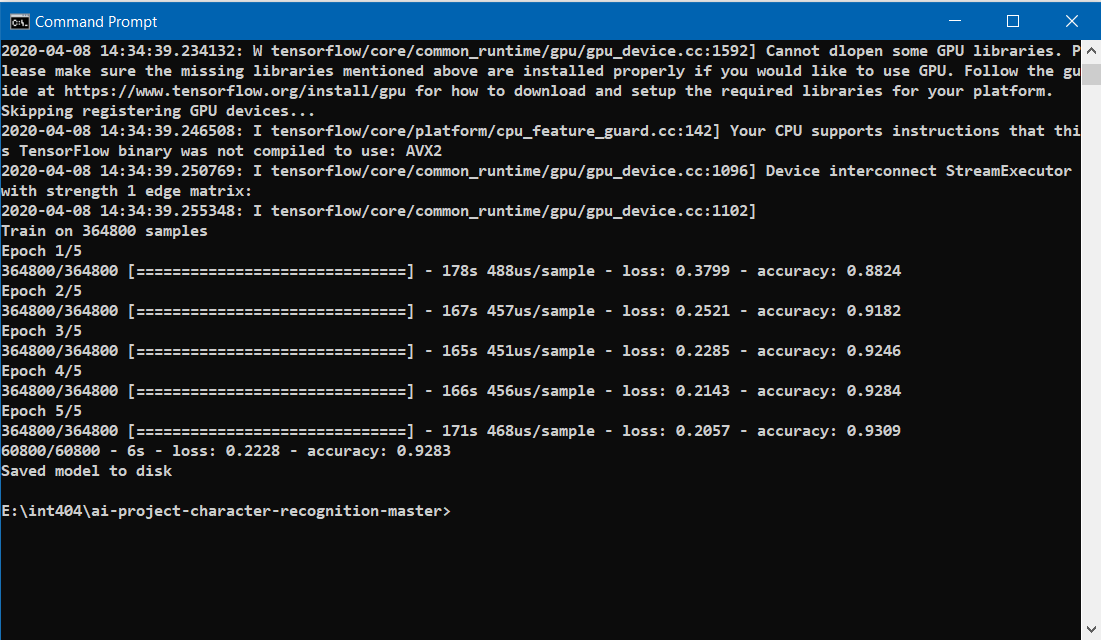
**python3 train.py**

As soon as the model is trained and saved to disk, we are good to go to scan the handwritten images and identify them using our trained model based on Deep Learning, by executing the “main.py” script.

**python3 main.py**

**Result**

Output of the given trained model based on Convolutional layers, in recognizing the alphabets and digits, by recognizing the patterns using saved weights which were obtained by training, is as follows.

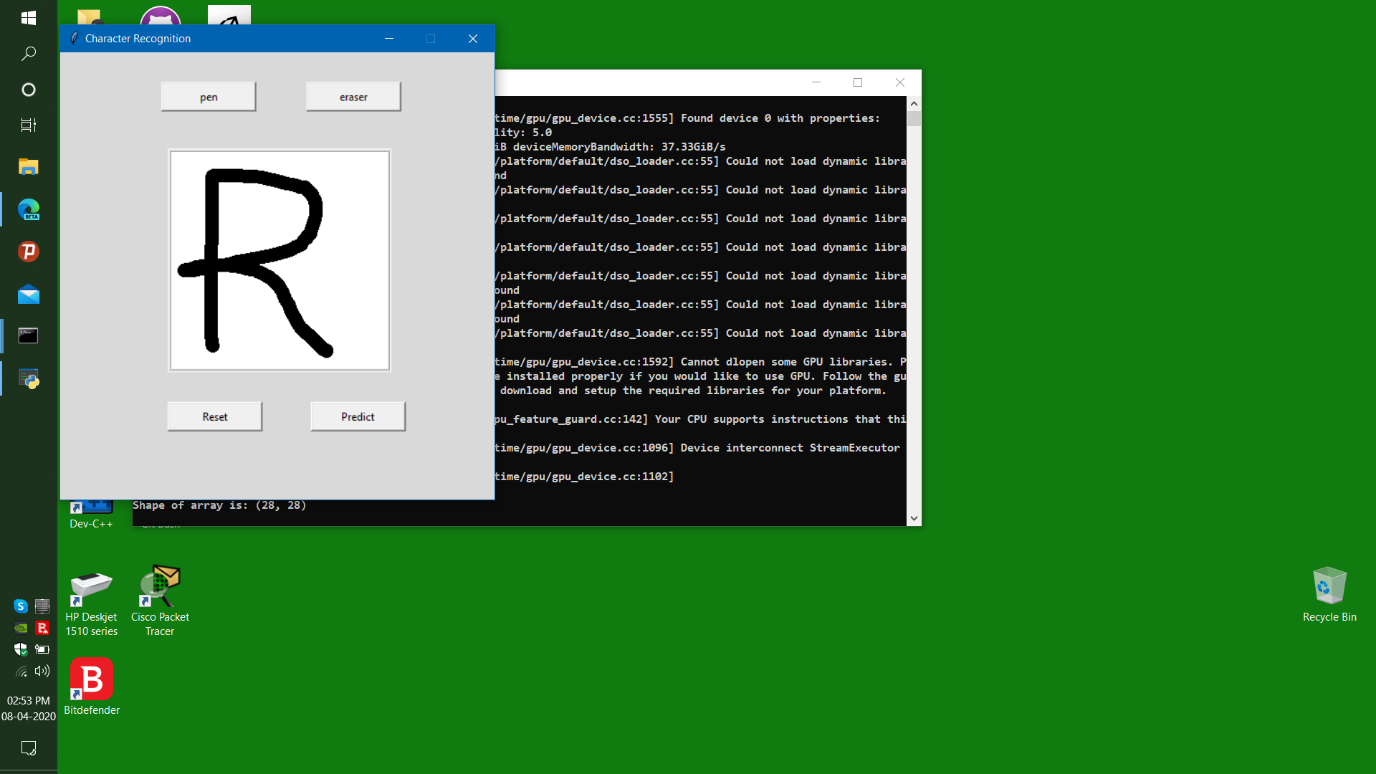


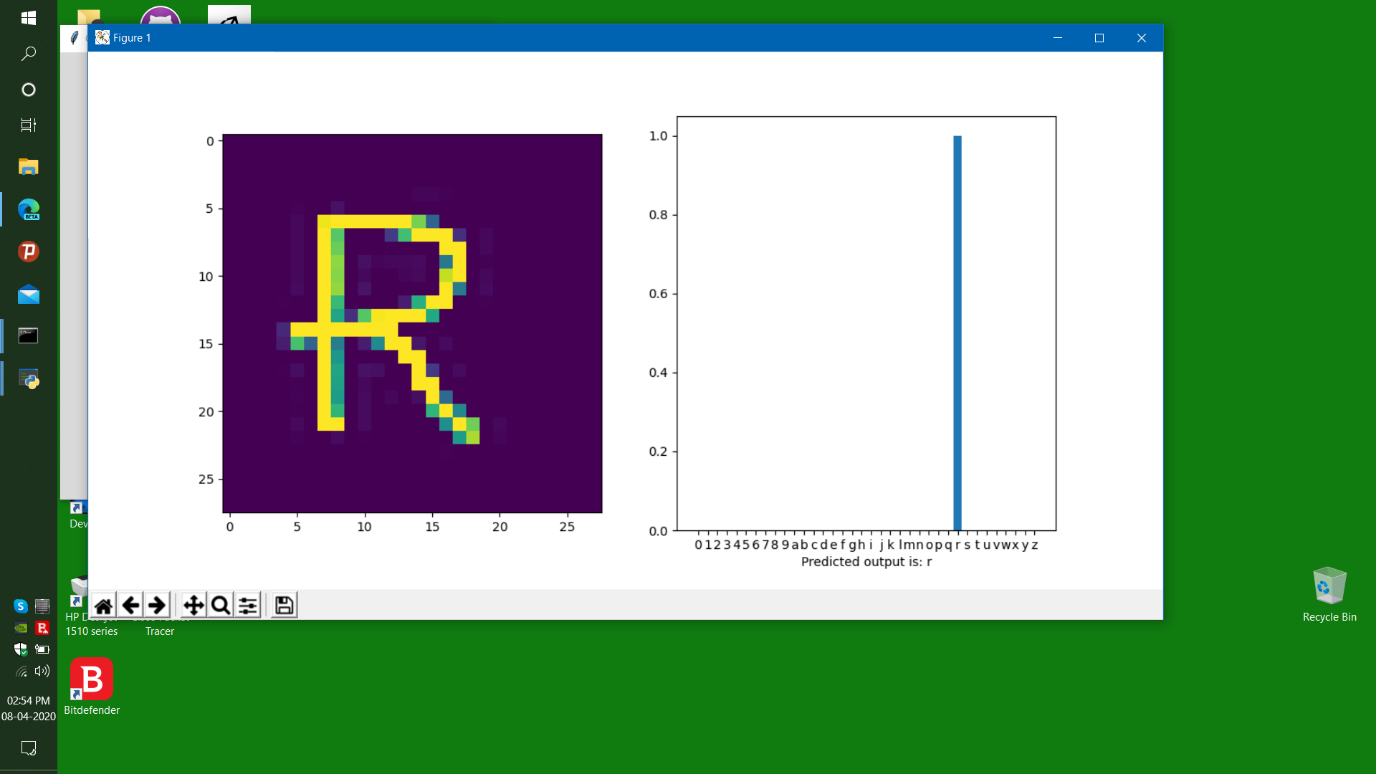
Epochs: 5

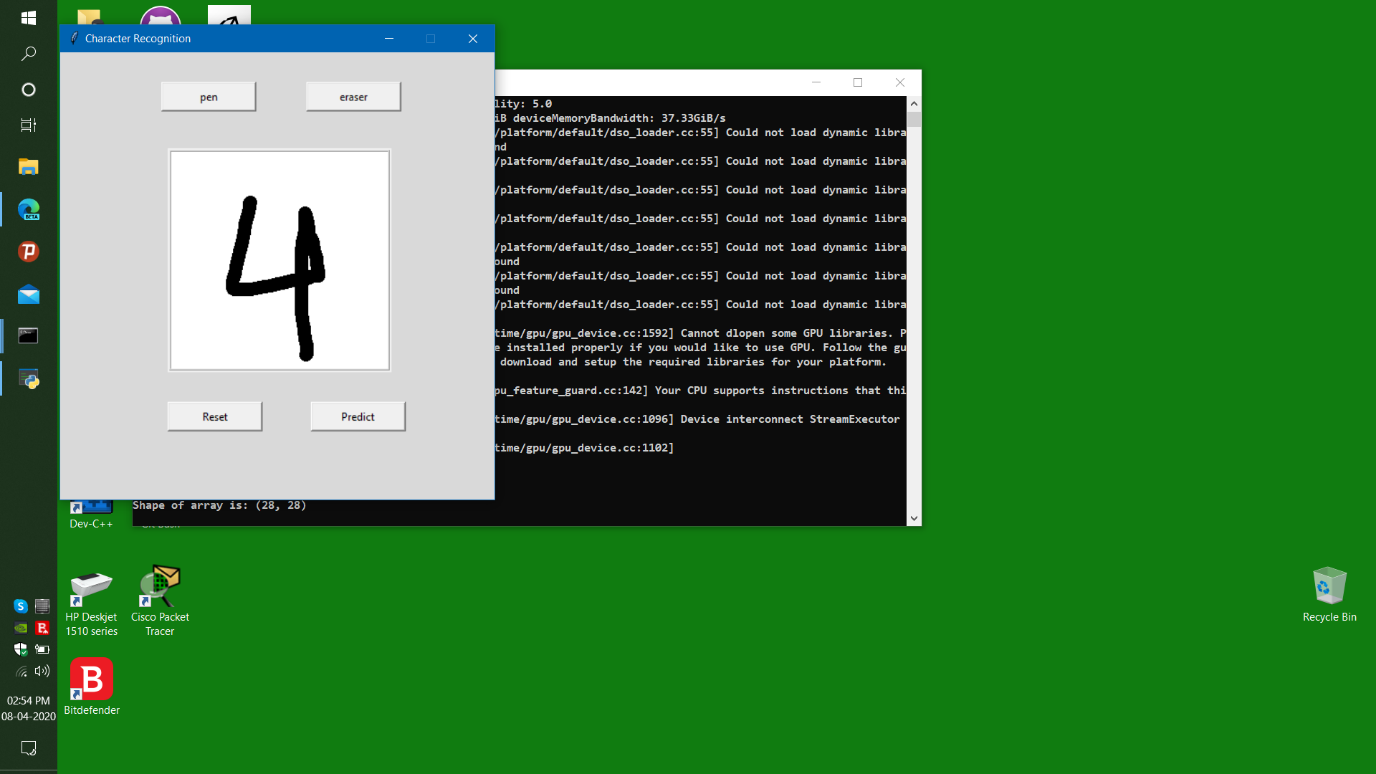
Training Accuracy: 93.09%

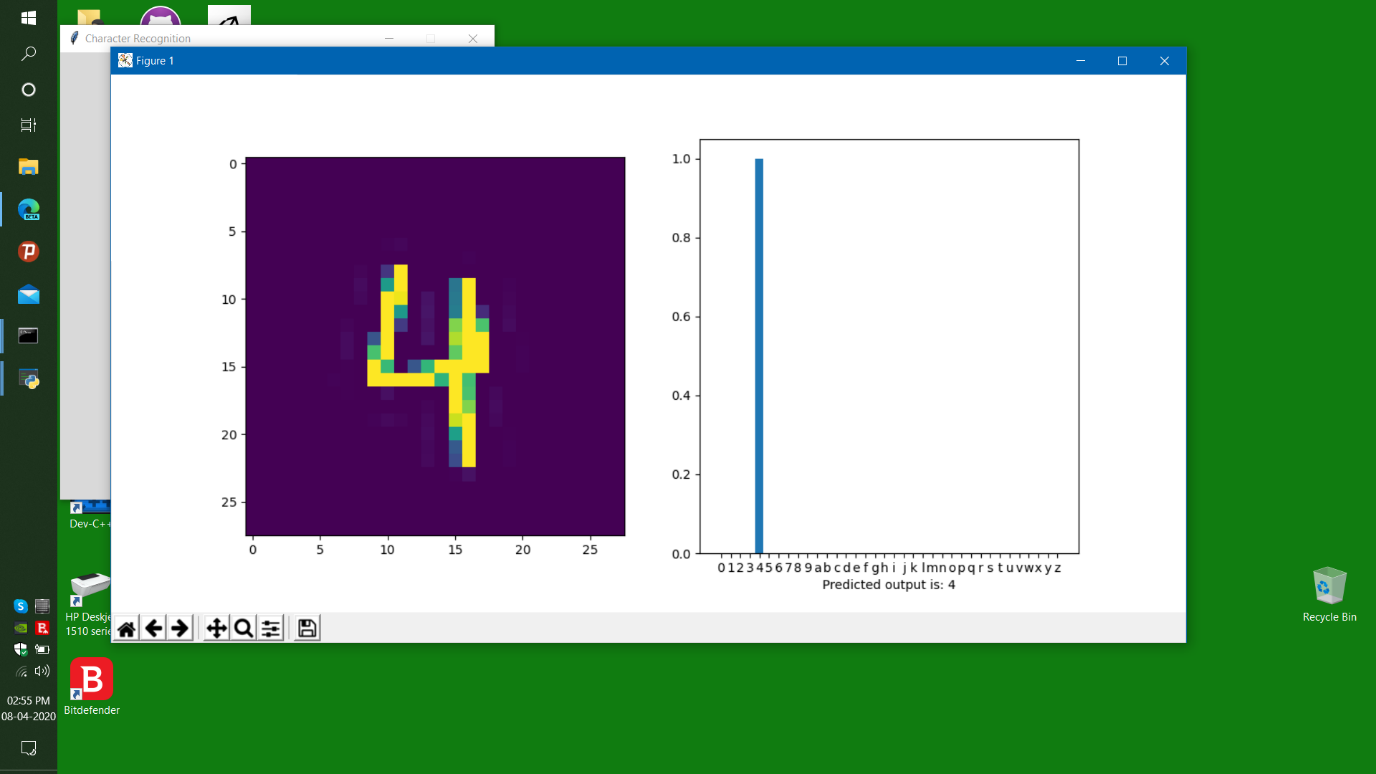
Testing Accuracy: 92.83%

The model was successful in recognizing the most of the alphabets and digits, given different handwritings, different way of writing them. Below are the example output using GUI.









**Important Libraries used**

**Tensorflow:**

It is the core module of this project, which consists of the definitions of all the neural network layers, models, optimizers and many more.

**Numpy:**

It is the module which consists of the functions which can be used to convert any data into n dimensional arrays so that we can analyse the same.

**Matplotlib:**

This module consists of the graph plotting definitions, which is very helpful in visualizing our data which helps us very much in analysing too.

**Emnist:**

One other core module of any project is the dataset. This module consists of our dataset, which contains different datasets including the required ones, that is ‘letters’ and ‘digits’. They contain training data as well as testing data so that we can test the accuracy of our model.

**Tkinter:**

It is the module that consists of the definitions of GUI and we can provide the user a better interactive interface (compared to command line) to write the characters (handwritten) and then identify them.

PIL:

To convert the input drawn in Canvas, we need some definitions/functions, which can be used to convert that into png image and then into 28x28 numpy array. This module consists of those functions/definitions.

**Work-Distribution**

**VUTUKURI SMS MANIDEEP –**

* Collection of data set i.e. mnist data set to train the model to predict the handwritten characters.
* Building model using CNN technique of deep learning model.

**ROHIT LALWANI-**

* Boosting the built model using various CNN boosting techniques.
* Building UI of the project using paint API and Tkinter.

**References**

<https://www.tensorflow.org>

<https://www.numpy.org>

<https://www.keras.io>

<https://www.matplotlib.org>s