CENTRE FOR DEVELOPMENT OF ADVANCED COMPUTING

(Ministry of Electronics and Information Technology (MeitY), India)

PROJECT REPORT ON

Development of Machine Learning Based Hand Written Digit Recognition System

Post Graduate Diploma
In
Big Data Analytics

SUBMITTED BY: -

Akshay Naxine - 220370625012	Amarja Kulkarni - 220370625015
Shubham Lokhande - 220370625024	Pratima Janardan - 220370625008
Santosh Murgude -220370625019	Sarmishtha Banik - 220370625017
Naveen Pandey - 220370625014	Akshay Bagul - 220370625009

CERTIFICATE OF APPROVAL

This is to Certify that the project report entitled-**Development of Machine Learning Based Hand Written Digit Recognition System** submitted by **Akshay Naxine, Shubham Lokhande, Santosh Murgude, Naveen Pandey, Amarja Kulkarni, Pratima Janardan, Sarmistha Banik & Akshay Bagul** is a bonafide work carried out by them under the supervision of **Mr. Asok Bandyopadhyay, Group Head,ICT&S C-DAC, Kolkata.** It is approved for the partial fulfilment of the requirement *Centre for Development of Advanced Computing (C-DAC) Kolkata*, for the completion of report. This project report has not been earlier submitted to any other Institute or University for the award of any degree or diploma.

Shri Debasis Muzumdar,	Mr. Asok Bandyopadhyay
Center Head,	Group Head, ICT&S,
	Associate Director
C-DAC, Kolkata	C-DAC, Kolkata

DECLARATION

We hereby declare that the project work entitled Development of Machine Learning Based Hand Written Digit Recognition System submitted to the Centre for Advance Computing(C-DAC), Kolkata, is a record of an original work done by us under the guidance of **Mr. Debabrata Pal ,** Principal Technical Officer, C-DAC, Kolkata and under the supervision of **Mr. Asok Bandyopadhyay, Group Head, ICT&S, Associate Director, C-DAC, Kolkata**, C-DAC, Kolkata. This project work has been performed for the award of ***Post Graduate Diploma in Big Data Analytics (PG-DBDA)*** course of C- DAC, Kolkata only and this or any similar project will not be used for any other Degree or Diploma's associateship / fellowship.

Signature: [illegible]	Signature: [illegible]
Name: Akshay Naxine	Name: Amarja Kulkarni
PRN No: 2203370625012	PRN No: 2203370625015
Signature: [illegible]	Signature: [illegible]
Name: Shubham Lokhande	Name: Pratima Janardan
PRN No: 2203370625024	PRN No: 2203370625008
Signature: [illegible]	Signature: [illegible]
Name: Santosh Murgude	Name: Sarmishtha Banik
PRN No: 2203370625019	PRN No: 2203370625017
Signature: [illegible]	Signature: [illegible]
Name: Naveen Pandey	Name: Akshay Bagul
PRN No: 2203370625014	PRN No: 2203370625009

ACKNOWLEDGEMENT

We express our utmost sincere regards and deep gratitude to our project guide **Mr. Debabrata Pal, Principal Technical Officer**, C-DAC, Kolkata for his valuable guidance towards us to navigate through the difficulty of project execution at all times**.**

We are very much thankful to **Mr. Asok Bandyopadhyay, Group Head, ICT&S, Associate Director, C-DAC, Kolkata**. We are also highly obliged to **Shri Debasis Mazumdar, Center Head, C-DAC, Kolkata** for their unconditional help and inspiration for giving us a wonderful opportunity to enhance our skills. Finally, we would like to extend our thanks to all thosewho have contributed, directly or indirectly to make this project successful.

[illegible]

Akshay Naxine

[illegible]

Amarja Kulkarni

[illegible]

Shubham Lokhande

[illegible]

Pratima Janardan

[illegible]

Santosh Murgude

Sarmistha Banik

Sarmishtha Banik

[illegible]

Naveen Pandey

[illegible]

Akshay Bagul

Table of Contents

CHAPTER 1

ABSTRACT

Traditional systems of handwritten Digit Recognition have depended on handcrafted functions and a massive amount of previous knowledge. Training an Optical character recognition (OCR) system primarily based totally on those stipulations is a hard task. Research in the handwriting recognition subject is centered on deep learning strategies and has accomplished breakthrough overall performance in the previous couple of years. Convolutional neural networks (CNNs) are very powerful in perceiving the structure of handwritten digits in ways that assist in automated extraction of features and make CNN the most appropriate technique for solving handwriting recognition problems. Here, our goal is to attain similar accuracy through the use of a pure CNN structure. CNN structure is proposed to be able to attain accuracy even higher than that of ensemble architectures, alongside decreased operational complexity and price. The proposed method gives 99.42 accuracy for real-world handwritten digit prediction with less than 0.1 % loss on training with 60000 digits while 10000 under validation.

INTRODUCTION

Character recognition is a fundamental, but most challenging in the field of pattern recognition with large number of useful applications. It has been an intense field of research since the early days of computer science due to it being a natural way of interactions between computers and humans. More precisely Character recognition is the process of detecting and recognizing characters from the input image and converts it into ASCII or other equivalent machine editable form

The project is to take a picture of a character and process it up to recognize the image of that character like a human brain recognize the various digits.The project contains the deep idea of the Image Processing techniques and the big research area of machine learning and the building block of the machine learning called Neural Network.There are two different parts of the project

1) Training part
2) Testing part.

Training part comes with the idea of to train a child by giving various sets of similar characters but not the totally same and to say them the output of this is “this”. Like this idea one has to train the newly built neural network with so many characters.This part contains some new algorithm which is self-created and upgraded as the project need.The testing part contains the testing of a new dataset .This part always comes after the part of the training .At first one has to teach the child how to recognize the character .Then one has to take the test whether he has given right answer or not. If not, one has to train him harder by giving new dataset and new entries. Just like that one has to test the algorithm also.

The technique by which a computer system can recognize characters and other symbols written by hand in natural handwriting is called handwriting recognition system. Handwriting recognition is classified into offline handwriting recognition and online handwriting recognition [3]. If handwriting is scanned and then understood by the computer, it is called offline handwriting recognition. In case, the handwriting is recognized while writing through touch pad using stylus pen, it is called online handwriting recognition. From the classifier perspective, character recognition systems are classified into two main categories i.e. segmentation free (global) and segmentation based (analytic).

CHAPTER 2

PROBLEM DEFINITION

The total world is working with the various problems of the machine learning.The goal of the machine learning is to factorize and to manipulate the real life data and the real life part of the human interaction or complex ideas or the problems in the real life.The most curious of those is Handwritten Character Recognition because it is the building block of the human certified and the classification interaction between other humans.

So, the goal was to create an appropriate algorithm that can give the output of the handwritten character by taking just a picture of that character. If one asks about Image processing then this problem can’t be solved because there can be a lot of noises in that taken image which can’t be controlled by human.The main thing is when human write a handwritten character or for our case digit he has no single idea whether he has to draw it in the circulated pixels or just same as a standard image given .A machine can do that but not the human.So by matching only the pixels one can’t recognize that.

The idea of machine learning lies on supervised data.Machine learning algorithm fully dependent on modeled data .If someone models the Image directly, the model will get a lot of flatten values because that picture can be drawn with various RGB format or with various pixels which can’t be modeled accurately due to noise.

So, for this project one has to create a model by image processing and the machine learning. Both the techniques will be needed because these two techniques will enhance the technique of the machine learning and that can shape this project.

Technology Used:

2.1.1 Jupyter_Notebook:

Jupyter Notebook is an open-source,web-based interactive environment, which allows you to create and share documents that contain live code, mathematical equations, graphics, maps, plots, visualizations, and narrative text. It intergrates with many programming languages like Python,PHP,R,C#,etc.

Components of Jupyter Notebook

There are the following **three** components of Jupyter Notebook -

1. **The notebook web application:**

It is an interactive web application for writing and running the code.

The notebook web application allows users to:

- Edit code in the browser with automatic syntax highlighting and indentation.
- Run code on the browser.
- See results of computations with media representations, such as HTML, LaTex, png, pdf, etc.
- Create and use JavaScript widgets.
- Includes mathematical equations using Markdown cells.

2. **Kernels:**

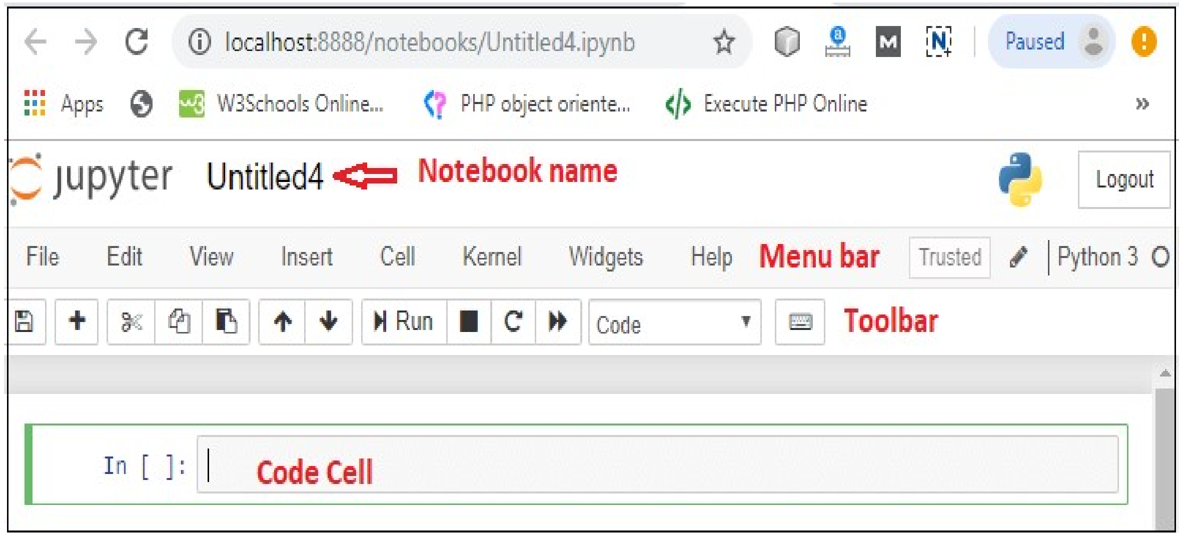
Kernels are the separate processes started by the notebook web application that is used to run a user's code in the given language and return output to the notebook web application.In Jupyter notebook kernel is available in the following languages:

- Python
- node.js
- Scala
- R

3. **Notebook documents:**

Notebook document contains a representation of all content which is visible in the notebook web application, including I nputs and outputs of the computations, text, mathematical equations, graphs, and images.

Jupyter Notebook UI:



2.1.2 Python:

- Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL). This tutorial gives enough understanding on Python programming language.
- Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- Python is a MUST for students and working professionals to become a great Software Engineer specially when they are working in Web Development Domain. I will list down some of the key advantages of learning Python:
    1. Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
    2. Python is Interactive − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
    3. Python is Object-Oriented − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
    4. Python is a Beginner's Language − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

2.2 Libraries Used:

2.2.1 Pandas: Pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series.

2.2.2 NumPy: NumPy is a library for the Python programming language, adding supportfor large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

2.2.3 Scikit Learn: It is a software library that allows us to use supervised and unsupervised learning algorithms for machine learning

CHAPTER 3

Model Details

3.1 IMAGE PROCESSING

Image processing technique has been implemented extensively at the very first part of the project.

So, what is Image Processing? Image processing is a method to perform some operations on an image, in order to get an enhanced image or to extract some useful information from it. It is a type of signal processing in which input is an image and output may be image or features or characteristics associated with that image. Nowadays, image processing is among rapidly growing technologies. It forms core research area within engineering and computer science disciplines too.

Image processing basically includes the following three steps:

1. Importing the image via image acquisition tools
2. Analyzing and manipulating the image
3. Output in which result can be altered image or report that is based on image analysis.

In this project at the last part of detection to take the input directly from the webcam, to reshape that image and in the very first part of training MNIST dataset image reshaping and real life dataset written reshaping, cutting, filtering all requires the idea of the image processing.

One by one the concepts of image processing in this project will be covered. As this project remains in two parts the impact of Image processing will be discussed in two parts:

1. Image processing in training data.
2. Image processing in testing data.

3.2 Image processing in training data

In the training dataset the neural network model has been trained with self created Dataset. This dataset almost adds 4500 entries in the training of the dataset and trains the model with excellence almost 5 times for iteration.

This dataset require a lot of Image processing works before sending into the neural model. There are various unlabelled factors in this image or dataset which needs to be controlled.

1. Multiple characters are given at random, need to cut them.
2. The real time Paint image can be of different shades.
3. Noisy image will come up; we need to make it noiseless.
4. One needs to resize the image depending on the quality loss.

As mentioned below as the various parts of Image Processing parts on the training data taken by self created dataset.

3.2.1 Characters written on a Paint

This step includes some writing of characters into a black and white Paint because that will be more efficient to detect, unless one can write in margin based boarder also.Writing with sketch pen or marker will be very nice because those create continuous characters.

3.2.2 Various algorithms applied for noise, filtering

This step is fully localised on cleaning the data,removing the noises and filter the data to get the actual subjectial data from the image.

3.2.3 Filtering

Filtering is the way to smoothen or sharpen an image.This process is done by removing very high or very low frequency cells.Low pass filters remove the higher frequencies by keeping only the lower frequencies.The work done by the low pass filter is to smoothen the picture and remove the noises.High pass filters remove the lower frequencies by keeping only the higher frequencies.The work done by high pass filter is to sharpen an image with no background(with black background).As the low filter used in noise removal, we have implemented it first.

3.3 Image processing in Testing Data

In the case of testing data at first I have thought it will be a picture which will be already in the database written by a human.To make the testing part more interesting, real time paint draw picture on the computer. This part has increased the complexity of the program and also of the project but made it more interesting.

3.3.1 Noise cancellation and resizing and scaling

From the Ipython display the value of the pixels has been copied to a numpy array and then using a 20,20 kernel the image has been designed. The normalisation of that image is done as previous training dataset entry and scaling and resizing done as the same.

The Image processing part is very important for this project because the total preprocessing of a machine learning algorithm lies through the part of the Image processing.The total unsupervised to supervised transformation can be done with the help of image processing. In the project, the preprocessing work concludes paint draw pictures , inputs of the pictures, scaling, resizing, filtering, noise cancellation and at last sending to the machine learning algorithm as a kernel for testing.

3.4 THE NEURAL NETWORK MODEL

Convolutional Neural Network (CNN) is a family of multi-layer neural networks it is particularly designed for use on two-dimensional data, such as images and videos. Basically, it is influenced by earlier work in time-delay neural networks, which reduce learning computation requirements by sharing weights in a temporal dimension and are intended for speech and time-series processing. It has many hierarchy layers to train in a robust manner.

This architecture that leverages spatial and temporal relationships to reduce the number of parameters which must be learned and thus improves upon general feed-forward backpropagation training. It is proposed as a deep learning framework that is motivated by minimal data preprocessing requirements. In CNN, small portions of the image are treated as inputs to the lowest layer of the hierarchical structure. The fully connected layers form a network in this first layer is named as “input layer” and the last layer are named as “output layer” and between these two layers and remaining all are known as “hidden layers”.

In the hidden layer, the inputs were passed and the output layer calculates the class probabilities for the classification. A regular neural network performs high computation if the size of data is increased or if the number of layers are increased. In these many parameters were calculated without overfitting results for accuracy.

On the other hand, CNN not fully connected in all levels of layers. The neurons in these layer connected to a small region. This encourages the local spatial relationship in the data and the hierarchy features to increase the abstraction from low-level to high-level when multiple were layers are stacked. In these first layers can see only a small portion of the input data and the last layers can see the whole of the input data and draw conclusions from it.

In CNN there are three types of layers were used in a convolutional network:

3.4.1 Convolutional layer:

In this layer few parameters like a number of filters, size of filters, stride, etc. Small window filter slid along with the dimensions of input data and performs dot products between the values stored in the filter and the input data points. Pooling layer: This layer reduces the dimensionality of the input data which reduces the computations, number of parameters and therefore reduces overfitting. Typically, the pooling layer is inserted between convolutional layers. It discards the activations of previous layers and hence forcing the next convolutional layers to learn from a limited variety of data.

3.4.2 Pooling layer:

This layer reduces the dimensionality of the input data which reduces the computations, number of parameters and therefore reduces overfitting. Typically, the pooling layer is inserted between convolutional layers. It discards the activations of previous layers and hence forcing the next convolutional layers to learn from a limited variety of data.

3.4.3 Fully-connected layer:

In this layer neurons that are connected to all neurons of the previous layer as explained it.

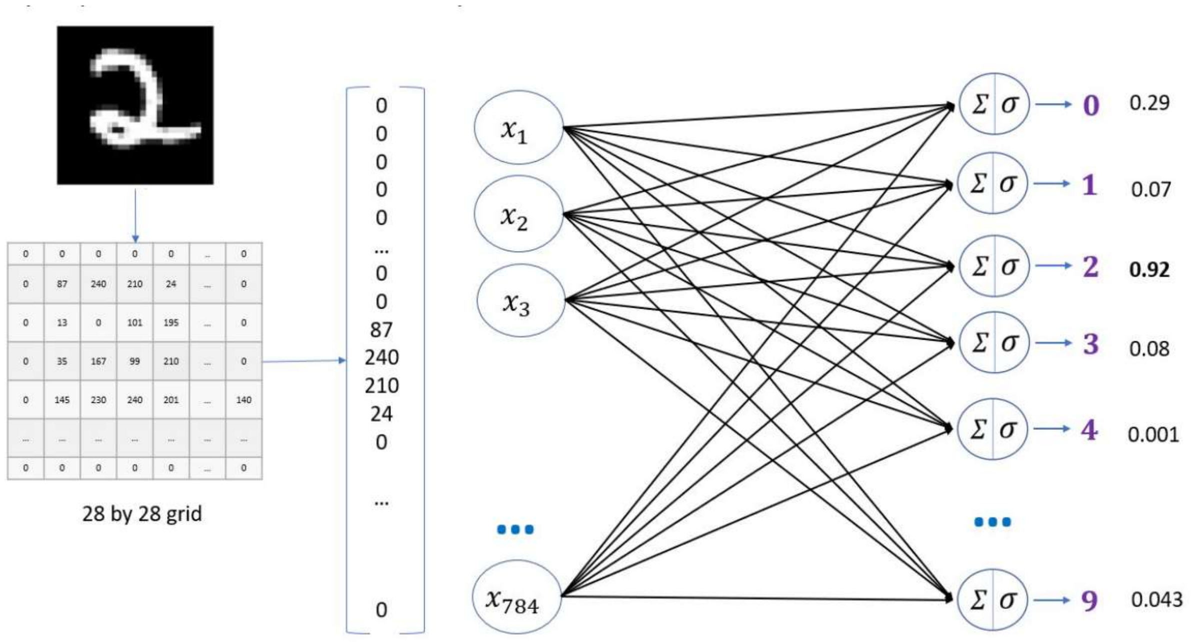


Fig- Neural Network Model

CHAPTER 4

FLOW CHARTS AND FIGURES

FLOW CHARTS

A flowchart is a type of diagram that represents a workflow or process.A flowchart can also be defined as diagrammatic representation of an algorithm, a step by step approach.

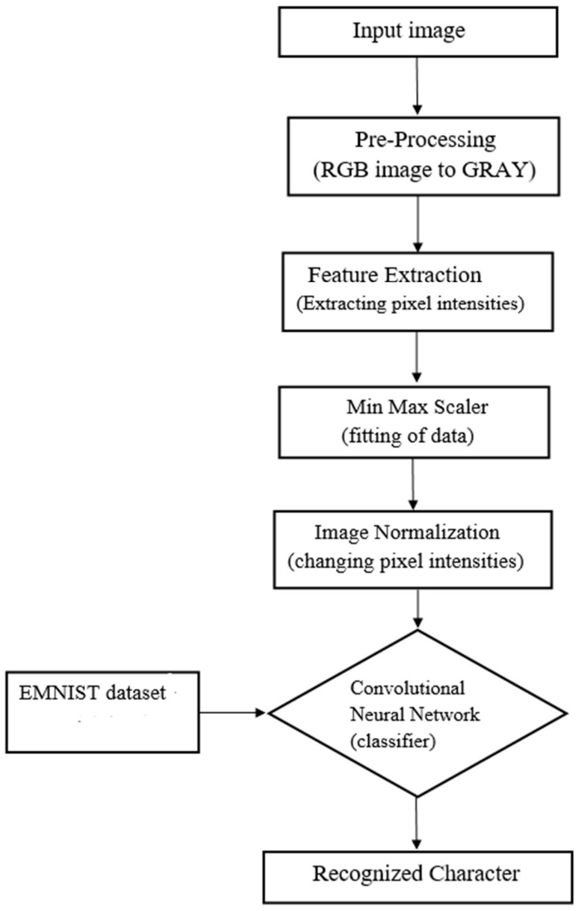

Input image
Pre-Processing (RGB image to GRAY)
Feature Extraction (Extracting pixel intensities)
Min Max Scaler (fitting of data)
Image Normalization (changing pixel intensities)
EMNIST dataset
Convolutional Neural Network (classifier)
Recognized Character

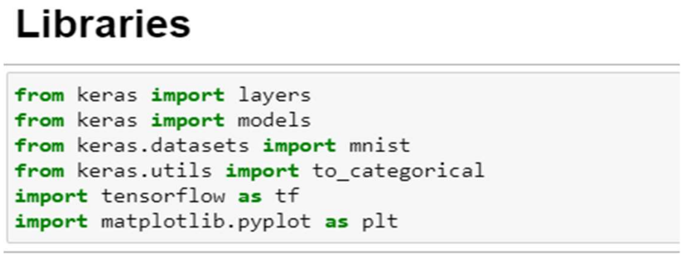
Image processing for Training Data

Neural Network Model gets the prediction from the testing image.Thanks to the Image processing part for the testing image that actually plots the data in the image into the pixels calculated the probabilistic values by sending it into the neural network.If matched then it is a successful one if not then it is an error. May be training will be done with it later. The above figure depicts the flow of the processes.

Chapter 5
Implementation

5.1 Importing Libraries

At first, we have imported all the required libraries:



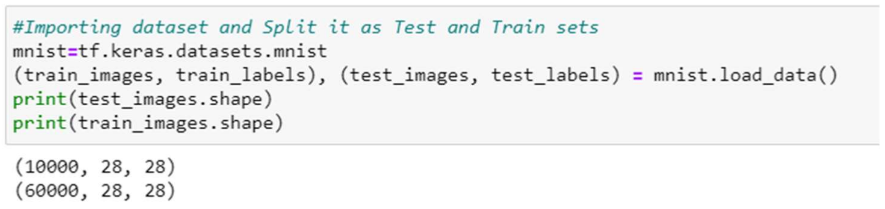
Libraries

```
from keras import layers
from keras import models
from keras.datasets import mnist
from keras.utils import to_categorical
import tensorflow as tf
import matplotlib.pyplot as plt
```

5.1.1 Keras : - Keras is an open-source high-level Neural Network library, It not only supports Convolutional Networks and Recurrent Networks individually but also their combination.

5.1.2 Tensorflow :- Tensorflow is a library that is used in machine learning and it is an open-source library for numerical computation

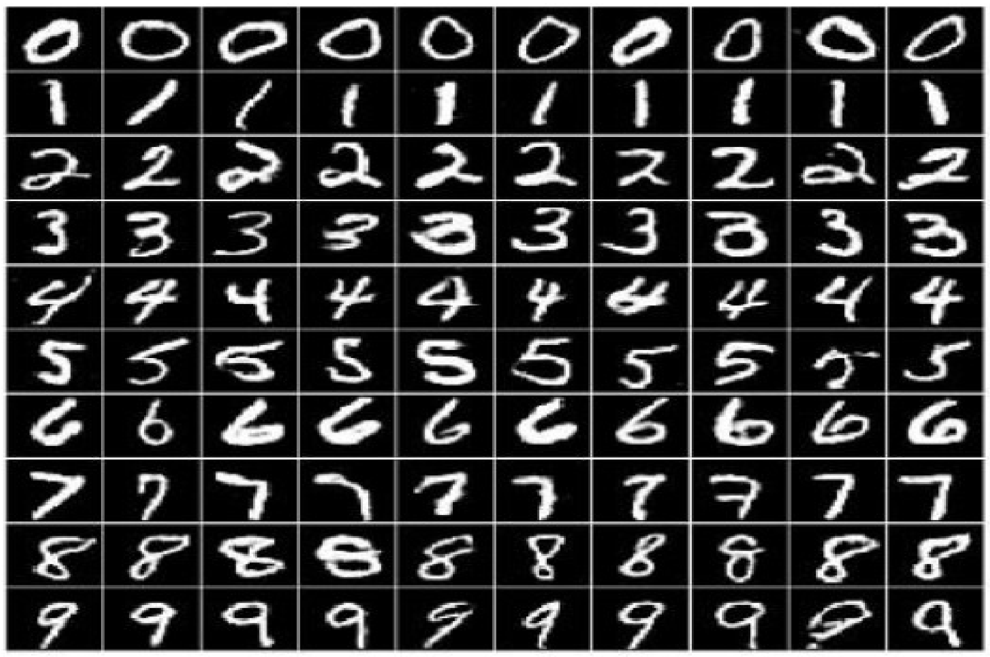
5.2 Importing mnist Dataset and splitting it into train and test (6:1)



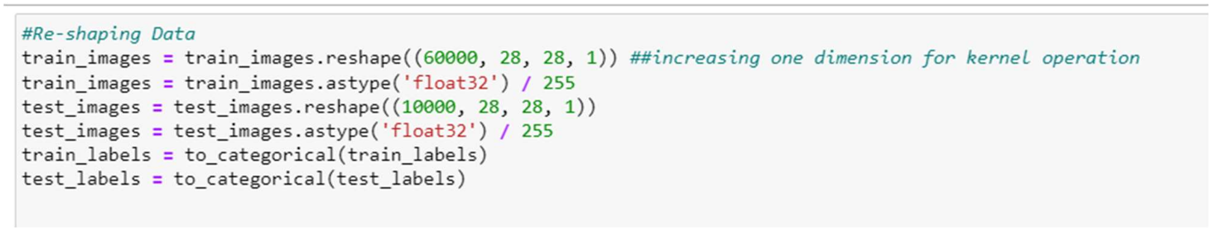
```
#Importing dataset and Split it as Test and Train sets
mnist=tf.keras.datasets.mnist
(train_images, train_labels), (test_images, test_labels) = mnist.load_data()
print(test_images.shape)
print(train_images.shape)
```

```
(10000, 28, 28)
(60000, 28, 28)
```

The MNIST database (Modified National Institute of Standards and Technology database) is a large collection of handwritten digits. It has a training set of 60,000 examples, and a test set of 10,000 examples.

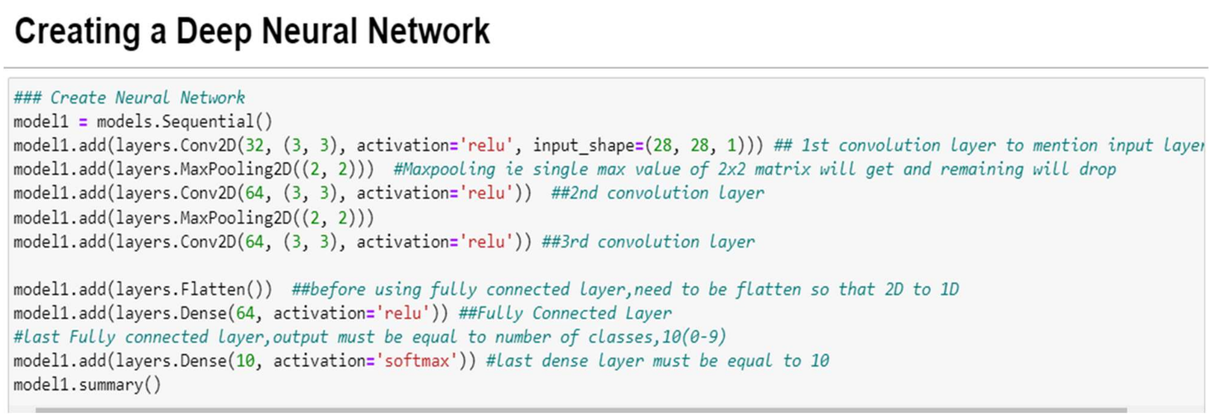


5.3 Re-shaping the data and converting it into matrix:



```
#Re-shaping Data
train_images = train_images.reshape((60000, 28, 28, 1)) ##increasing one dimension for kernel operation
train_images = train_images.astype('float32') / 255
test_images = test_images.reshape((10000, 28, 28, 1))
test_images = test_images.astype('float32') / 255
train_labels = to_categorical(train_labels)
test_labels = to_categorical(test_labels)
```

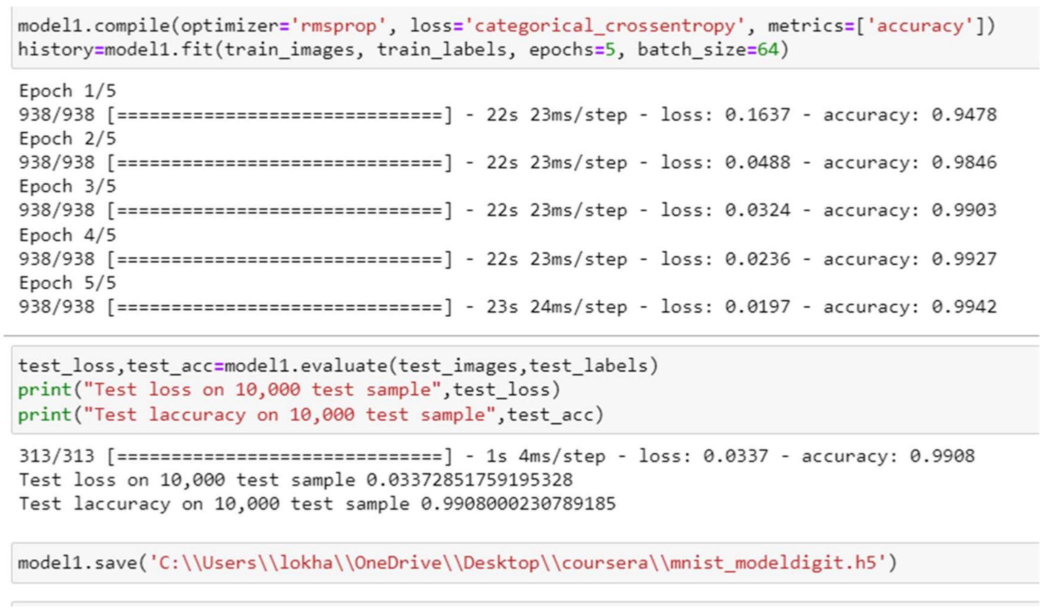
5.4 Creating a CNN Model:



Creating a Deep Neural Network

```
### Create Neural Network
model1 = models.Sequential()
model1.add(layers.Conv2D(32, (3, 3), activation='relu', input_shape=(28, 28, 1))) ## 1st convolution layer to mention input Layer
model1.add(layers.MaxPooling2D((2, 2)))  #Maxpooling ie single max value of 2x2 matrix will get and remaining will drop
model1.add(layers.Conv2D(64, (3, 3), activation='relu'))  ##2nd convolution layer
model1.add(layers.MaxPooling2D((2, 2)))
model1.add(layers.Conv2D(64, (3, 3), activation='relu')) ##3rd convolution layer

model1.add(layers.Flatten())  ##before using fully connected layer,need to be flatten so that 2D to 1D
model1.add(layers.Dense(64, activation='relu')) ##Fully Connected Layer
#last Fully connected layer,output must be equal to number of classes,10(0-9)
model1.add(layers.Dense(10, activation='softmax')) #last dense layer must be equal to 10
model1.summary()
```

5.5 Model Evaluation & Saving it as mnist_modeldigit.h5:



```
model1.compile(optimizer='rmsprop', loss='categorical_crossentropy', metrics=['accuracy'])
history=model1.fit(train_images, train_labels, epochs=5, batch_size=64)
```

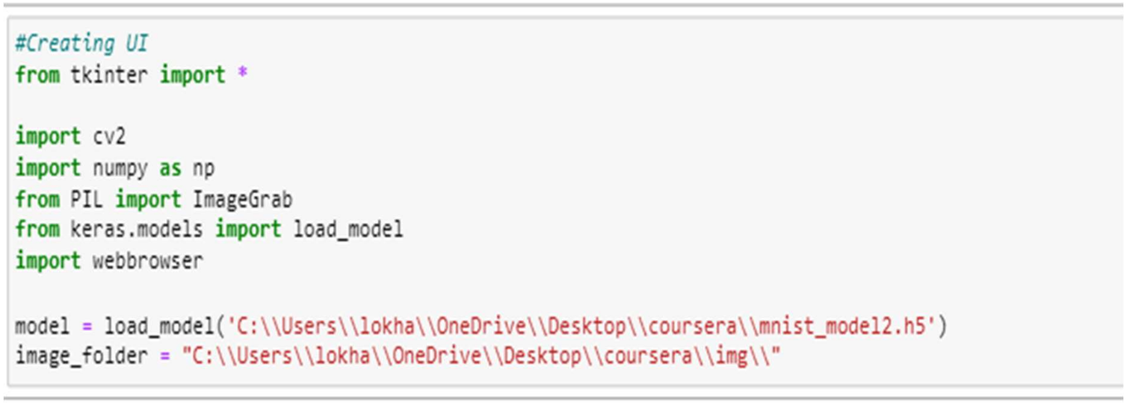
```
Epoch 1/5
938/938 [==============================] - 22s 23ms/step - loss: 0.1637 - accuracy: 0.9478
Epoch 2/5
938/938 [==============================] - 22s 23ms/step - loss: 0.0488 - accuracy: 0.9846
Epoch 3/5
938/938 [==============================] - 22s 23ms/step - loss: 0.0324 - accuracy: 0.9903
Epoch 4/5
938/938 [==============================] - 22s 23ms/step - loss: 0.0236 - accuracy: 0.9927
Epoch 5/5
938/938 [==============================] - 23s 24ms/step - loss: 0.0197 - accuracy: 0.9942
```

```
test_loss,test_acc=model1.evaluate(test_images,test_labels)
print("Test loss on 10,000 test sample",test_loss)
print("Test laccuracy on 10,000 test sample",test_acc)
```

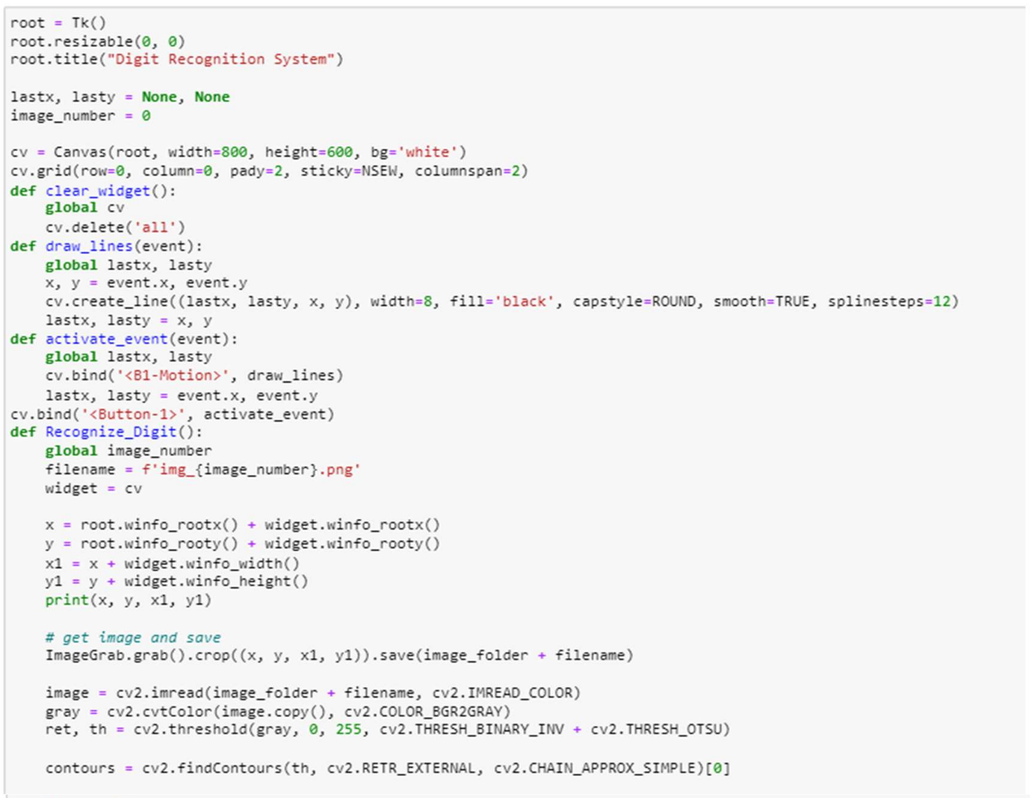
```
313/313 [==============================] - 1s 4ms/step - loss: 0.0337 - accuracy: 0.9908
Test loss on 10,000 test sample 0.03372851759195328
Test laccuracy on 10,000 test sample 0.9908000230789185
```

```
model1.save('C:\\Users\\lokha\\OneDrive\\Desktop\\coursera\\mnist_modeldigit.h5')
```

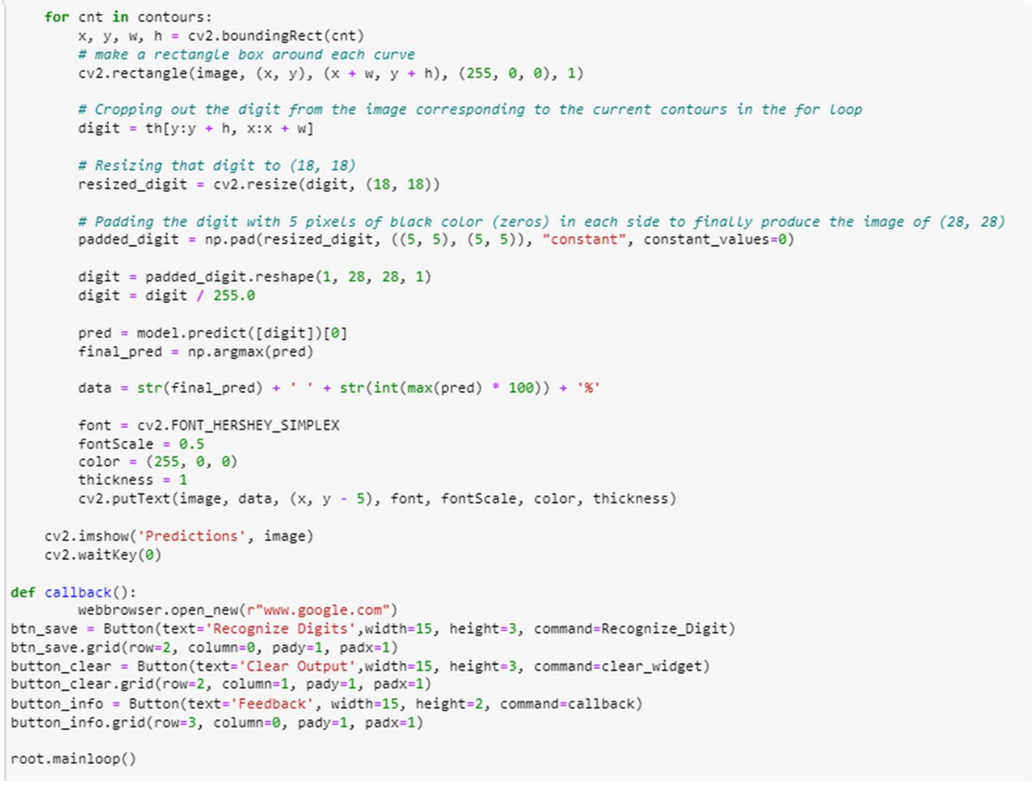
5.6 Creating a UI:



```
#Creating UI
from tkinter import *

import cv2
import numpy as np
from PIL import ImageGrab
from keras.models import load_model
import webbrowser

model = load_model('C:\\Users\\lokha\\OneDrive\\Desktop\\coursera\\mnist_model2.h5')
image_folder = "C:\\Users\\lokha\\OneDrive\\Desktop\\coursera\\img\\"
```



```
root = Tk()
root.resizable(0, 0)
root.title("Digit Recognition System")

lastx, lasty = None, None
image_number = 0

cv = Canvas(root, width=800, height=600, bg='white')
cv.grid(row=0, column=0, pady=2, sticky=NSEW, columnspan=2)
def clear_widget():
    global cv
    cv.delete('all')
def draw_lines(event):
    global lastx, lasty
    x, y = event.x, event.y
    cv.create_line((lastx, lasty, x, y), width=8, fill='black', capstyle=ROUND, smooth=TRUE, splinesteps=12)
    lastx, lasty = x, y
def activate_event(event):
    global lastx, lasty
    cv.bind('<B1-Motion>', draw_lines)
    lastx, lasty = event.x, event.y
cv.bind('<Button-1>', activate_event)
def Recognize_Digit():
    global image_number
    filename = f'img_{image_number}.png'
    widget = cv

    x = root.winfo_rootx() + widget.winfo_rootx()
    y = root.winfo_rooty() + widget.winfo_rooty()
    x1 = x + widget.winfo_width()
    y1 = y + widget.winfo_height()
    print(x, y, x1, y1)

    # get image and save
    ImageGrab.grab().crop((x, y, x1, y1)).save(image_folder + filename)

    image = cv2.imread(image_folder + filename, cv2.IMREAD_COLOR)
    gray = cv2.cvtColor(image.copy(), cv2.COLOR_BGR2GRAY)
    ret, th = cv2.threshold(gray, 0, 255, cv2.THRESH_BINARY_INV + cv2.THRESH_OTSU)

    contours = cv2.findContours(th, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)[0]
```



```
    for cnt in contours:
        x, y, w, h = cv2.boundingRect(cnt)
        # make a rectangle box around each curve
        cv2.rectangle(image, (x, y), (x + w, y + h), (255, 0, 0), 1)

        # Cropping out the digit from the image corresponding to the current contours in the for loop
        digit = th[y:y + h, x:x + w]

        # Resizing that digit to (18, 18)
        resized_digit = cv2.resize(digit, (18, 18))

        # Padding the digit with 5 pixels of black color (zeros) in each side to finally produce the image of (28, 28)
        padded_digit = np.pad(resized_digit, ((5, 5), (5, 5)), "constant", constant_values=0)

        digit = padded_digit.reshape(1, 28, 28, 1)
        digit = digit / 255.0

        pred = model.predict([digit])[0]
        final_pred = np.argmax(pred)

        data = str(final_pred) + ' ' + str(int(max(pred) * 100)) + '%'

        font = cv2.FONT_HERSHEY_SIMPLEX
        fontScale = 0.5
        color = (255, 0, 0)
        thickness = 1
        cv2.putText(image, data, (x, y - 5), font, fontScale, color, thickness)

    cv2.imshow('Predictions', image)
    cv2.waitKey(0)

def callback():
        webbrowser.open_new(r"www.google.com")
btn_save = Button(text='Recognize Digits',width=15, height=3, command=Recognize_Digit)
btn_save.grid(row=2, column=0, pady=1, padx=1)
button_clear = Button(text='Clear Output',width=15, height=3, command=clear_widget)
button_clear.grid(row=2, column=1, pady=1, padx=1)
button_info = Button(text='Feedback', width=15, height=2, command=callback)
button_info.grid(row=3, column=0, pady=1, padx=1)

root.mainloop()
```

About Tkinter:

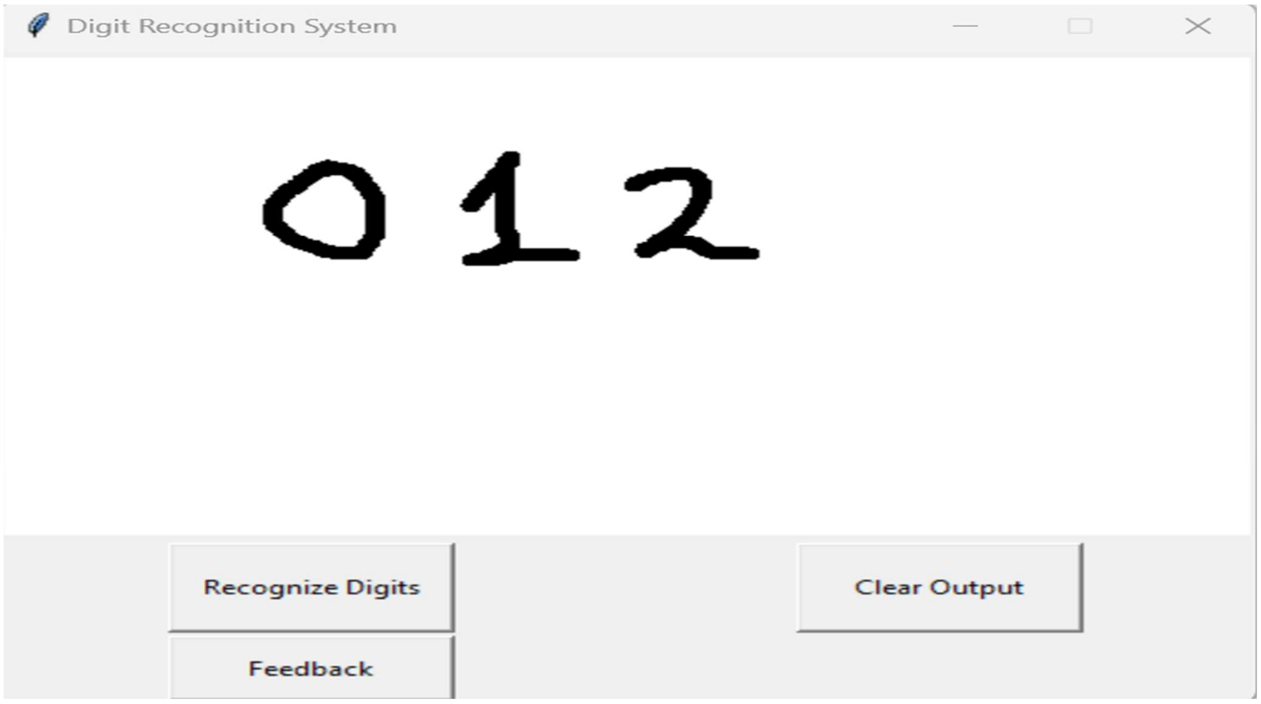
Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

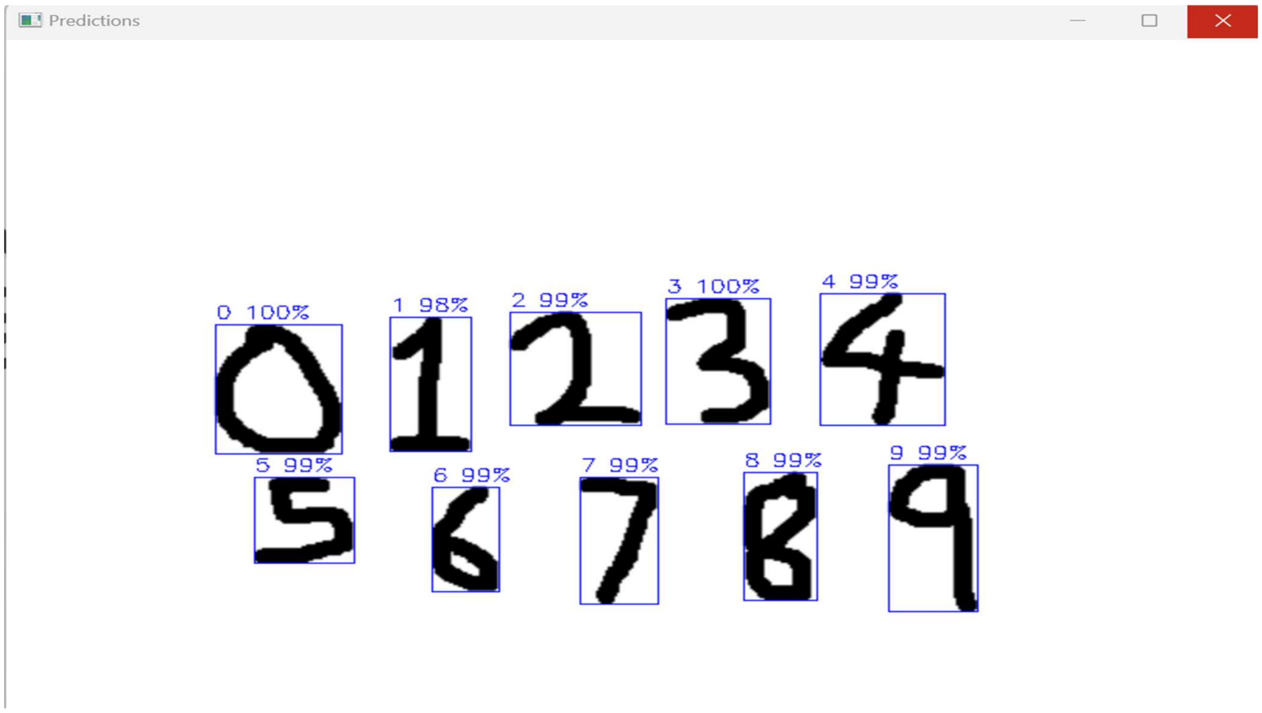
Creating a GUI application using Tkinter is an easy task. All you need to do is perform the following steps –

- Import the *Tkinter* module.
- Create the GUI application main window.
- Add one or more of the above-mentioned widgets to the GUI application.
- Enter the main event loop to take action against each event triggered by the user.

5.7 Interface

Here below we show the interface design how it look like.





Future Scope

This project can be enhanced with a great field of machine learning and arfiticial intelligence. The world can think of a software which can recognise the text from a picture and can show it to the others, for example a the shop name detector. Or this project can be extended to a greater concept of all the character sets in the world. Think of a AI modeled car sensor going with a direction modeling in the roadside, user shall give only the destination.All of these enhancement is an application of the texture analysis where advanced image processing,Neural network model for training and advanced AI concepts will come.These applications can be modeled further .As this project is fully done by free and available resources and packages this can be also a limitation of the project.

The fund is very important because all machine learning libraries and advanced packages are not available for free.Unless of those the most of the visualizing platforms like on which developers are doing some works like Watson Studio or Aws.These all are mainly paid platforms where a lot of ML projects are going on.

Conclusion

We would like to thank all our faculties of CDAC Kolkata for giving us this opportunity to work on such a lucrative project which not only enhanced our knowledge but also helped us in touching the lives of people in a good way. The learning of this project will be carried forward by us to our professional life and we will try to build upon the learnings and direction given to us by our beloved faculties in the long run.

Reference

[1] Non-recursive Thinning Algorithms using Chain Codes Paul C K Mwok Department of Computer Science The University of Calgary Calgary, Canada T2N 1N4

[2] A dynamic shape preserving thinning algorithm Louisa Lam and Ching Y. Suen Centre for Pattern Recognition and Machine Intelligence and Department of Computer Science, Concordia University, 1455 de Maisonneuve Blvd. W., Montrdal, Qudbec H3G 1MS, Canada

[3] Object Contour Detection with a Fully Convolutional Encoder-Decoder Network Jimei Yang Adobe Research jimyang@adobe.com Brian Price Adobe Research bprice@adobe.com Scott Cohen Adobe Research scohen@adobe.com Honglak Lee University of Michigan, Ann Arbor honglak@umich.edu Ming-Hsuan Yang UC Merced mhyang@u

[4] Contour Detection and Image Segmentation by Michael Randolph Maire B.S. (California Institute of Technology) 2003

[5] Three-Dimensiaonal Nonlinear invisible Boundary detection ,IEEE Transaction on Image Processing VassiliKovalev,J,Chen

[6] Unconstrained OCR for Urdu using Deep CNN – RNN Hybrid Networks; Mohit Jain, Minesh Mathew et al.

[7] Neural Network and Deep Learning by Michael Nielsen.

[8] How to implement a Neural Network intermezzo 2, Peter Roelants(2016)

[9] Comparative analysis of methods used to remove salt and pepper noise IJCSMC Journal ZiadAlquadi,Eng. Mahmoud Alleddawi

[10] Understanding Convolutional Neural Network with a Mathematical model, C.C.JAY.KUO(2016)

[11] Delving deep into Rectifiers: Surpassing Human level performance on Image Net Classification, Kaiming He et al.