PROJECT REPORT

A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION

submitted by

PNT2022TMID08626

Srinithi Nivashini K	-	727619BCS066
Thulasimathi T	-	727619BCS026
Sujaykanth S	-	727619BCS054
Rudrapati Thrivendra Naidu	-	727619BCS108

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

Machine learning and deep learning play an important role in computer technology and artificial intelligence. With the use of deep learning and machinelearning, human effort can be reduced in recognizing, learning, predictions, and inmany more areas.

Handwritten Digit Recognition is the ability of computer systems to recognise handwritten digits from various sources, such as images, documents, and so on. Thisproject aims to let users take advantage of machine learning to reduce manual tasksin recognizing digits.

1.2 PURPOSE

Digit recognition systems are capable of recognizing the digits from differentsources like emails, bank cheques, papers, images, etc. and in different real-world scenarios for online handwriting recognition on computer tablets or systems, recognizing number plates of vehicles, processing bank cheque amounts, numeric entries in forms filled up by hand (tax forms) and so on.

CHAPTER 2

LITERATURE SURVEY

2.1 EXISTING PROBLEM

The fundamental problem with handwritten digit recognition is that handwritten digits do not always have the same size, width, orientation, and marginssince they vary from person to person. Additionally, there would be issues with identifying the numbers because of similarities between numerals like 1 and 7, 5 and6, 3 and 8, 2 and 5, 2 and 7, etc. Finally, the individuality and variation of each individual's handwriting influence the structure and appearance of the digits.

2.2 REFERENCES

Handwritten digits recognition with artificial neural network

Authors: K.Islam , G. Mujtaba, H.F.Nweke, R.G.Raj

In a computer vision system, handwritten digit recognition is a complex task central to various emerging applications. It has been widely used by machine learning and computer vision researchers for implementing practical applications like computerized bank check numbers reading. In this study, we implement a multi-layer fully connected neural network with one hidden layer for handwritten digit recognition. The testing has been conducted from a publicly available MNIST handwritten database. From the MNIST database, we extracted 28,000 digits images for training and 14,000 images for performing the test. Our multi-layer artificial neural network has an accuracy of 99.60% with the test performance.

Handwritten digit recognition using various neural network approaches

Authors: Sakshi ca, K. Gupta

Handwritten digit recognition is one of the important problems in computer vision these days. There is a great interest in this field because of many potential applications, most importantly where a large number of documents must be detailed such as post mail sorting, bank

cheque analysis, handwritten form processing, etc. so a system should be designed in such a way that it is capable of reading handwritten digits and provide appropriate results. This paper presents a survey on various neural approaches to recognizing handwritten digits.

Handwritten Digit Recognition of MNIST dataset using Deep Learning state-of-the-art Artificial Neural Network (ANN) and Convolutional Neural Network (CNN)

Authors: Drishti Beohar, A. Rasool

Handwritten digit recognition is an intricate assignment that is vital for developing applications, in computer vision digit recognition is one of the major applications. There has been a copious exploration done in the Handwritten Character Recognition utilizing different deep learning models. Deep learning is rapidly increasing in demand due to its resemblance to the human brain. The two major Deep learning algorithms Artificial Neural Network and Convolutional Neural Network have been compared in this paper considering their feature extraction and classification stages of recognition. The models were trained using categorical cross-entropy loss and ADAM optimizer on the MNIST dataset. Backpropagation along with Gradient Descent is being used to train the networks along with reLU activations in the network which do automatic feature extraction. In neural networks, Convolution Neural Network (Conv Nets or Convolutional neural networks) is one of the primary classifiers to do image recognition, and image classification tasks in Computer Vision.

Neural Network methods for handwritten digit Recognition provided with MNIST data set

Authors: Lars Kai, Christian

In Handwritten Digit Recognition, there are different challenges faced while attempting to solve this problem. The handwritten digits are not always of the same size, thickness, or orientation and position relative to the margins. Our goal was to implement a pattern classification method to recognize the handwritten digits provided in the MINIST data set of images of hand written digits (0 to 9). The data set used for our application is composed of 300 training images and 300 testing images, and is a subset of the MNIST data set (originally composed of 60,000 training images and 10,000 testing images). Each image is a 28x28 grayscale (0t o 255) labelled representation of an individual digit.

Extract data from image using Convolutional Neural Networks

Author: Gaganashree J

The main purpose of digital identifier extraction is to eliminate data redundancy and use set of digital attributes to get a more effective realization of word images. The point is to extract most of the important information from the original image data. In addition, the curve should not be as smooth as printed characters. In addition, characters dataset can be drawn in different sizes. This should always be written on the guide in a straight or vertical position. The task of recognizing handwritten digits with the help of a classifier is particularly important for the following applications such as online digit recognition on a tablet computer, recognize zip codes on mail, processing bank check applications. Various problems arise when trying to solve this problem. The size, thickness, direction, and position relative to the edge of handwritten numbers are not always the same. The main goal is to update the method of describing perception patterns characterization method to Note the handwritten digits represented in the MNIST Handwritten Digit Image Record.

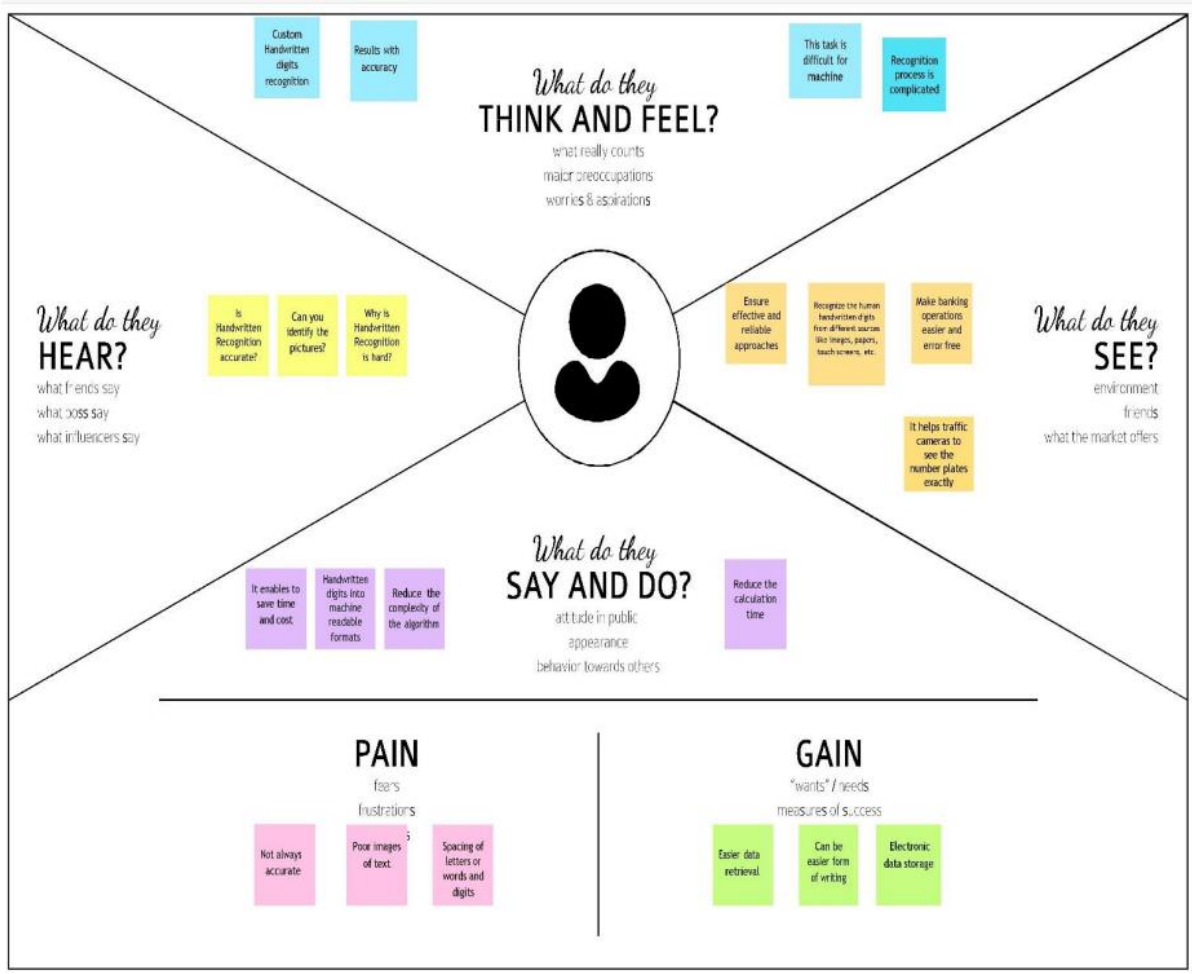
2.3 PROBLEM STATEMENT DEFINITION

The problem to be solved is to recognize the handwritten digits uploaded by the user as an image and to display the equivalent digits as output. The aim is to make the computer identify and automatically understand various styles of handwritten digits to reduce the human efforts. Identifying the handwritten digits manually in the field of bank cheque processing, postal mail sorting and form data entry tasks are tedious and error-prone and hence there is a need for this recognition system to perform these tasks efficiently with improved speed and accuracy. The main goal of this system is to use artificial neural networks to train the user input images and to build a model to detect the digits.

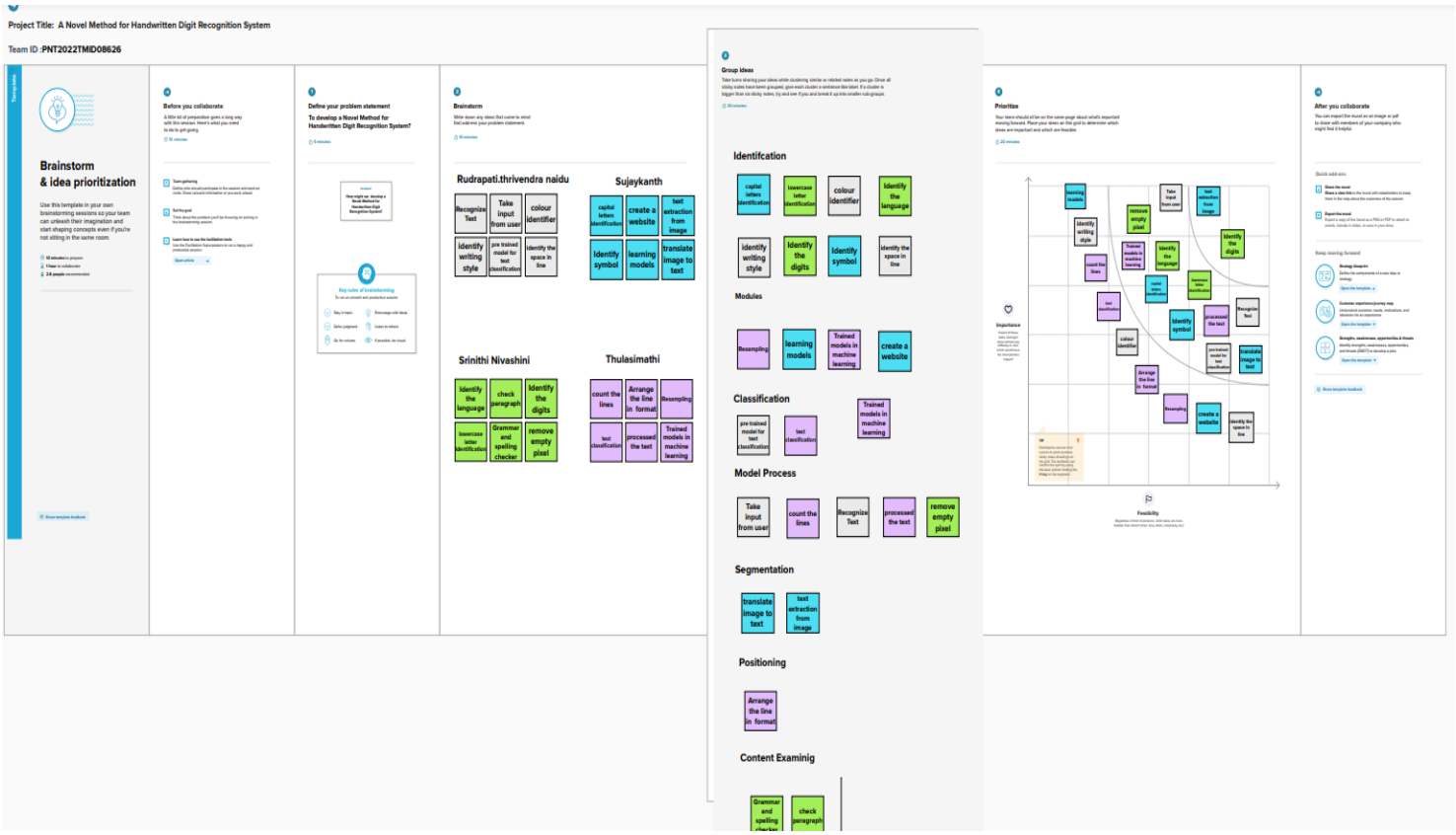
CHAPTER 3

IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS



3.2 IDEATION & BRAINSTORMING



3.3 PROPOSED SOLUTION

S.No.	Parameter	Description
1.	Problem Statement (Problem to besolved)	Handwriting recognition is one of the compelling research works going on becauseevery individual in this world has their own style of writing. It is the capability of the computer to identify and understand handwritten digits or characters automatically. Because of the progress in the field of science and technology,everything is being digitalized to reduce humaneffort. Hence, there comes a need for handwritten digit recognition in many real-time applications.MNIST data set is widely used for this recognition process and it has 70000 handwritten digits. We use Artificial neural networks to train theseimages and build a deep learning model. Web application is created where the user can upload an image of a handwritten digit. this image is analyzed by the model and the detected result is returned on to UI.
2.	Idea / Solution description	HANDWRITTEN digit recognition is the ability ofa computer system to recognize handwritten inputs like digits, characters etc. from a wide variety of sources like emails, papers, images, letting ers. Here comes the use of Deep Learning. In the past decade, deep learning has become the toolfor Image Processing, object detection, handwritten digit, and character recognition etc.A lot of machine learning tools have been developed like sci-kit-learn, scipy-image, etc. andmy brains Keras, Theano, and Tensorflow by google, TFLearn, etc. for Deep Learning. These tools make the applications robust and therefore more accurate.

3.	Novelty / Uniqueness	The first layer of the architecture is the User layer. The user layer will comprise the people who interact with the app and the requiredresults. The next three layers are the front-end architecture of the application. The application will be developed using Bootstrap which is the open-source platform for HTML, CS, S , and JavaScript.The application is deployed in the localhost which is shown on the browser. Through the app, the user will be able to upload pictures of handwritten digits and convert them intahe digitalized forms. The one in between the database and view layer is the business layer which is the logical calculations based on the request from the client side. It also has a service interface.The backend layer consists of two datasets: Training Data and Test Dat a. The MNIST database has been used for thatwhich is already divided into a training set of 60,000 examples and a test
4.	Social Impact / Customer Satisfaction	In addition to reading postal addresses and bank check amounts, it is also useful for readingforms. Furthermore, it's used in fraud detection because it makes it easy to compare two textsand determine which one is a copy. The application has been tested using threemodels: Multi-Layer Perceptron (MLP), and Convolution Neural Network (CNN). As a result, this system fulfill customer's expectations, as it is a novel method for recognizing handwritten digits, ensuring highaccuracy for the model
5.	Business Model (Revenue Model)	For efficient traffic control, this technology can be connected with traffic surveillance camerasto read license plates. Pin-code details can be easily identified and recognized by integrating them with the postal system. Some of the security areas include signatureverification, bank cheque processing, postal address interpretation from envelopes, etc.

3.4 PROBLEM-SOLUTIONTION FIT

Project Title: **A Novel Method For Handwritten Digit Recognition System,**
Team ID: **PNT2022TMID08626**

Project Design Phase - I **: Problem Solution Fit,**

Define CS, fit into CC

1. CUSTOMER SEGMENTS CS	6. CUSTOMER CONSTRAINTS CC	5. AVAILABLE SOLUTIONS AS
• The customers who deal with handwritten digits like Banking sectors, schools, colleges, railways, firm, etc.	• They believe that the alternatives will result in errors and faults and will be inconvenient.	• There are no widely used software's to detect handwriting; instead, they check with other people to affirm what number it is.

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS J&P	9. PROBLEM ROOT CAUSE RC	7. BEHAVIOUR BE
• Handwritten digits can be difficult to understand and interpret at times. It may cause errors when dealing with rough handwriting.	• We face numerous challenges in handwritten number recognition because of different people's jotting styles and the lack of Optic character recognition. This investigation offers an in-depth comparison of various machine literacy and deep literacy.	• Finding the best software for detecting accurate digits in a more efficient manner.

Focus on J&P, tap into BE, understand RC

Identify Strong TR & EM

3. TRIGGERS TR	10. OUR SOLUTION SL	8. CHANNELS of BEHAVIOUR CH
• To obtain the numbers accurately and quickly.	• A solution to this problem is the handwritten digit recognition system, which uses a picture of a digit and recognizes the digit present in the image. Convolutional Neural Network model built with PyTech and applied to the MINIST dataset to recognize handwritten digits.	• Using software that is available on the internet. Obtaining assistance from those nearby in order to recognize the digits written by their customers.
4. EMOTIONS: BEFORE / AFTER		
• Feels frustrated and sad when numbers are not entered		

Identify Strong TR & EM

CHAPTER 4

REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through WebsiteRegistration through Gmail Registration through LinkedIn
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Pre-processing	The model cannot take the image data directly so we need to perform some basic operations and process the data.The CNN model will require one more dimension so we reshape the matrix to shape
FR-4	Create model	Creating CNN model in Python data science project. A CNN model generally consists of convolutional andpooling layers
FR-5	Documentation	Captured in use case
FR-6	Evaluation	The Modified National Institute of Standards and Technology It is a collection of 60,000 tiny square grayscale photographs, each measuring 28 by 28, comprising handwritten single digits between 0 and 9.The MNIST dataset is well-balanced so we can get around 90% accuracy

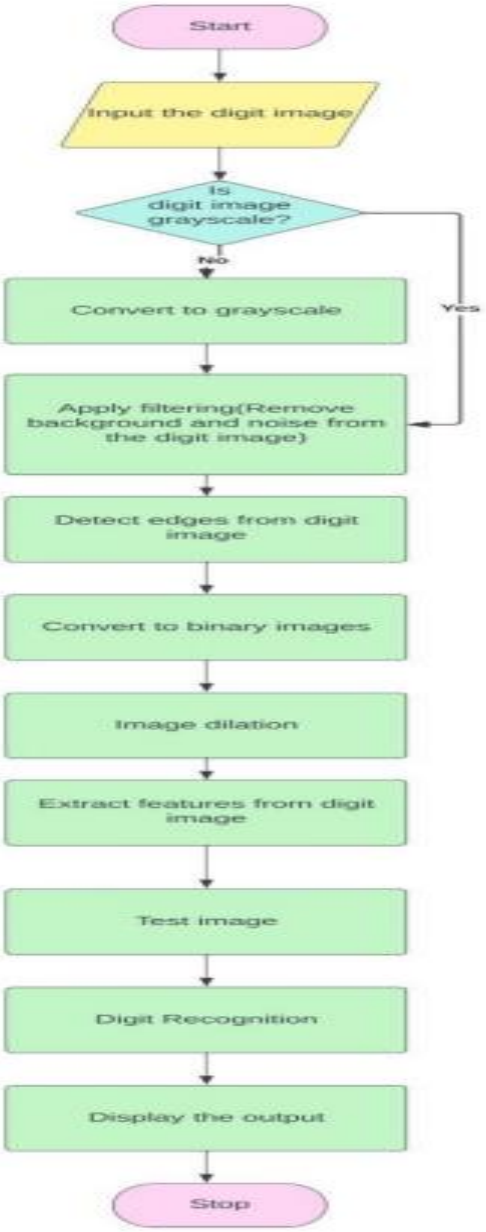
4.2 NON-FUNCTIONAL REQUIREMENTS

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Handwritten character recognition is one of the practically important issues in pattern recognition applications. The applications of digit recognition include postal mail sorting.
NFR-2	**Security**	Most PC efforts to establish safety include information encryption and passwords, OCR plays an important role for digital libraries, allowing the entry of image textual information into computers by digitization, image restoration, and recognition methods
NFR-3	**Reliability**	Specifies the probability of the software performing without failure for a specific number of uses or amount of time
NFR-4	**Performance**	Most standard implementations of neural networks achieve accuracy in correctly classifying the handwritten digits.
NFR-5	**Availability**	The system is not down due to outages or maintenance activities. CNN model can determine and recognize handwritten digits with high accuracy,as it combines the weights of convolution layers during feature extraction with fully connected layers.
NFR-6	**Scalability**	The ability of a solution or system to increase its capacity to serve clients and/or increase processing rates to match demand and speed, robustness, flexible and suitable for text and document formats

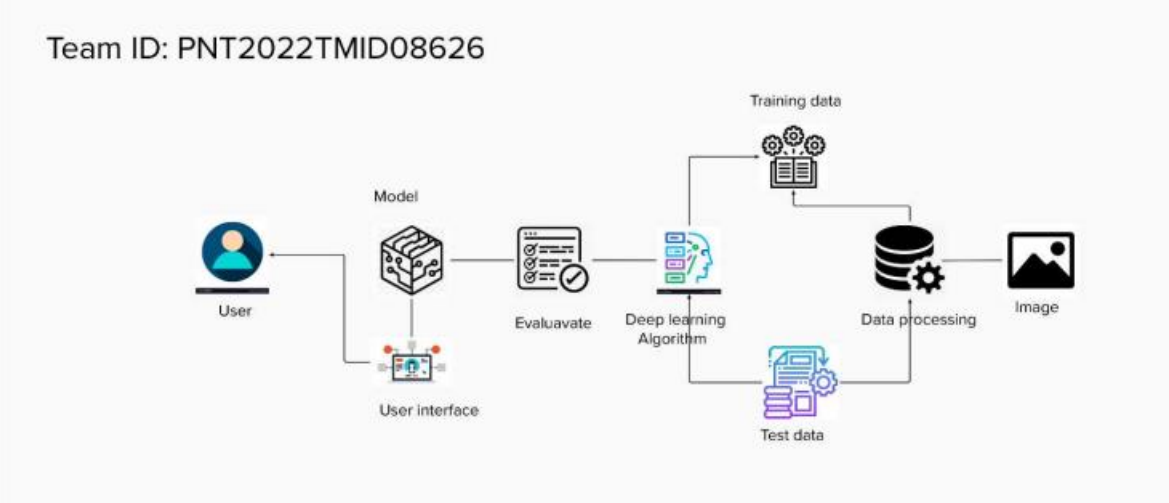
CHAPTER 5

PROJECT DESIGN

5.1 DATA FLOW DIAGRAM



5.2 SOLUTION & TECHNICAL ARCHITECTURE



5.3 USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
User in website	Registration	USN-1	Users can register for the application by entering a username, email, password, and phone number.	Account-specific tasks andactions can be performed	High	Sprint-1
	Login	USN-2	Enter the Username and Password to login intothe application.	Right account credentialsshould be entered	High	Sprint-1
	Dashboard	USN-3	Users can view the website		Medium	Sprint-2

Core development team	Core function	USN-4	Design the application in a way such that itprovides a good user interface.	The interface should beuser-friendly and easy to use	High	Sprint-3
		USN-5	The website should be responsive on all devices irrespective of screen size.	The user experience should betaken into account	Medium	Sprint-3
		USN-6	Dataset collection and processing	Efficient data should becollected	High	Sprint-3
Maintenance team	Maintenance	USN-7	The website should be maintained and the user queries should be fixed as quickly as possible		High	Sprint-4
User in website	Upload the handwritten image	USN-8	The user should upload the handwritten images of the digits which are to be processed.	The input image should be in a properformat.	High	Sprint-4
	View output	USN-9	The system detects the digits and displays the output to the user.		High	Sprint-4

CHAPTER 6

PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNING AND ESTIMATION

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data Collection	USN-1	Users could collect the dataset from various resources with different sets of handwriting.	10	Low	Srinithi Nivashini K Thulasimathi T Sujaykanth S Rudrapati Thrivendra Naidu
Sprint-1	Data Preprocessing	USN-2	Users can load the dataset, handle the missingdata, and scale and split data into train and test.	10	Medium	Srinithi Nivashini K Thulasimathi T Sujaykanth S Rudrapati Thrivendra Naidu
Sprint-2	Building the Model	USN-3	The user will create an application with an ML model that provides high accuracy of recognized handwritten digits.	5	High	Srinithi Nivashini K Thulasimathi T Sujaykanth S Rudrapati Thrivendra Naidu
Sprint-2	Add CNN layers	USN-4	Creating the model and addin the input andoutput layers to	5	High	Srinithi Nivashini K Thulasimathi T Sujaykanth S Rudrapati Thrivendra Naidu

Sprint-2	Compiling the model	USN-5	Configure the learning process of trainingdata and the model which is already designed.	2	Medium	Srinithi Nivashini K Thulasimathi T Sujaykanth S Rudrapati Thrivendra Naidu
Sprint-2	Train & test the model	USN-6	Train the model with the image dataset.	6	Medium	Srinithi Nivashini K Thulasimathi T Sujaykanth S Rudrapati Thrivendra Naidu
Sprint-2	Save the model	USN-7	The model is saved and integrated with theweb application to predict something.	2	Low	Srinithi Nivashini K Thulasimathi T Sujaykanth S Rudrapati Thrivendra Naidu
Sprint-3	Building UI Application	USN-8	Users will upload the handwritten digit imageto the application by clicking an upload button.	5	High	Srinithi Nivashini K Thulasimathi T Sujaykanth S Rudrapati Thrivendra Naidu
Sprint-3	Predict the output	USN-9	Users should know the fundamental usage ofthe application.	5	Low	Srinithi Nivashini K Thulasimathi T Sujaykanth S Rudrapati Thrivendra Naidu
Sprint-3	Evaluating Application	USN-10	The user could see the predicted/recognizeddigits as output in the application.	5	Medium	Srinithi Nivashini K Thulasimathi T Sujaykanth S Rudrapati Thrivendra Naidu

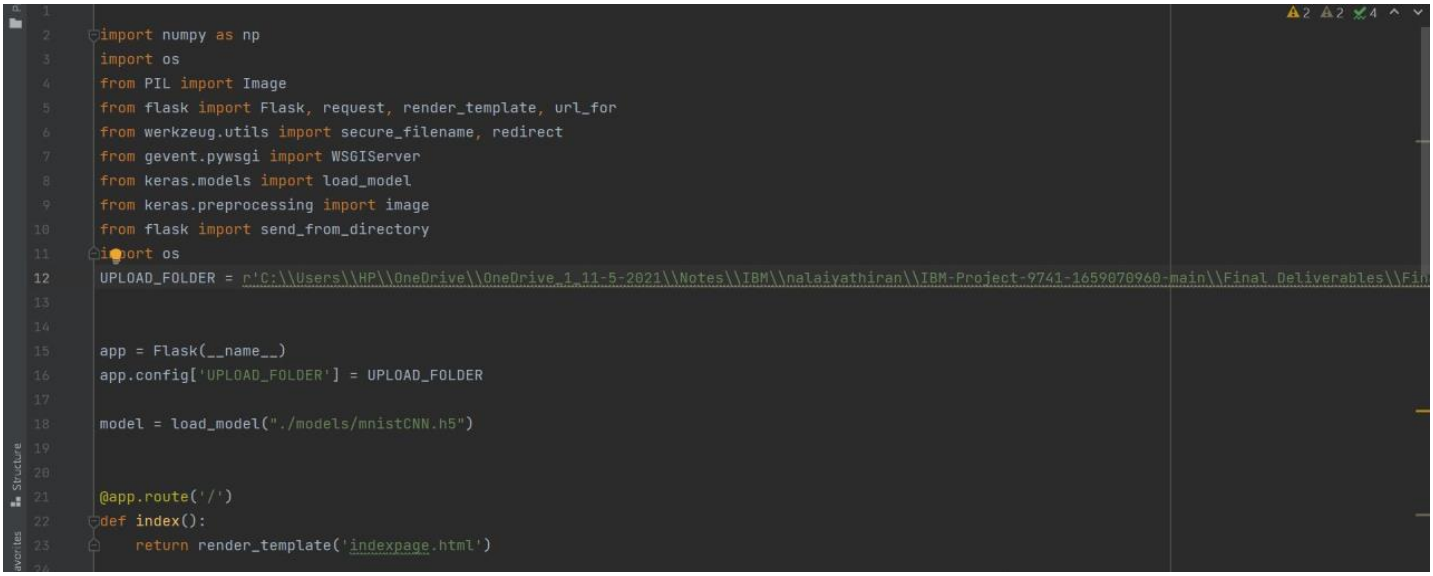
Sprint-4	Train the model on IBM	USN-11	Train the model on IBM and integrate flask/Django with scoring endpoints	10	High	Srinithi Nivashini K Thulasimathi T Sujaykanth S Rudrapati Thrivendra Naidu
Sprint-4	Cloud Deployment	USN-12	Users can access the web application and make use of the product from anywhere.	10	High	Srinithi Nivashini K Thulasimathi T Sujaykanth S Rudrapati Thrivendra Naidu

6.2 SPRINT DELIVERY PLAN

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

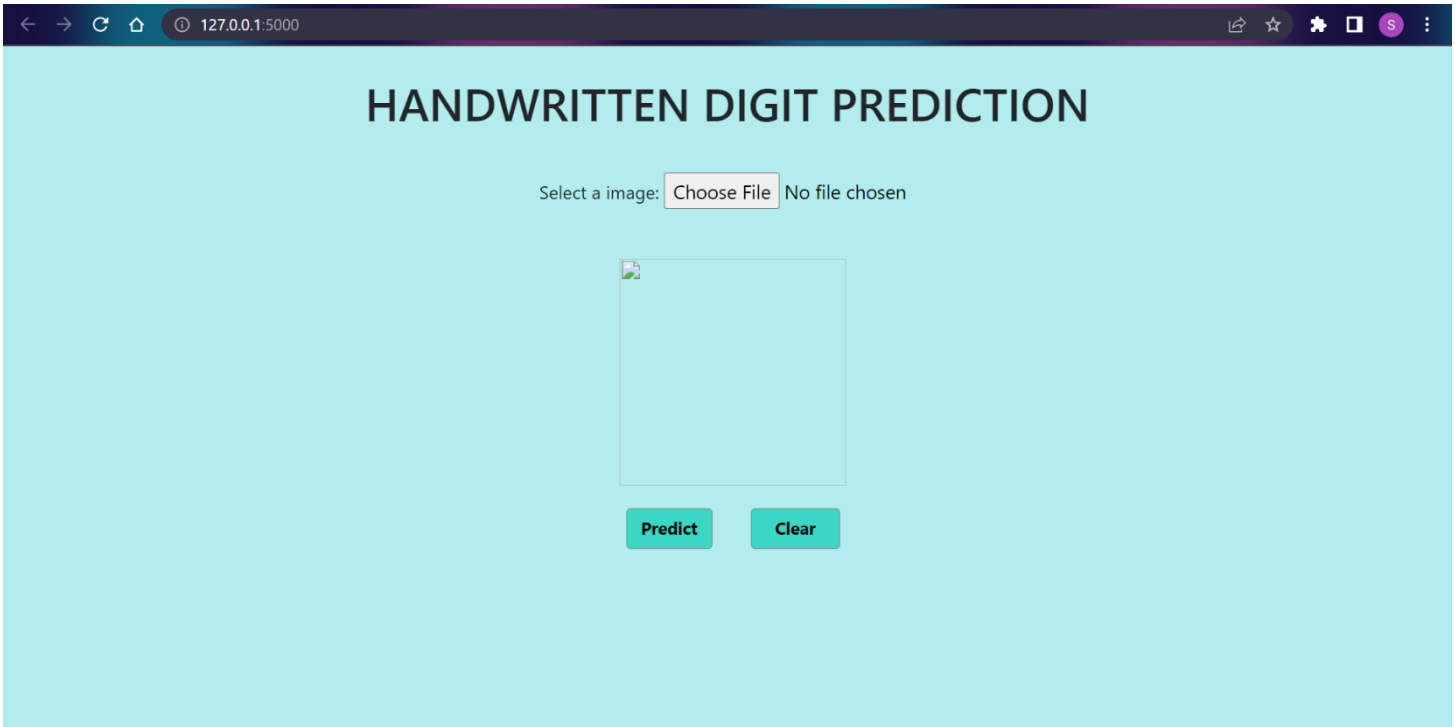
CHAPTER 7

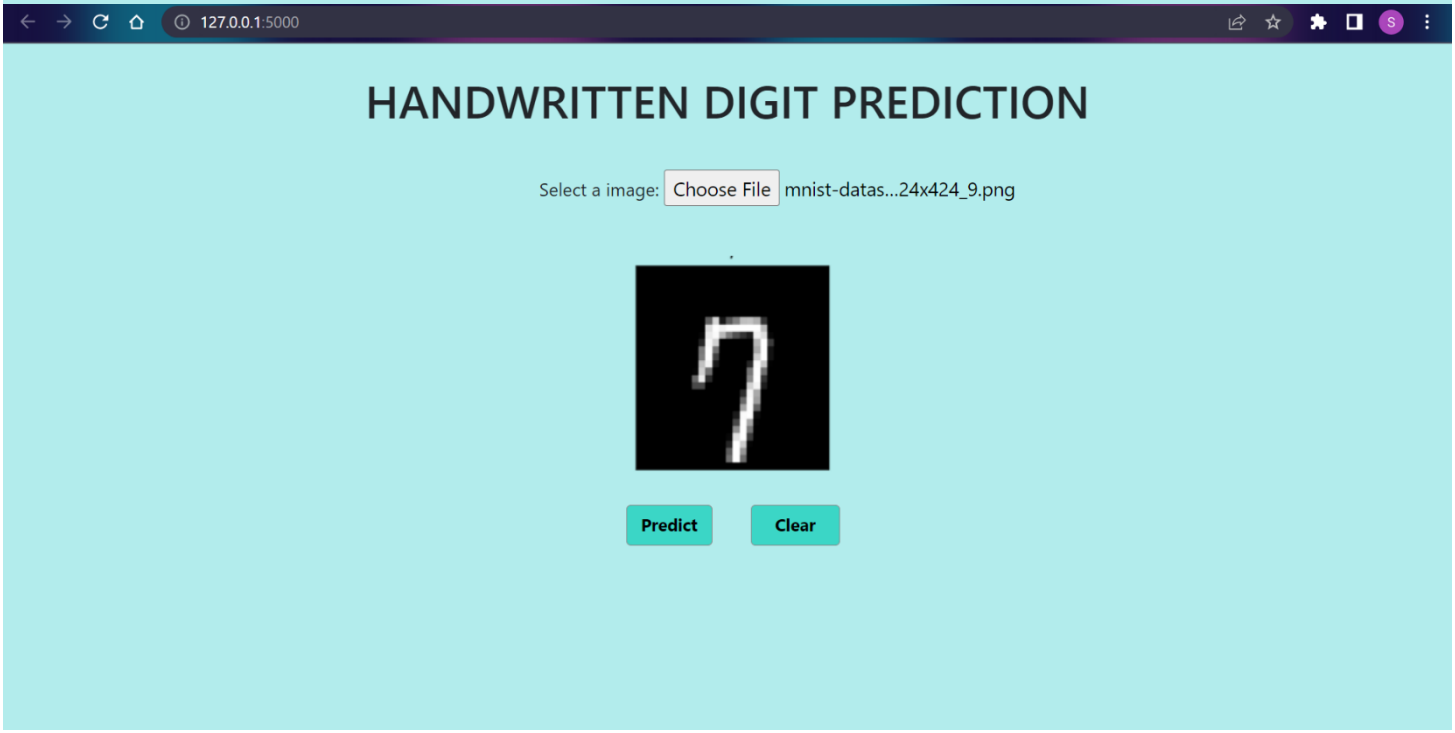
CODING & SOLUTIONING

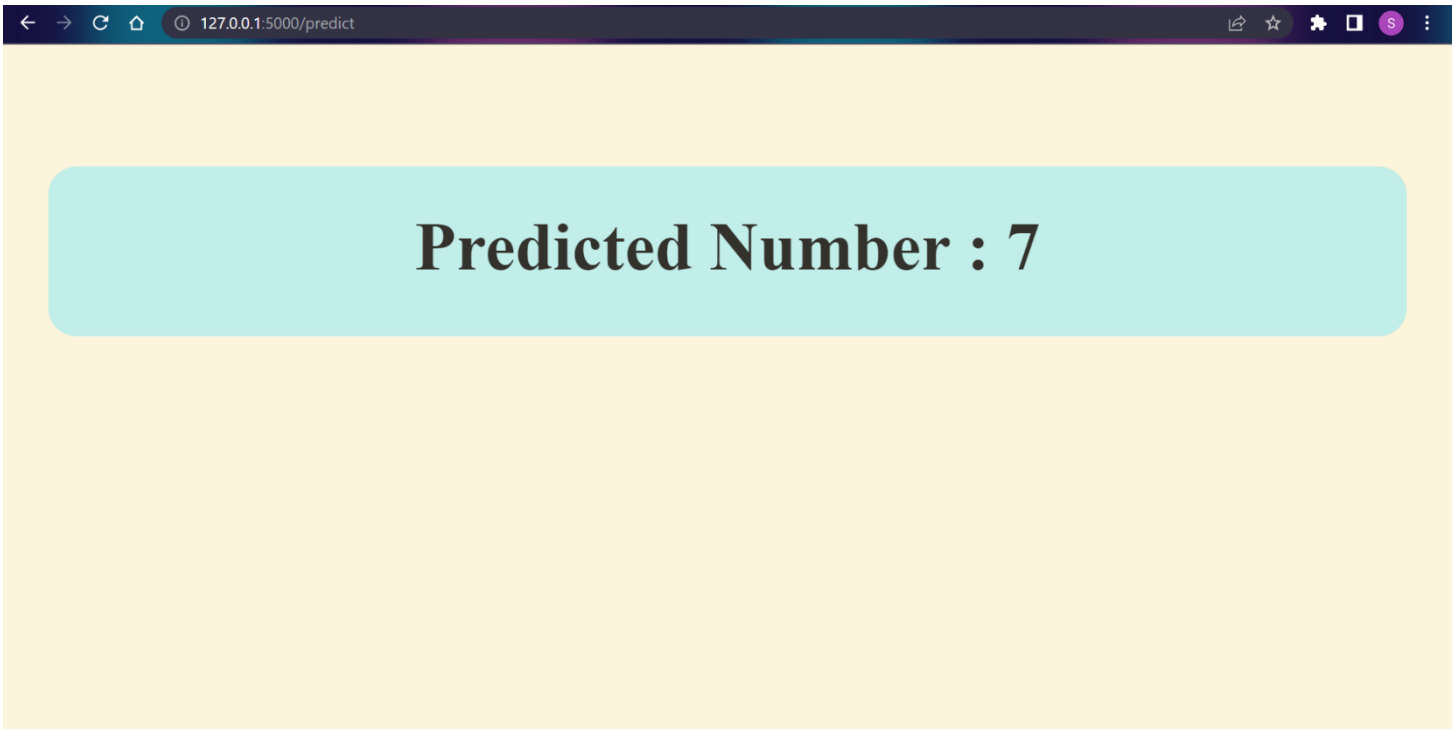


```
import numpy as np
import os
from PIL import Image
from flask import Flask, request, render_template, url_for
from werkzeug.utils import secure_filename, redirect
from gevent.pywsgi import WSGIServer
from keras.models import load_model
from keras.preprocessing import image
from flask import send_from_directory
import os
UPLOAD_FOLDER = r'C:\\Users\\HP\\OneDrive\\OneDrive_1_11-5-2021\\Notes\\IBM\\nalaiyathiran\\IBM-Project-9741-1659070960-main\\Final Deliverables\\Fin


app = Flask(__name__)
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER

model = load_model("./models/mnistCNN.h5")


@app.route('/')
def index():
    return render_template('indexpage.html')
```

```
import numpy as np
import os
from PIL import Image
from flask import Flask, request, render_template, url_for
from werkzeug.utils import secure_filename, redirect
from gevent.pywsgi import WSGIServer
from keras.models import load_model
from keras.preprocessing import image
from flask import send_from_directory
import os
UPLOAD_FOLDER = r'C:\\Users\\HP\\OneDrive\\OneDrive_1_11-5-2021\\Notes\\IBM\\nalaiyathiran\\IBM-Project-9741-1659070960-main\\Final Deliverables\\Fi


app = Flask(__name__)
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER

model = load_model("./models/mnistCNN.h5")


@app.route('/')
def index():
    return render_template('indexpage.html')
```


127.0.0.1:5000
HANDWRITTEN DIGIT PREDICTION
Select a image:
Choose File
No file chosen
Predict
Clear


127.0.0.1:5000
HANDWRITTEN DIGIT PREDICTION
Select a image:
Choose File
mnist-datas...24x424_9.png
Predict
Clear


127.0.0.1:5000/predict
Predicted Number : 7

CHAPTER 8

TESTING

8.1 TEST CASES

Test case ID	Feature Type	Component	Test Scenario	Expected Result	Actual Result	Status
HP_TC_001	Functional	Home Page	Verify the user can see the Homepage when clicking on the link	The home page should be displayed	Working as expected	PASS
HP_TC_002	UI	Home Page	Verify the UI elements on the Homepage	The application should show below UI elements: a.choose file button b.predict button c.clear button	Working as expected	PASS
HP_TC_003	Functional	Home Page	Verify the can to choose the file from the local system and click on predict	Choose file popup screen must be displayed and the user should be able to click on predict button	Working as expected	PASS
HP_TC_004	Functional	Home Page	Verify the user selects the invalid	The application won't allow attaching formats	Working as expected	PASS

			file format	other than ".png, .jiff, .pjp, jpeg, .jpg, .pjpeg"		
Predict_TC_005	Functional	Predict Page	Verify the user can navigate to the predict to and view the predicted result	The user must be navigated to the predicted page and must view the predicted result	Working as expected	PASS

8.2 USER ACCEPTANCE TESTING

8.2.1 DEFECT ANALYSIS

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Total
By Design	1	0	1	0	2
Duplicate	0	0	0	0	0
External	0	0	2	0	2
Fixed	4	1	0	1	6
Not Reproduced	0	0	0	1	1
Skipped	0	0	0	1	1
Won't Fix	1	0	1	0	2
Total	6	1	4	3	14

8.2.2 TEST CASE ANALYSIS

Section	Total Cases	Not Tested	Fail	Pass
Client Application	10	0	3	7
Security	2	0	1	1
Performance	3	0	1	2
Exception Reporting	2	0	0	2

CHAPTER 9

RESULTS

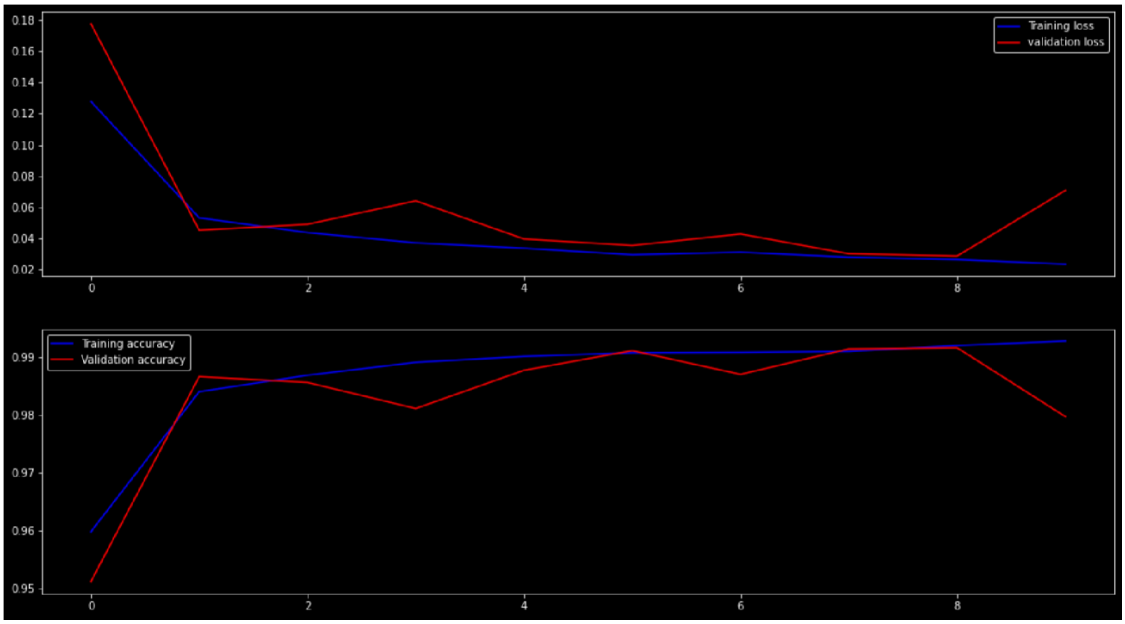
9.1 PERFORMANCE METRICS

MODEL SUMMARY

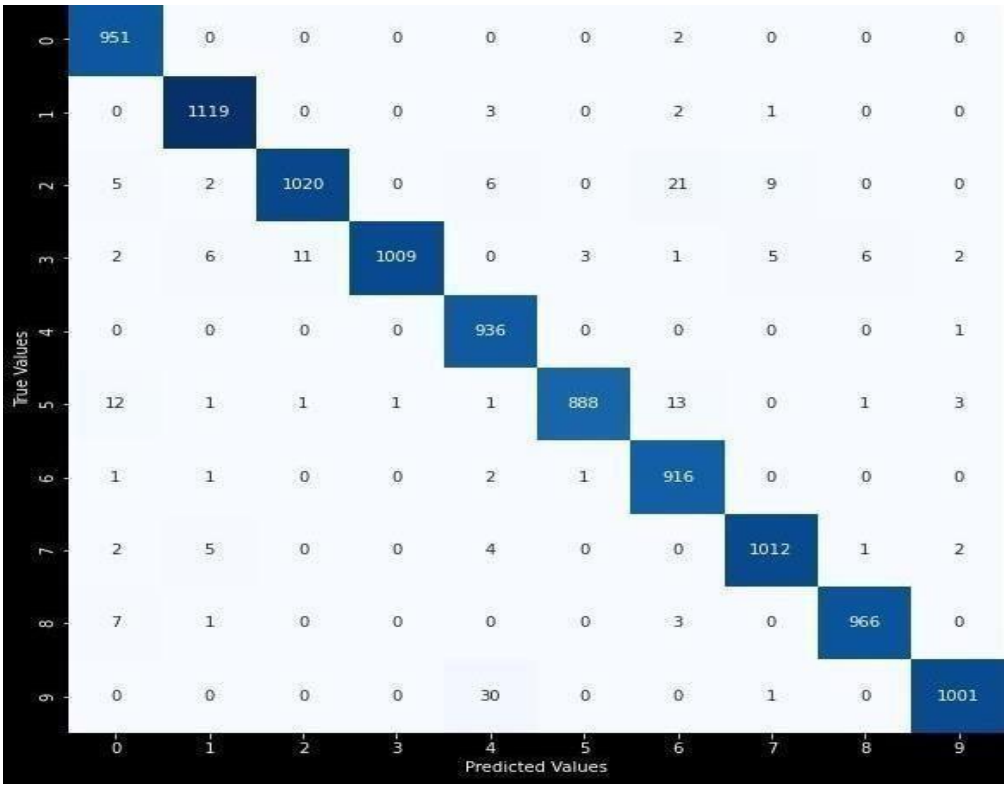
```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 26, 26, 64)        640

 conv2d_1 (Conv2D)           (None, 24, 24, 32)        18464

 flatten (Flatten)           (None, 18432)             0

 dense (Dense)               (None, 10)                184330

=================================================================
Total params: 203,434
Trainable params: 203,434
Non-trainable params: 0
```

ACCURACY

C CONTENT	VALUE
Training Accuracy	99.14%
Training Loss	2.70 %
Validation Accuracy	97.76%
Validation Loss	10 .36%



CONFUSION MATRIX



CLASSIFICATION REPORT

	precision	recall	f1-score	support
0	1.00	0.97	0.98	980
1	0.99	0.99	0.99	1135
2	0.96	0.99	0.97	1032
3	0.97	1.00	0.98	1010
4	1.00	0.95	0.98	982
5	0.96	1.00	0.98	892
6	0.99	0.96	0.97	958
7	0.99	0.98	0.99	1028
8	0.99	0.99	0.99	974
9	0.97	0.99	0.98	1009
accuracy			0.98	10000
macro avg	0.98	0.98	0.98	10000
weighted avg	0.98	0.98	0.98	10000

APPLICATION TEST REPORT

Locust Test Report

During: 11/12/2022, 7:05:40 AM - 11/12/2022, 7:14:47 AM

Target Host: http://127.0.0.1:5000/

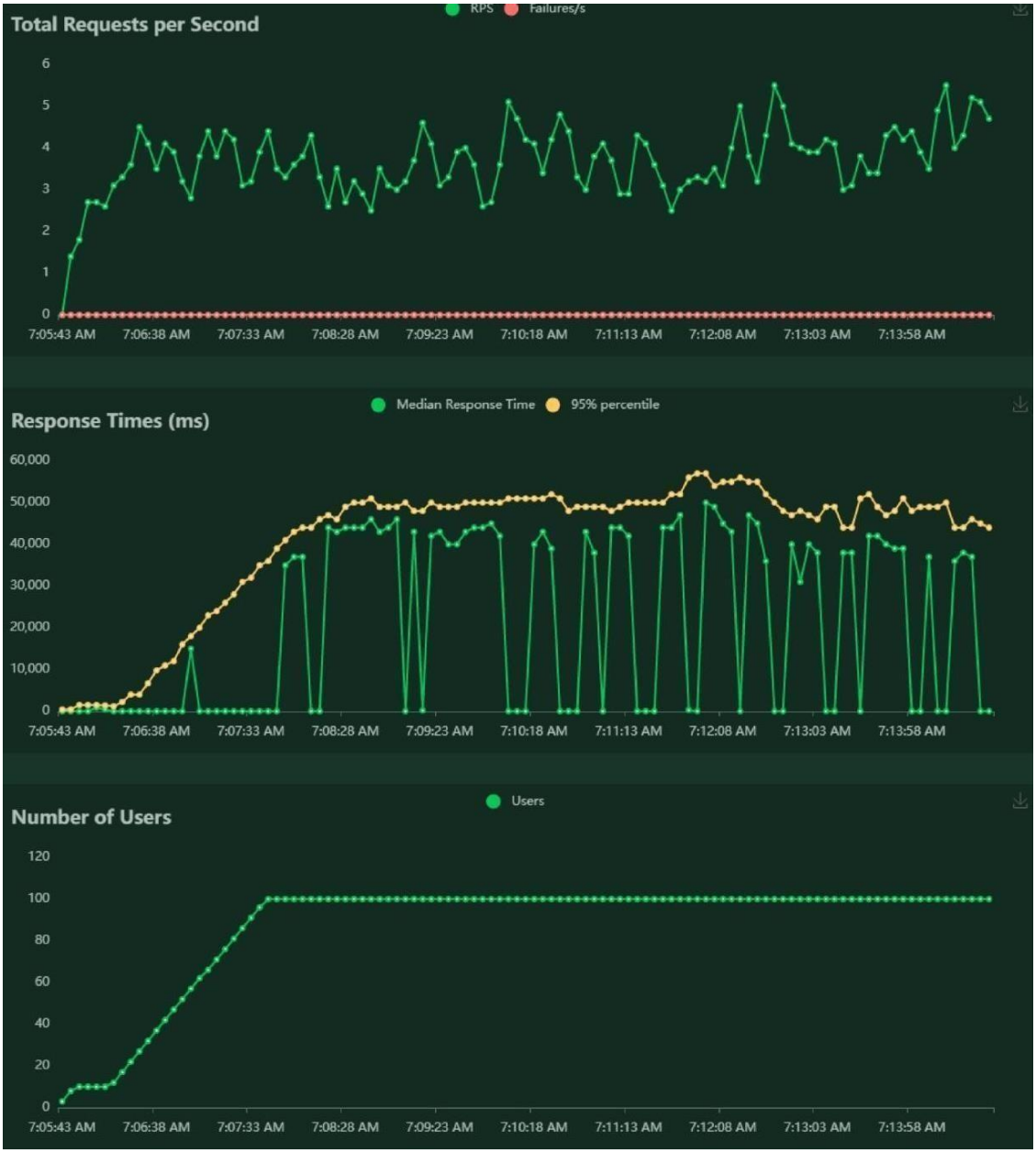
Script: locust.py

Request Statistics

Method	Name	# Requests	# Fails	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	RPS	Failures/s
GET	//	1043	0	13	4	290	1079	1.9	0.0
GET	//predict	1005	0	39648	385	59814	2670	1.8	0.0
	Aggregated	**2048**	**0**	**19462**	**4**	**59814**	**1859**	**3.7**	**0.0**

Response Time Statistics

Method	Name	50%ile (ms)	60%ile (ms)	70%ile (ms)	80%ile (ms)	90%ile (ms)	95%ile (ms)	99%ile (ms)	100%ile (ms)
GET	//	10	11	13	15	19	22	62	290
GET	//predict	44000	46000	47000	48000	50000	52000	55000	60000
	Aggregated	**36**	**36000**	**43000**	**45000**	**48000**	**50000**	**54000**	**60000**


RPS
Failures/s
Total Requests per Second
6
5
4
3
2
1
0
7:05:43 AM
7:06:38 AM
7:07:33 AM
7:08:28 AM
7:09:23 AM
7:10:18 AM
7:11:13 AM
7:12:08 AM
7:13:03 AM
7:13:58 AM
Median Response Time
95% percentile
Response Times (ms)
60,000
50,000
40,000
30,000
20,000
10,000
0
7:05:43 AM
7:06:38 AM
7:07:33 AM
7:08:28 AM
7:09:23 AM
7:10:18 AM
7:11:13 AM
7:12:08 AM
7:13:03 AM
7:13:58 AM
Users
Number of Users
120
100
80
60
40
20
0
7:05:43 AM
7:06:38 AM
7:07:33 AM
7:08:28 AM
7:09:23 AM
7:10:18 AM
7:11:13 AM
7:12:08 AM
7:13:03 AM
7:13:58 AM

CHAPTER 10

ADVANTAGES & DISADVANTAGES

ADVANTAGES

- Reduces manual work
- More accurate than the average human
- Capable of handling a lot of data
- Can be used anywhere from any device

DISADVANTAGES

- Cannot handle complex data
- All the data must be in digital format
- Requires a high-performance server for faster predictions
- Prone to occasional errors

CHAPTER 11

CONCLUSION

This project demonstrated a web application that uses machine learning to recognize handwritten numbers. Flask, HTML, CSS, JavaScript, and a few other technologies were used to create this project. The model predicts the handwritten digit using a CNN network. During testing, the model achieved a 99.61% recognition rate. The proposed project is scalable and can easily handle a huge number of users.Since it is a web application, it is compatible with any device that can run a browser.This project is extremely useful in real-world scenarios such as recognizing the numberplates of vehicles, processing bank cheque amounts, numeric entries in forms filled up by hand (tax forms), and so on. There is so much room for improvement, which can be implemented in subsequent versions.

CHAPTER 12

FUTURE SCOPE

This project is far from complete and there is a lot of room for improvement.
Some of the improvements that can be made to this project are as follows:

- Add support to detect digits in multiple images and save the results
- Add support to detect multiple digits
- Improve the model to detect digits from complex images
- Add support to different languages to help users from all over the world

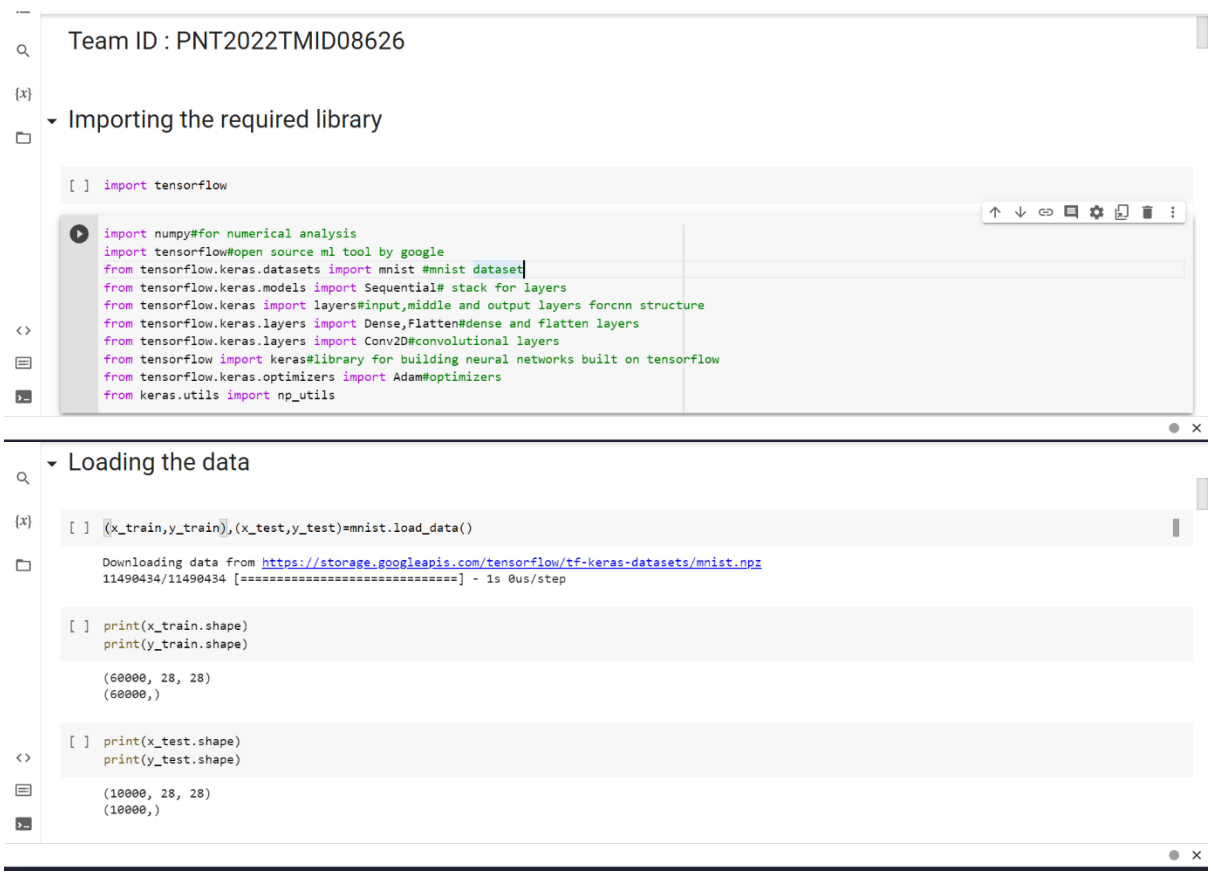
This project has endless potential and can always be enhanced to become better.Implementing this concept in the real world will benefit several industries and reduce the workload on many workers, enhancing overall work efficiency.

CHAPTER 13

APPENDIX

Source code:

Model:



Team ID : PNT2022TMID08626

Importing the required library

```
[ ] import tensorflow
```

```
import numpy#for numerical analysis
import tensorflow#open source ml tool by google
from tensorflow.keras.datasets import mnist #mnist dataset
from tensorflow.keras.models import Sequential# stack for layers
from tensorflow.keras import layers#input,middle and output layers forcnn structure
from tensorflow.keras.layers import Dense,Flatten#dense and flatten layers
from tensorflow.keras.layers import Conv2D#convolutional layers
from tensorflow import keras#library for building neural networks built on tensorflow
from tensorflow.keras.optimizers import Adam#optimizers
from keras.utils import np_utils
```

Loading the data

```
[ ] (x_train,y_train),(x_test,y_test)=mnist.load_data()
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz
11490434/11490434 [==============================] - 1s 0us/step
```

```
[ ] print(x_train.shape)
    print(y_train.shape)
```

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

```
[ ] print(x_test.shape)
    print(y_test.shape)
```

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

Analyzing the data

```
[ ] x_train[5]
```

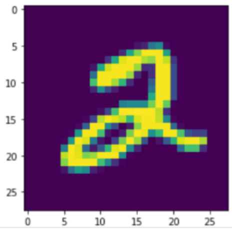
```
array([[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,  13,  25, 100, 122,   7,   0,   0,   0,   0,   0,   0,
          0,   0],
```

```
[ ] y_train[5]
```

```
2
```

```
import matplotlib.pyplot as plt
plt.imshow(x_train[5])
```

```
<matplotlib.image.AxesImage at 0x7fa18f25ecd0>
```



Reshaping the data

```
[ ] x_train=x_train.reshape(60000,28,28,1).astype('float32')
    x_test=x_test.reshape(10000,28,28,1).astype('float32')
```

```
print ("Shape of X_train: {}".format(x_train.shape))
print ("Shape of y_train: {}".format(y_train.shape))
print ("Shape of X_test: {}".format(x_test.shape))
print ("Shape of y_test: {}".format(y_test.shape))
```

```
Shape of X_train: (60000, 28, 28, 1)
Shape of y_train: (60000,)
Shape of X_test: (10000, 28, 28, 1)
Shape of y_test: (10000,)
```

Applying one Hotencoding

convert numerical values to classes where 0 to 9 are 10 seperate classes if value is 5 class 5 is 1 else 0

```
[ ] no_of_classes=10
    y_train=np_utils.to_categorical(y_train,no_of_classes)
    y_test=np_utils.to_categorical(y_test,no_of_classes)
```

```
[ ] y_test[3]
```

```
array([1., 0., 0., 0., 0., 0., 0., 0., 0., 0.], dtype=float32)
```

```
[ ] from keras.layers import Dense, Flatten, MaxPooling2D, Dropout
```

Adding CNN layer

```
[ ] model = Sequential()

    model.add(Conv2D(64,(3,3),input_shape=(28,28,1),activation='relu'))
    model.add(Conv2D(32,(3,3),activation='relu'))

    model.add(Flatten())
    model.add(Dense(no_of_classes,activation='softmax'))
```

Compile the model

```
[ ] model.compile(loss='categorical_crossentropy',optimizer='Adam',metrics=['accuracy'])
```

Train the model

```
[ ] model.fit(x_train,y_train,validation_data=(x_test,y_test),epochs=5,batch_size=32)
```

```
Epoch 1/5
1875/1875 [==============================] - 193s 103ms/step - loss: 0.2600 - accuracy: 0.9511 - val_loss: 0.0938 - val_accuracy: 0.9728
Epoch 2/5
1875/1875 [==============================] - 187s 100ms/step - loss: 0.0719 - accuracy: 0.9785 - val_loss: 0.0950 - val_accuracy: 0.9707
Epoch 3/5
1875/1875 [==============================] - 187s 100ms/step - loss: 0.0477 - accuracy: 0.9847 - val_loss: 0.0900 - val_accuracy: 0.9781
Epoch 4/5
1875/1875 [==============================] - 185s 99ms/step - loss: 0.0372 - accuracy: 0.9890 - val_loss: 0.1050 - val_accuracy: 0.9747
Epoch 5/5
1875/1875 [==============================] - 186s 99ms/step - loss: 0.0304 - accuracy: 0.9907 - val_loss: 0.0842 - val_accuracy: 0.9820
<keras.callbacks.History at 0x7fa18aab1490>
```


```
[ ] 1875/1875 [==============================] - 193s 103ms/step - loss: 0.2600 - accuracy: 0.9511 - val_loss: 0.0938 - val_accuracy: 0.9728
    Epoch 2/5
    1875/1875 [==============================] - 187s 100ms/step - loss: 0.0719 - accuracy: 0.9785 - val_loss: 0.0950 - val_accuracy: 0.9707
    Epoch 3/5
    1875/1875 [==============================] - 187s 100ms/step - loss: 0.0477 - accuracy: 0.9847 - val_loss: 0.0900 - val_accuracy: 0.9781
    Epoch 4/5
    1875/1875 [==============================] - 185s 99ms/step - loss: 0.0372 - accuracy: 0.9890 - val_loss: 0.1050 - val_accuracy: 0.9747
    Epoch 5/5
    1875/1875 [==============================] - 186s 99ms/step - loss: 0.0304 - accuracy: 0.9907 - val_loss: 0.0842 - val_accuracy: 0.9820
    <keras.callbacks.History at 0x7fa18aab1490>
```


Observing the metrices

```
[ ] metrics=model.evaluate(x_test,y_test,verbose=0)
    print("metrics-score=>test loss & accuracy")
    print(metrics)
```

```
metrics-score=>test loss & accuracy
[0.08420193940401077, 0.9819999933242798]
```

Test the model

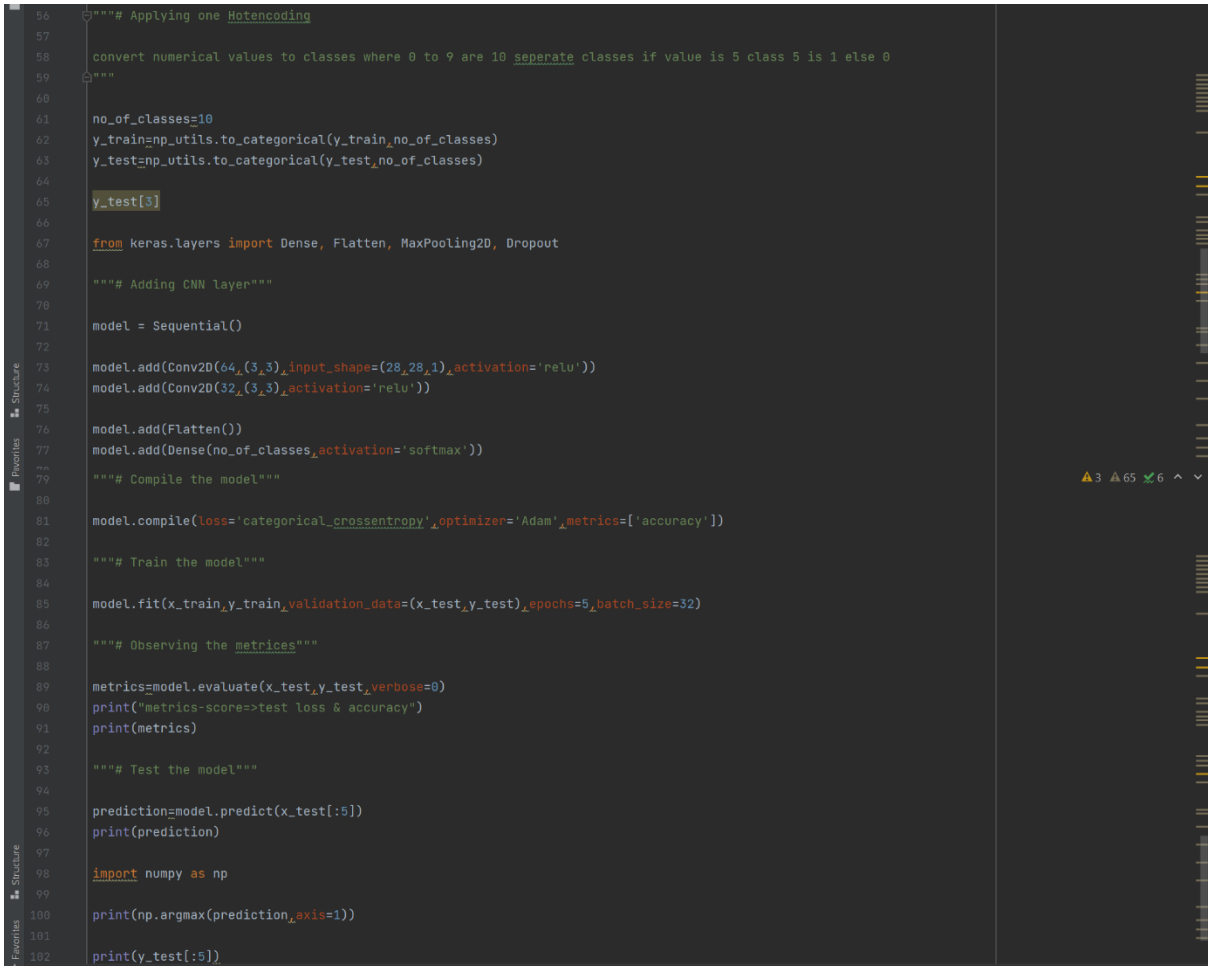
```
[ ] prediction=model.predict(x_test[:5])
    print(prediction)
```

```
1/1 [==============================] - 0s 84ms/step
[[4.0351332e-13 4.6145593e-15 7.9426864e-12 3.6988692e-07 1.2474837e-18
  5.0288202e-13 9.9698470e-20 9.9999964e-01 1.4521950e-09 5.5775353e-09]
 [2.0407164e-11 2.9936387e-10 1.0000000e+00 1.5163411e-14 6.8775140e-22
  1.0346708e-17 3.6543427e-10 1.4415800e-19 4.9613288e-14 4.5817813e-21]
 [3.4889942e-12 9.9998963e-01 6.7495657e-06 5.2625865e-15 1.2803760e-07
  1.8978109e-10 5.0176570e-09 2.6155569e-10 3.4661741e-06 5.3021099e-12]
 [1.0000000e+00 4.0936742e-15 2.1811340e-10 1.5707134e-16 4.3095729e-14
  2.9080857e-12 7.7849167e-09 1.4526050e-10 7.3599512e-12 9.0574536e-12]
 [1.0947845e-14 1.1511424e-11 4.0602462e-18 9.3403068e-18 1.0000000e+00
  7.6617902e-18 1.4756050e-16 2.2480922e-14 2.9495194e-11 1.7560243e-09]]
```

```
[ ] import numpy as np
```

```
[ ] import numpy as np

[ ] print(np.argmax(prediction,axis=1))

    [7 2 1 0 4]

[ ] print(y_test[:5])

    [[0. 0. 0. 0. 0. 0. 0. 1. 0. 0.]
     [0. 0. 1. 0. 0. 0. 0. 0. 0. 0.]
     [0. 1. 0. 0. 0. 0. 0. 0. 0. 0.]
     [1. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
     [0. 0. 0. 0. 1. 0. 0. 0. 0. 0.]]
```

```
import tensorflow

import numpy#for numerical analysis
import tensorflow#open source ml tool by google
from tensorflow.keras.datasets import mnist #mnist dataset
from tensorflow.keras.models import Sequential# stack for layers
from tensorflow.keras import layers#input,middle and output layers forcnn structure
from tensorflow.keras.layers import Dense,Flatten#dense and flatten layers
from tensorflow.keras.layers import Conv2D#convolutional layers
from tensorflow import keras#library for building neural networks built on tensorflow
from tensorflow.keras.optimizers import Adam#optimizers
from keras.utils import np_utils

"""# Loading the data"""

(x_train,y_train),(x_test,y_test)=mnist.load_data()

print(x_train.shape)
print(y_train.shape)

print(x_test.shape)
print(y_test.shape)

"""# Analyzing the data"""

x_train[5]

y_train[5]

import matplotlib.pyplot as plt
plt.imshow(x_train[5])

"""# Reshaping the data"""

x_train=x_train.reshape(60000,28,28,1).astype('float32')
x_test=x_test.reshape(10000,28,28,1).astype('float32')

print("Shape of X_train: {}".format(x_train.shape))
print("Shape of y_train: {}".format(y_train.shape))
print("Shape of X_test: {}".format(x_test.shape))
print("Shape of y_test: {}".format(y_test.shape))

"""# Applying one Hotencoding

convert numerical values to classes where 0 to 9 are 10 seperate classes if value is 5 class 5 is 1 else 0
"""
```



```
"""# Applying one Hotencoding

convert numerical values to classes where 0 to 9 are 10 seperate classes if value is 5 class 5 is 1 else 0
"""

no_of_classes=10
y_train=np_utils.to_categorical(y_train,no_of_classes)
y_test=np_utils.to_categorical(y_test,no_of_classes)

y_test[3]

from keras.layers import Dense, Flatten, MaxPooling2D, Dropout

"""# Adding CNN layer"""

model = Sequential()

model.add(Conv2D(64,(3,3),input_shape=(28,28,1),activation='relu'))
model.add(Conv2D(32,(3,3),activation='relu'))

model.add(Flatten())
model.add(Dense(no_of_classes,activation='softmax'))
"""# Compile the model"""

model.compile(loss='categorical_crossentropy',optimizer='Adam',metrics=['accuracy'])

"""# Train the model"""

model.fit(x_train,y_train,validation_data=(x_test,y_test),epochs=5,batch_size=32)

"""# Observing the metrices"""

metrics=model.evaluate(x_test,y_test,verbose=0)
print("metrics-score=>test loss & accuracy")
print(metrics)

"""# Test the model"""

prediction=model.predict(x_test[:5])
print(prediction)

import numpy as np

print(np.argmax(prediction,axis=1))

print(y_test[:5])
```

HTML Code:

Index Page:

```
<html>


<head>
  <title>Digit Recognition WebApp</title>


  <meta name="viewport" content="width=device-width">
  <!-- GoogleFont -->
  <link href="https://fonts.googleapis.com/css2?family=Prompt:wght@600&display=swap" rel="stylesheet">
  <link href="https://fonts.googleapis.com/css2?family=Varela+Round&display=swap" rel="stylesheet">
```

```
        <link
href="https://fonts.googleapis.com/css2?family=Source+Code+Pro:wght@500&display=swap"
rel="stylesheet">

        <link
href="https://fonts.googleapis.com/css?family=Calistoga|Josefin+Sans:400,700|Pacifico&display=sw
ap" rel="stylesheet">

        <!-- bootstrap -->

        <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" integrity="sha384-
ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">

        <link rel="stylesheet" type= "text/css" href="{{ url_for('static',filename='css/styleNT.css')
}}">

        <!-- fontawesome -->

        <script src="https://kit.fontawesome.com/b3aed9cb07.js"
crossorigin="anonymous"></script>


        <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-
q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
crossorigin="anonymous"></script>

        <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"
integrity="sha384-
UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1"
crossorigin="anonymous"></script>

        <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"
integrity="sha384-
JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM"
crossorigin="anonymous"></script>

        <script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@latest"></script>


    </head>


    <script>
      function preview() {
        frame.src=URL.createObjectURL(event.target.files[0]);
    }


        $(document).ready(function() {
```

```
            $('#clear_button').on('click', function() {
                $('#image').val('');
                $('#frame').attr('src',"");
              });
            });

    </script>

    <body>


      <section id="content">
          <center><h1>HANDWRITTEN DIGIT PREDICTION</h1><center>
          <div class="leftside">
          <form action="/predict" method="POST" enctype="multipart/form-data">
          <label>Select a image:</label>

          <input id="image" type="file" name="image" accept="image/png, image/jpeg" onchange="preview()"><br><br>
            <img id="frame" src="" width="200px" height="200px"/>
            <div class="buttons_div">
              <button type="submit" class="btn btn-light" id="predict_button">Predict</button>
              <button type="button" class="btn btn-light" id="clear_button">  Clear  </button>
            </div>
          </form>
          </div>
      </section>

    </body>

    </html>
```

Predict Page:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Prediction</title>
    <link rel="stylesheet" type= "text/CSS" href= "{{
url_for('static',filename='css/predictNT.css') }}">
</head>

<body>
    <div id="rectangle">
        <h1 id="ans">Predicted Number : {{num}}</h1>
    </div>
</body>
</html>
```

CSS Code:

Index Page:

```
                    body{
  background:#b2ecec;
}
#clear_button{
  margin-left: 15px;
  font-weight: bold;
  color: black;
}
```

```
#confidence{
  font-family: 'Josefin Sans', sans-serif;
  margin-top: 7.5%;
}

#content{
  margin: 0 auto;
  padding: 2% 15%;
  padding-bottom: 0;

}

.welcome{
   text-align: center;
   position: relative;
   color: honeydew;
   background-color:#0E6655;
   padding-top: 1%;
   padding-bottom: 1%;
   font-weight: bold;
   font-family: 'Prompt', sans-serif;
}

#team_id{
   text-align: right;
   font-size: 20px;
   padding-right: 3%;
}

#predict_button{
  margin-right: 15px;
```

```
  color: black;
  font-weight: bold;
}

#prediction_heading{
  font-family: 'Josefin Sans', sans-serif;
  margin-top: 7.5%;
}

#result{
  font-size: 5rem;
}

#title{
  padding: 1.5% 15%;
  margin: 0 auto;
  text-align: center;
}

.btn {
    font-size: 15px;
    -WebKit-appearance: none;
    background: #3bd6c6;
    border: 1px solid #888;
    margin-top: 20px;
    margin-bottom: 20px;
}

.buttons_div{
  margin-bottom: 30px;
  margin-right: 80px;
}
```

```
.heading{
  font-family: 'Varela Round', sans-serif;
  font-weight: 700;
  font-size: 2rem;
  display: inline;
}

.leftside{
  text-align: center;
  margin: 0 auto;
  margin-top: 2%;
  /* padding-left: 10%; */
}

#frame{
  margin-right: 10%;
}

.predicted_answer{
  text-align: center;
  margin: 0 auto;
  padding: 3% 5%;
  padding-top: 0;
  /* padding-left: 10%; */
}

p{
  font-family: 'Source Code Pro', monospace,sans-serif;
  margin-top: 1%;
}
```

```
@media (min-width: 720px) {
  .leftside{
    padding-left: 10%;
  }
}
input[type="file"] {
  font-size: 17px;
  color: black;
   margin-top: 20px;
    margin-bottom: 20px;
   display: inline-block;
}
```

Predict Page:

```
  body{
       background:  #FDF4DC;
       background-repeat: no-repeat;
       background-size: cover;
      }

      #rectangle{
       width:1200px;
       height:150px;
       background-color: #b3ecec;
       opacity: 0.8;
       border-radius: 25px;
       position: absolute;
       top:30%;
       left:50%;
       transform:translate(-50%,-50%);
      }
```

```
 #ans{
text-align: center;
font-size: 60px;
margin: 0 auto;
padding: 3% 5%;
padding-top: 3%;
color: black;
 }
```

GITHUB

https://github.com/IBM-EPBL/IBM-Project-9741-1659070960

PROJECT DEMO

https://drive.google.com/file/d/1jlEU0Z2OMkYRIUzfYBSwtNz212gRXbHw/view?usp=share_link