A Novel Method for Handwritten Digit Recognition System

Project Based Learning Report

Submitted by

Team ID : PNT2022TMID11386

Team Members

Santhosh Kumar KR	910619104071
Subramanian VR	590420104301
Vasantha Kumar K	910619104097
Vishveshwaran M	910619104102

INTRODUCTION

1.1 Project overview

Handwriting recognition is one of the compelling research works going on because every 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 human effort. 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 these images and build a deep learning model.

1.2 Purpose

The purpose of the project is to accurately predict the handwritten digits by every individual words. Because, in this world everyone has their own style of writing. Find their hand written digits properly by the system.

Handwritten digits are not perfect and can be made with many different flavors. The handwritten digit recognition is the solution to this problem which uses the image of a digit and recognizes the digit present in the image.

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

LITERATURE SURVEY

2.1 Existing Problem

Handwriting recognition tends to have problems when it comes to accuracy. People can struggle to read others’ handwriting. How, then, is a computer going to do it?

The issue is that there’s a wide range of handwriting – good and bad. This makes it tricky for programmers to provide enough examples of how every character might look. Plus, sometimes, characters look very similar, making it hard for a computer to recognise accurately.

Joined-up handwriting is another challenge for computers. When your letters all connect, it makes it hard for computers to recognise individual characters. Consider, for instance, an ‘r’ and an ‘n’. Joined up, these letters could be mistaken for an ‘m’.

In the case of handwriting recognition from photos, there are also awkward angles to consider. The angle the photo is taken could obscure the character, making it harder for the computer to identify.

2.2 References

1.)”HANDWRITTEN DIGITS RECOGNITION WITH DECISION TREE CLASSIFICATION”-Tsehay Admassu Assegie -2019

2.)”HANDWRITTEN DIGIT RECOGNITION USING VARIOUS MACHINE LEARNING ALGORITHMS AND MODELS “- Pranit s.patil ,Bhupinder kaur -2020

3.)A Recognition System for Handwritten Digits Using CNN -Siddiga,chakrapani -2021

4.)“Handwritten Digit String Recognition using Convolutional Neural Network”-honhjianzhan -2019

5.)”Multi-Digit Handwritten Sindhi Numerals Recognition using SOM Neural Network“-A.chandio ,H.jalbani-2019

2.3 Problem Statement

To identify the letters of the sentence written by the user in their devices and to convert the handwritten text into digital format.

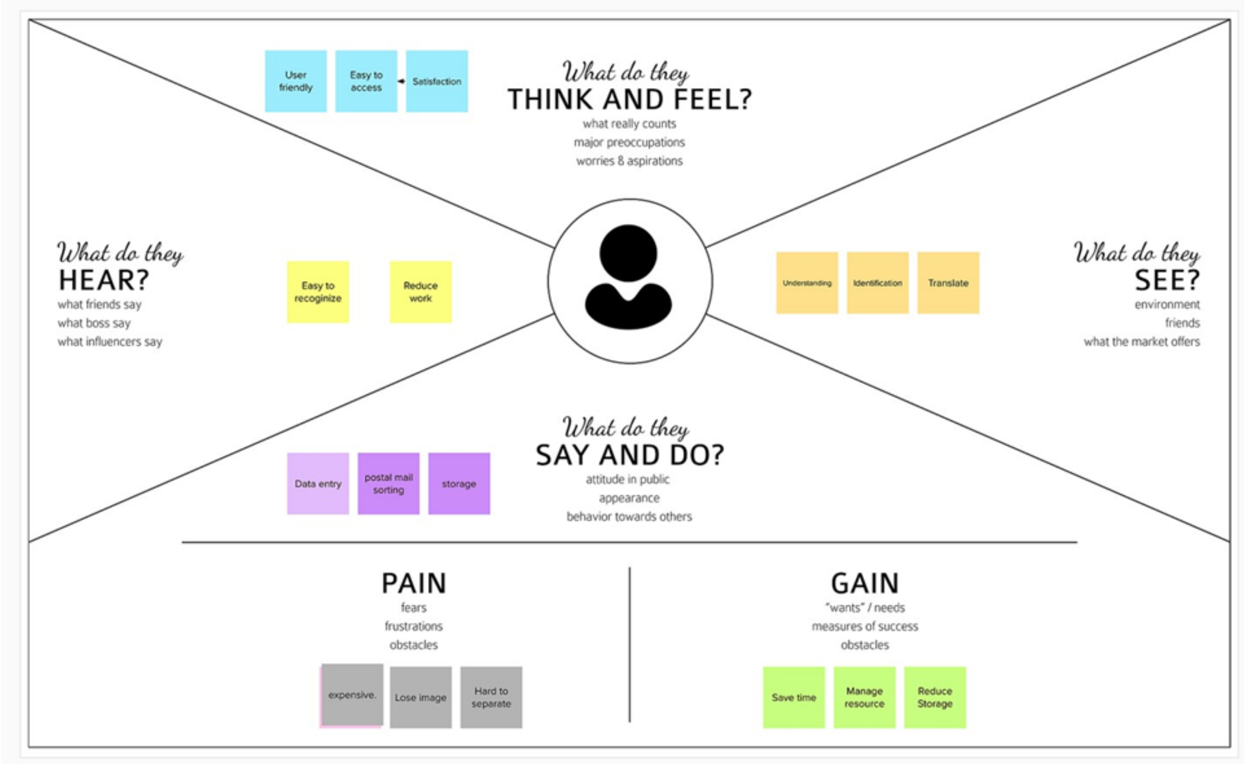
What doesthis problem focus on?	The generative models can perform recognition drivensegmentation. Themethod involves a relatively small numberof parameter and hence training is relatively easyand fast.
When does this occur?	This matter occurs when written by hand digitsare not necessarily alwaysof typically the same size, thickness, orientation and validated to margins since they differcoming from writingof personal to individual.
Why do we need this?	Automating these tasks removes theneed for human effort which is error prone in performing these kind of tedious worksand improves speed as well as efficiency.
How to do this?	Unlike many other recognition schemes, it does not rely on some form of pre-normalization of inputimages, but can handle arbitrary scalings, translations and a limited degree of image rotation.

Where it is used?	The digit recognization system isused in postal mail sorting, bank checkprocessing, form dataentry.

IDEATION AND PROPOSED SOLUTION

3.1 Empathy Map

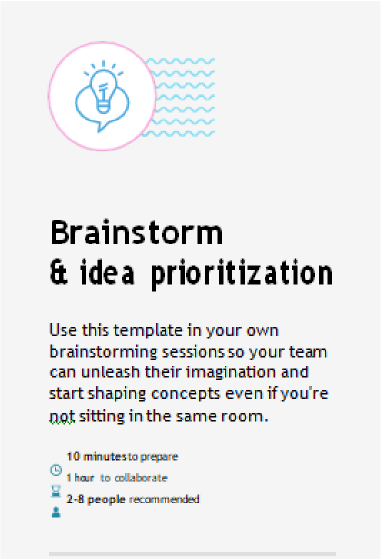
Empathy map helps is simple and easy to digest visually that captures knowledge about a user's behaviour and attitudes.



3.2 Ideation and Brain Storming

Brainstorming provides a free and open environment that encourages everyone within a team to participate in creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.

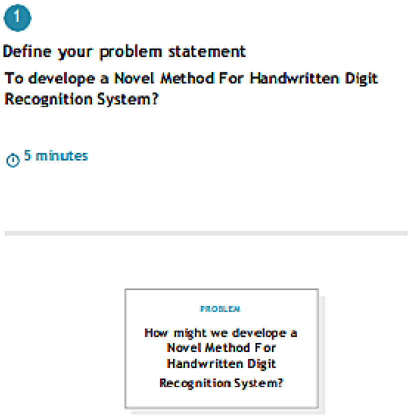
3.2.1 Defining the Problem



Brainstorm
& idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

10 minutes to prepare
1 hour to collaborate
2-8 people recommended



1

Define your problem statement

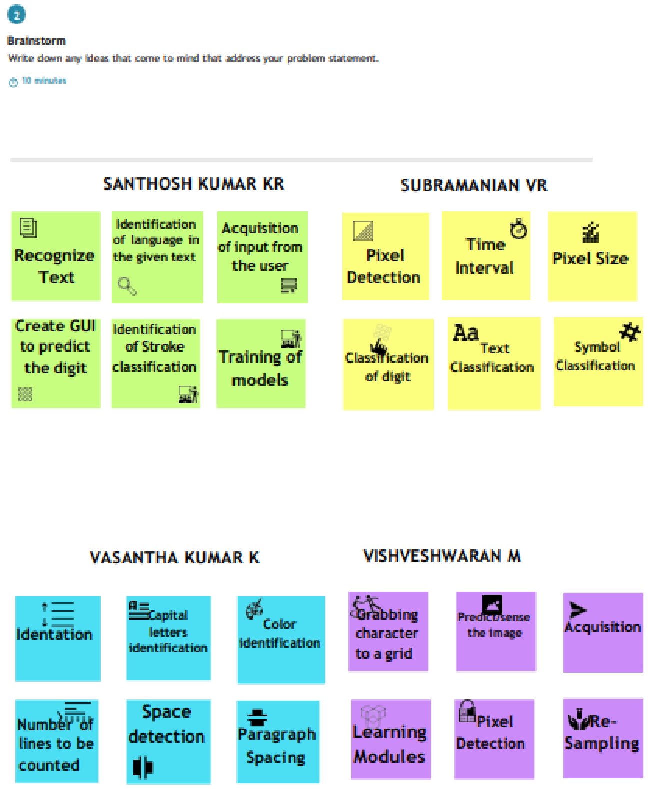
To develope a Novel Method For Handwritten Digit Recognition System?

5 minutes

PROBLEM

How might we develope a Novel Method For Handwritten Digit Recognition System?

3.2.2 Brainstorming



2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

SANTHOSH KUMAR KR

- Recognize Text
- Identification of language in the given text
- Acquisition of input from the user
- Create GUI to predict the digit
- Identification of Stroke classification
- Training of models

SUBRAMANIAN VR

- Pixel Detection
- Time Interval
- Pixel Size
- Classification of digit
- Aa Text Classification
- Symbol Classification

VASANTHA KUMAR K

- Identation
- Capital letters identification
- Color identification
- Number of lines to be counted
- Space detection
- Paragraph Spacing

VISHVESHWARAN M

- Grabbing character to a grid
- Predict/sense the image
- Acquisition
- Learning Modules
- Pixel Detection
- Re-Sampling

3.3 Proposed solution

S.NO	Parameter	Description
1.	Problem Statement (Problemto be solved)	**Statement**-The handwritten digit recognition isthe capability of computer applications to recognize the human handwritten digits. **Description**:It is a hard task for the machinebecause handwritten digits arenot perfect andcan be made with many different shapes and sizes.

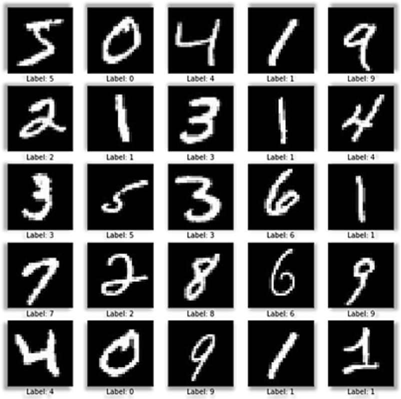
2.	Idea / Solution description	1. It is the capability of a computer to fete themortal handwritten integers from different sources like images, papers, touch defences. 2. It allows user to translate all those signature and notesinto electronic words in a text document format and this dataonly requires farless physical space thanthe storage of the physical copies.
3.	Novelty / Uniqueness	Accurately recognize the digits ratherthanrecognizing all the characters like OCR.
4.	Social Impact /CustomerSatisfaction	1. Artificial Intelligence developed the app calledHandwritten digit Recognizer. 2. It converts the written word into digital approximations and utilizes complex algorithmsto identify characters before churning out a digital approximation.
5.	Business Model (RevenueModel)	1. This system can be integrated with traffic surveillance cameras to recognize the vehicle’snumber platesfor effective traffic management.

3.4 Problem Solution Fit

MNIST (“Modified National Institute of Standards and Technology”) is considered an unofficial computer vision “hello-world” dataset. This is a collection of thousands of handwritten pictures used to train classification models using Machine Learning techniques

.

Handwriting recognition is one of the compelling research works going on because every 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 human effort. 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 these images and build a deep learning model. Web application is created where the user can upload an image of a handwritten digit. this image is analysed by the model and the detected result is returned on to UI.



The MNIST Handwritten Digit Recognition Dataset contains 60,000 training and 10,000 testing labelled handwritten digit pictures.

REQUIREMENT ANALYSIS

4.1 Functional Requirements

FR No.	Sub Requirement (Story / Sub-Task)
FR-1	Image Data: Handwritten digit recognition refers to a computer's capacity to identify humanhandwritten digits froma variety of sources, such as photographs, documents, touch screens, etc., and categorise them into tenestablished classifications (0-9). In the realmof deep learning, this has beenthe subject of countless studies.
FR-2	Website: Web hosting makes the code, graphics, and other items that make up a website accessible online. A server hostsevery website you'veever visited. The typeof hosting determines how much space is allotted to a website on a server. Shared, dedicated, VPS, and reseller hosting are the four basic varieties.
FR-3	Digit Classifier Model: To train a convolutional network to predict the digit from an image, use the MNIST database of handwritten digits. get the training and validationdata first.
FR-4	Cloud: The cloud offers a range of IT services, including virtual storage, networking, servers, databases, and applications. In plain English, cloud computing is described asa virtual platform that enables unlimited storage and access to your data over the internet.
FR-5	Modified National Institute of Standards and Technology dataset: The abbreviation MNIST stands for the MNIST dataset. It is a collection of 60,000tiny square grayscale photographs, each measuring 28 by 28, comprising handwritten single digitsbetween 0 and 9.

4.2 Non-Functional Requirements

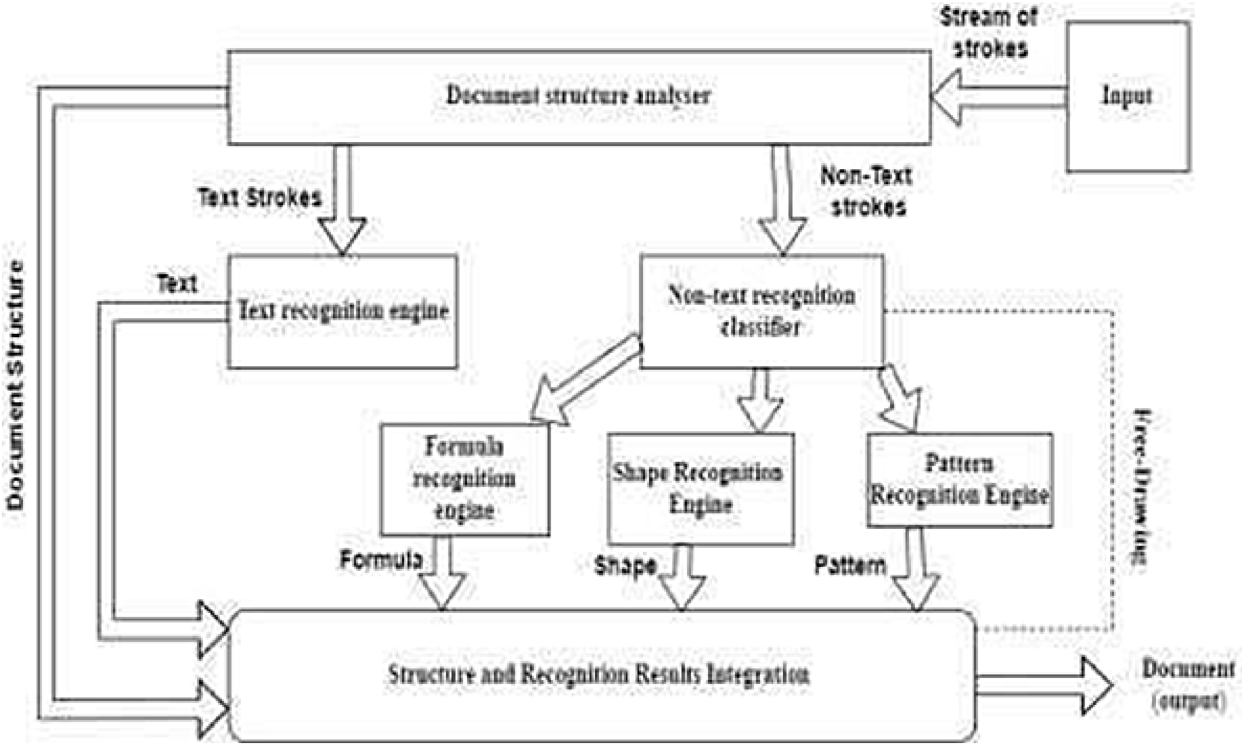
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	One of the very significant problems in pattern recognition applications is the recognition of handwritten characters. Applications for digit recognition includefilling out forms,processingbank checks, and sorting mail.
NFR-2	**Security**	1) The system generates a thorough descriptionof theinstantiation parameters, whichmight

PROJECT DESIGN

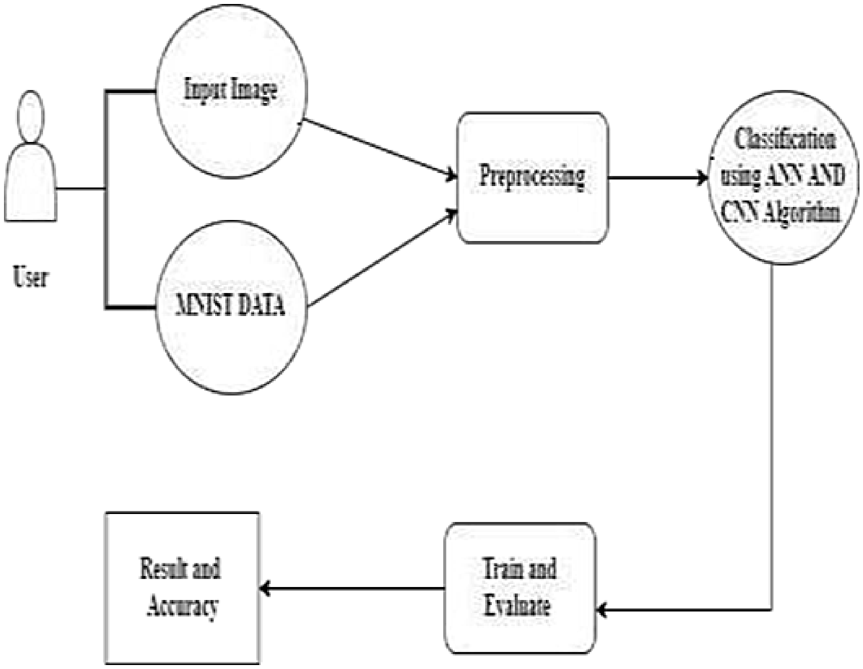
5.1 Data Flow Diagram

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

Level 0:

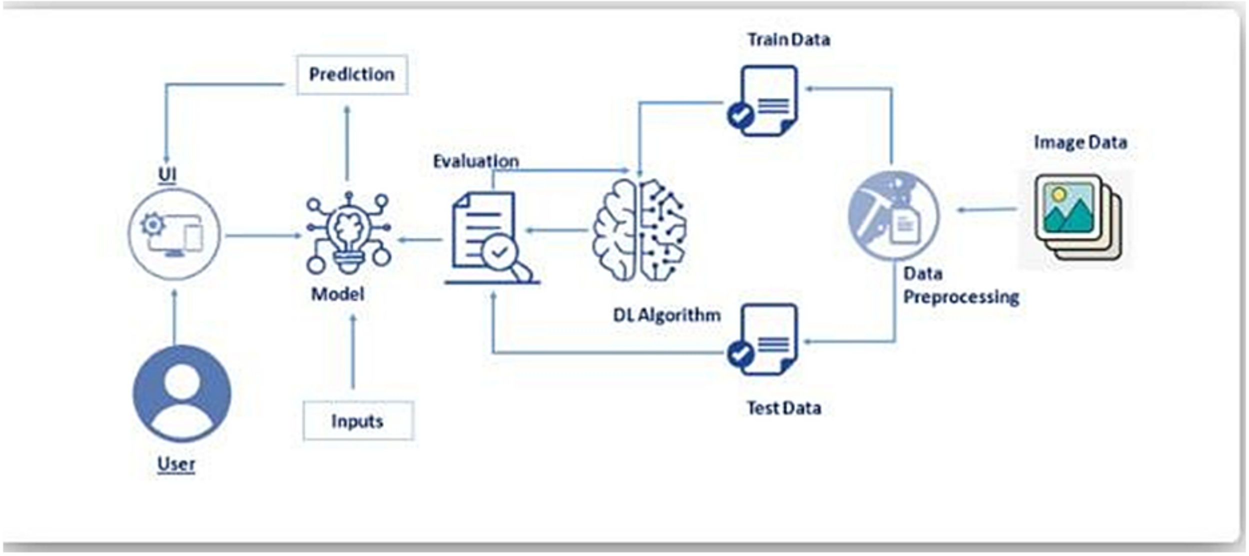


Simplified Diagram:

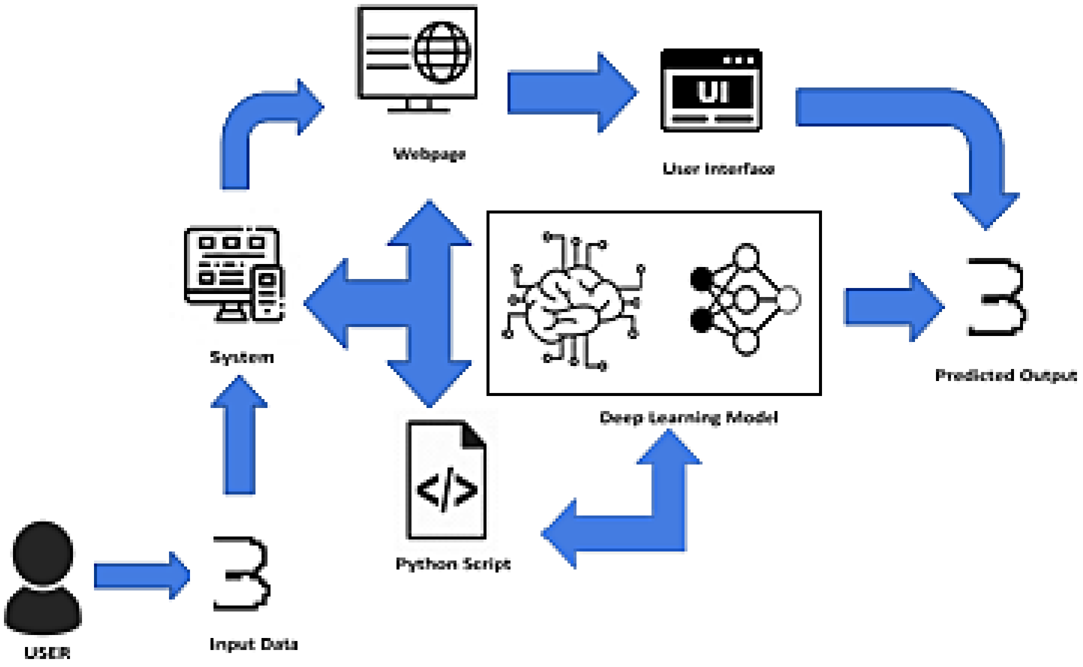


5.2 Solution and Technical Architecture

5.2.1 Solution Architecture



5.2.2 Technical Architecture



5.3 User Stories

User Type	Functional Requirement(Epic)	User Story Number	User Story/ Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Home	USN-1	As a user, I can viewthe guide andawareness to usethis application.	I can view the awarenessto use this application and its limitations.	Low	Sprint-1
		USN-2	As a user, I'm allowed to view theguidedvideo to use the interface of this application.	I can gain knowledge touse this application by a	Low	Sprint-1

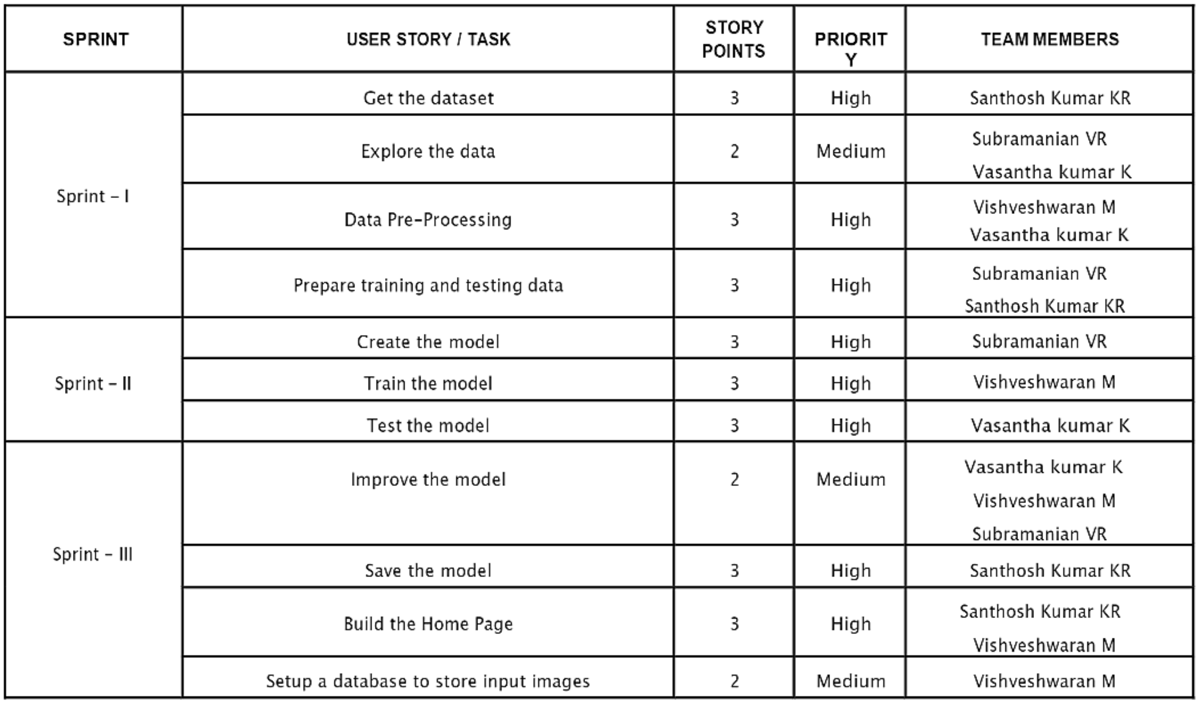
				practical method.		
		USN-3	As a user, I can readthe instructions to use thisapplication.	I can read instructionsal so to use it in a user-friendly method.	Low	Sprint-2
	Recognize	USN-4	As a user, In this prediction page I get tochoose the image.	I can choose theimage from our local system and predict the output.	High	Sprint-2
	Predict	USN-6	As a user, I’m Allowed to upload and choose the image to be uploaded	I can uploadand choosethe image from the system storage and alsoin any virtualstorage.	Medium	Sprint-3
		USN-7	As a user, I will trainand test the input togetthe maximum accuracy of output.	I can able to train and test the application untilit gets maximum accuracy of the result.	High	Sprint-4
		USN-8	As a user,I can access the MNISTdata set	I can access the MNISTdata set to produce the accurate result.	Medium	Sprint-3

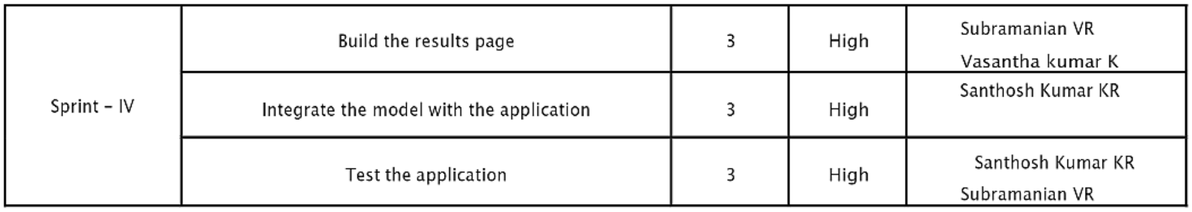
Customer (Web user)	Home	USN-9	As a user, I can viewthe guide to use theweb app.	I can view the awarenessof this application and its limitations.	Low	Sprint-1
User Type	**Functional Requirement(Epic)**	**User Story Number**	**User Story/ Task**	**Acceptance criteria**	**Priority**	**Release**
Customer (Mobile user)	Home	USN-1	As a user, I can viewthe guide andawareness to usethis application.	I can view the awarenessto use this application and its limitations.	Low	Sprint-1
		USN-2	As a user, I’m allowed to view theguidedvideo to use the interface of this application.	I can gain knowledge touse this application by a practical method.	Low	Sprint-1
		USN-3	As a user, I can readthe instructions to use thisapplication.	I can read instructions also to use it in a user-friendly method.	Low	Sprint-2
	Recognize	USN-10	As a user, I can usethe web application virtually anywhere.	I can use the application portably anywhere.	High	Sprint-1
		USN-11	As it is an open source,can use it costfreely.	I can use it without any payment to be paid for itto	Medium	Sprint-2

				access.		
		USN-12	As it is a web application, it is installationfree	I canuse it without the installation of the application or any software.	Medium	Sprint-4
	Predict	USN-13	As a user, I’m Allowed to upload and choose the image to be uploaded	I can upload and choosethe image from the system storage and alsoin any virtualstorage.	Medium	Sprint-3

PROJECT PLANNING AND SCHEDULING

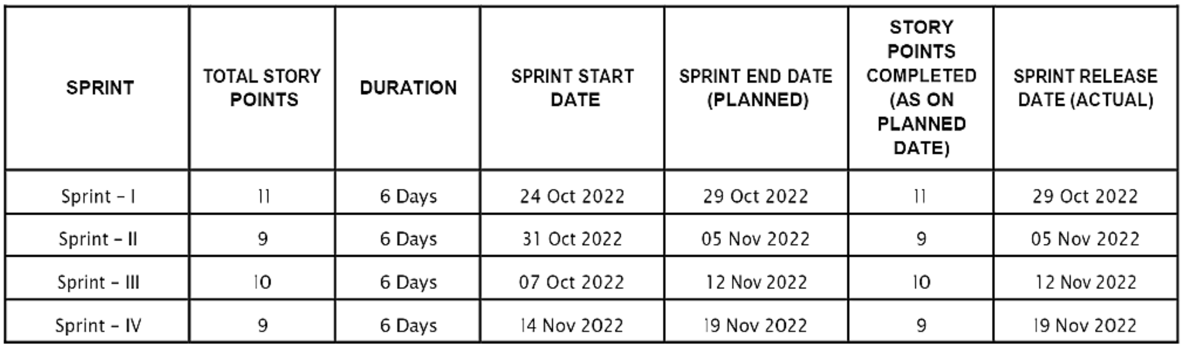
6.1 Sprint Planning and Estimation





SPRINT	USER STORY / TASK	STORY POINTS	PRIORITY	TEAM MEMBERS
Sprint – I	Get the dataset	3	High	Santhosh Kumar KR
	Explore the data	2	Medium	Subramanian VR Vasantha kumar K
	Data Pre-Processing	3	High	Vishveshwaran M Vasantha kumar K
	Prepare training and testing data	3	High	Subramanian VR Santhosh Kumar KR
Sprint – II	Create the model	3	High	Subramanian VR
	Train the model	3	High	Vishveshwaran M
	Test the model	3	High	Vasantha kumar K
Sprint – III	Improve the model	2	Medium	Vasantha kumar K Vishveshwaran M Subramanian VR
	Save the model	3	High	Santhosh Kumar KR
	Build the Home Page	3	High	Santhosh Kumar KR Vishveshwaran M
	Setup a database to store input images	2	Medium	Vishveshwaran M
Sprint – IV	Build the results page	3	High	Subramanian VR Vasantha kumar K
	Integrate the model with the application	3	High	Santhosh Kumar KR
	Test the application	3	High	Santhosh Kumar KR Subramanian VR

6.2 Sprint Delivery Schedule

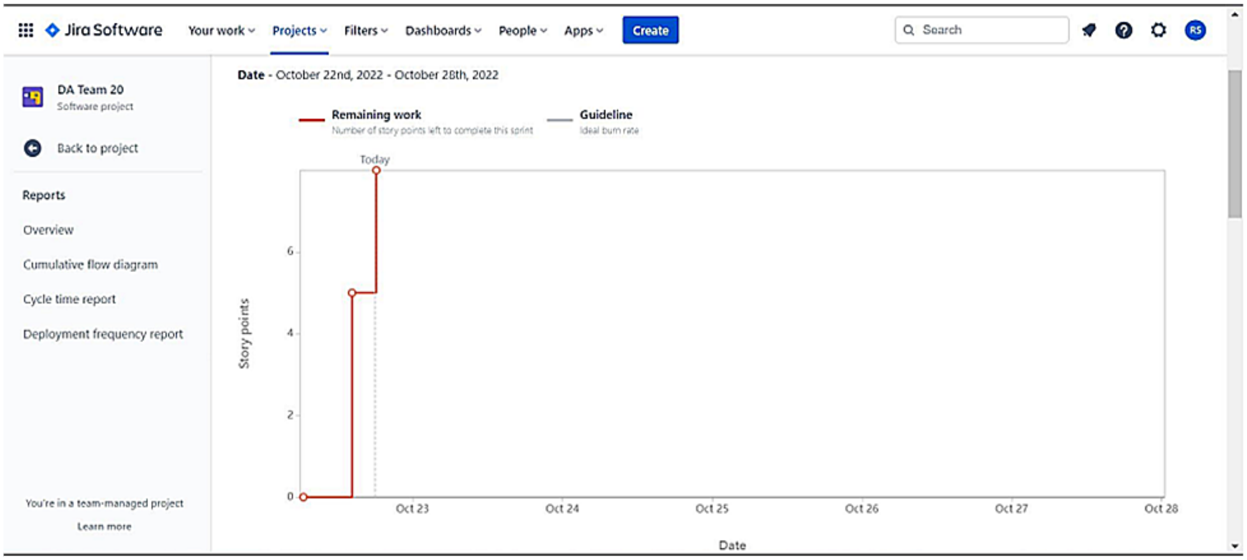


SPRINT	TOTAL STORY POINTS	DURATION	SPRINT START DATE	SPRINT END DATE (PLANNED)	STORY POINTS COMPLETED (AS ON PLANNED DATE)	SPRINT RELEASE DATE (ACTUAL)
Sprint – I	11	6 Days	24 Oct 2022	29 Oct 2022	11	29 Oct 2022
Sprint – II	9	6 Days	31 Oct 2022	05 Nov 2022	9	05 Nov 2022
Sprint – III	10	6 Days	07 Oct 2022	12 Nov 2022	10	12 Nov 2022
Sprint – IV	9	6 Days	14 Nov 2022	19 Nov 2022	9	19 Nov 2022

6.3 Reports from Jira

6.3.1 Burndown Charts:

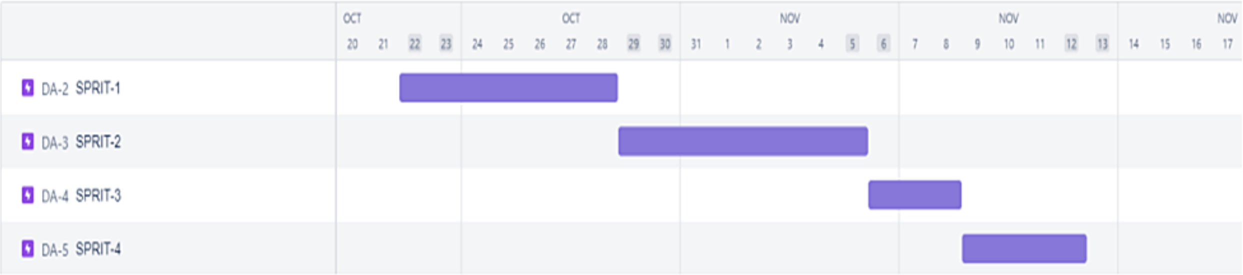
A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.



6.3.2 Jira Progress Chart

Jira is an open-source software tool used for managing project tasks, issue tracking and other issues

1. It is a platform-independent tool; that can be worked with different types of operating systems.
2. As we know, Jira is an issue tracking platform, so it is used by development and technical support teams to get work



CODING & SOLUTIONING

7.1 Feature 1

```
from keras.datasets import mnist
import matplotlib.pyplot as plt
from keras.utils import np_utils
(X_train,y_train),(X_test,y_test) =mnist.load_data()
print(X_train.shape)
print(X_test.shape)
print("The label value is ",y_test[22])
plt.imshow(X_test[22])
print("The label value is ",y_train[27])
plt.imshow(X_train[27])
X_train = X_train.reshape(60000, 28, 28, 1).astype('float32')
X_test = X_test.reshape(10000, 28, 28, 1).astype('float32')
number_of_classes= 10
y_train = np_utils.to_categorical(y_train,number_of_classes)
y_test = np_utils.to_categorical(y_test,number_of_classes)
print("After encoding the value 6 of y_test[22] become", y_test[22])
```

7.2 Feature 2

```
import numpy
import matplotlib.pyplot as plt
from keras.utils import np_utils
from tensorflow.keras.datasets import mnist
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, Dense, Flatten
from tensorflow.keras.optimizers import Adam
(X_train, y_train), (X_test, y_test) = mnist.load_data()
```

```
print(X_train.shape)
print(X_test.shape)
X_train[0]
y_train[0]
plt.imshow(X_train[0])
X_train = X_train.reshape(60000, 28, 28, 1).astype('float32')
X_test = X_test.reshape(10000, 28, 28, 1).astype('float32')
number_of_classes = 10
Y_train = np_utils.to_categorical(y_train, number_of_classes)
Y_test = np_utils.to_categorical(y_test, number_of_classes)
Y_train[0]
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(number_of_classes, activation="softmax"))
model.compile(loss='categorical_crossentropy', optimizer="Adam",
metrics=["accuracy"])
model.fit(X_train, Y_train, batch_size=32, epochs=5,
validation_data=(X_test,Y_test))
metrics = model.evaluate(X_test, Y_test, verbose=0)
print("Metrics (Test Loss & Test Accuracy): ")
print(metrics)
prediction = model.predict(X_test[:4])
print(prediction)
print(numpy.argmax(prediction, axis=1))
print(Y_test[:4])
```

Testing

8.1 Testcases

testcase ID	Feature Type	Component	Test Scenario	Expeced Result	Actual result	Status
HP_TC_ 001	UI	Home Page	Verify UI elements in the Home Page	The Home page must be displayed properly	Working as expected	PASS
HP_TC_ 002	UI	Home Page	Check if the UI elements are displayed properly in different screen sizes	The Home page must be displayed properly in all sizes	The UI is not displayed properly in screen size 2353 x 1651 and 758 x 630	FAIL
HP_TC_ 003	Functional	Home Page	Check if user can upload their file	The input image shouldbe uploaded to the application successfully	Working as expected	PASS
HP_TC_ 004	Functional	Home Page	Check if user cannot uploadunsupported files	The applicationshould notallow userto select a	User is able to upload any file	FAIL

				non imagefile		

8.2 User Acceptance Testing

Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the [ProductName] project at the time of the release to User Acceptance Testing (UAT).

Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	2	3	20
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	24	14	13	26	7

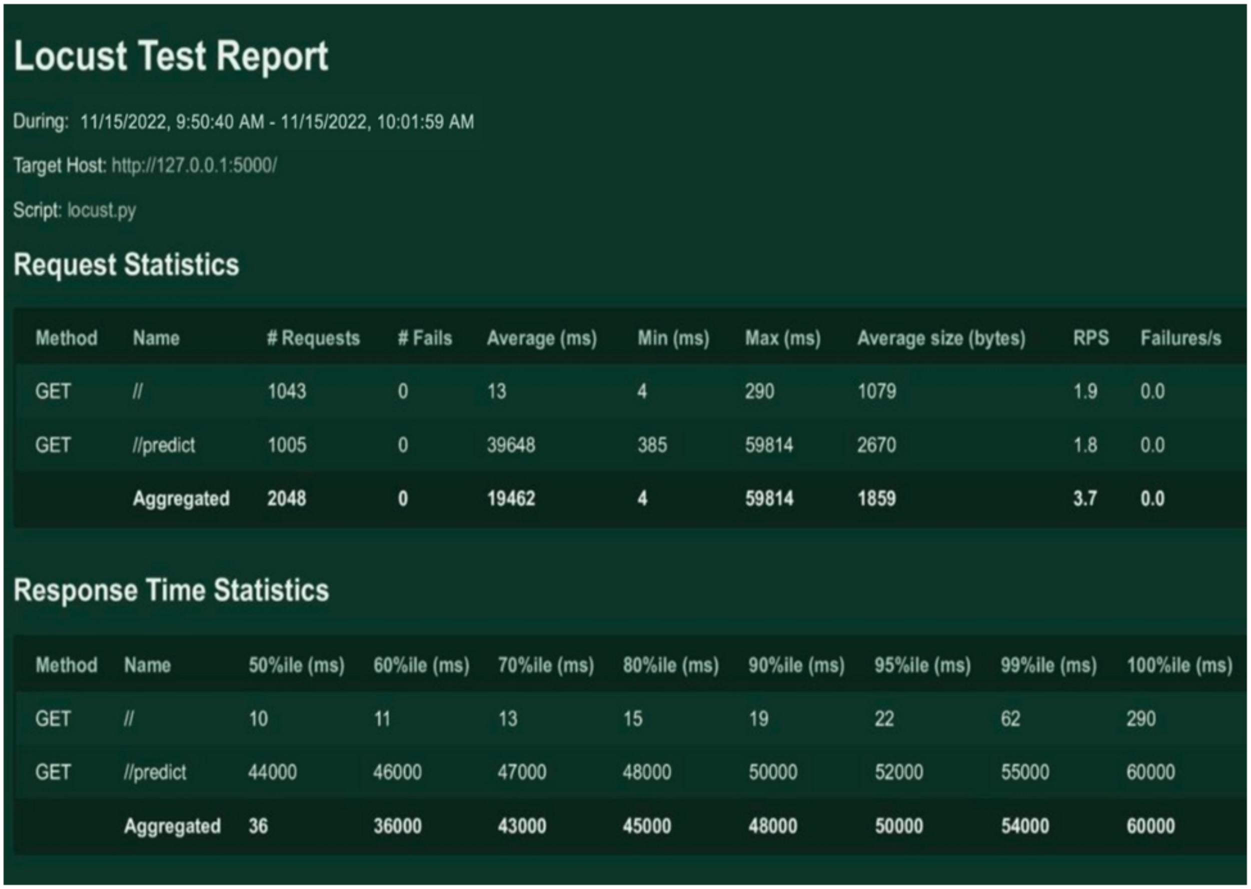
Test Case Analysis

This report shows the number of test cases that have passed, failed, and untested

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	51	0	0	51
Security	2	0	0	2
Outsource Shipping	3	0	0	3
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

Result

9.1 Performance and Metrices



Locust Test Report

During: 11/15/2022, 9:50:40 AM - 11/15/2022, 10:01:59 AM

Target Host: http://127.0.0.1:5000/

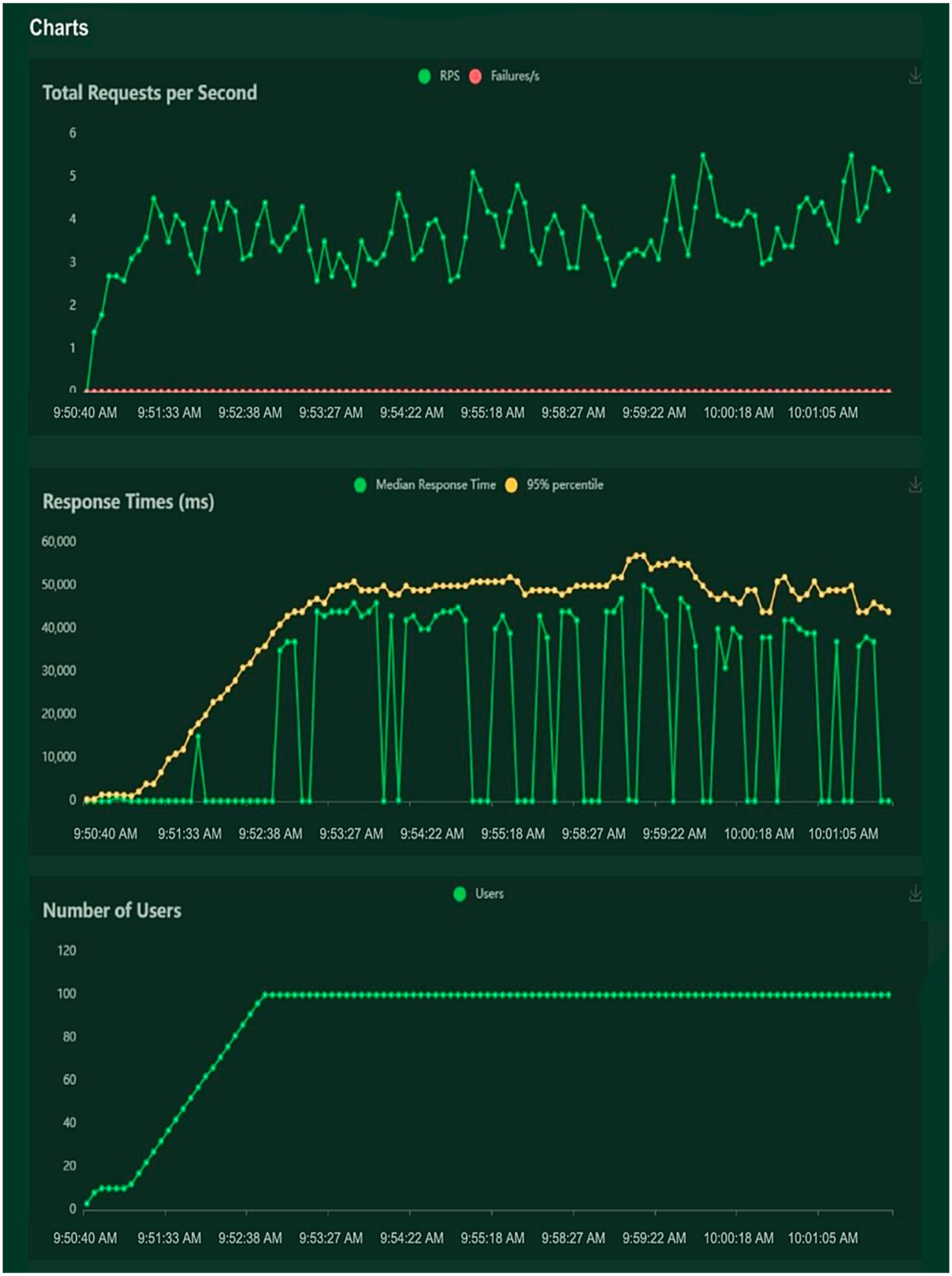
Script: locust.py

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**



Charts

Total Requests per Second

Response Times (ms)

Number of Users

Advantages and Disadvantages

10.1 ADVANTAGES

1. Reduces manual work
2. Backups
3. More accurate than average human
4. Capable of handlinga lot of data
5. Can be used anywhere from any device

10.2 DISADVANTAGES

1. Cannot handle complexdata
2. Low retention

3. All the data mustbe in digital format

4 .Requires a high performance server for faster prediction

5. Prone to occasional errors

Conclusion

This projectdemonstrated a web application that uses machinelearning to recognise 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 devicethat can run a browser. This project is extremely useful in real-world scenarios such as recognizing number plates of vehicles, processing bank cheque amounts, numeric entries in forms filled up by hand (tax forms) and soon. There is so much room for improvement, which can be implemented in subsequent versions.

Future Scope

This project is far from complete and there is a lot

of room forimprovement.Some of the improvements

that can be made to this project are as follows:

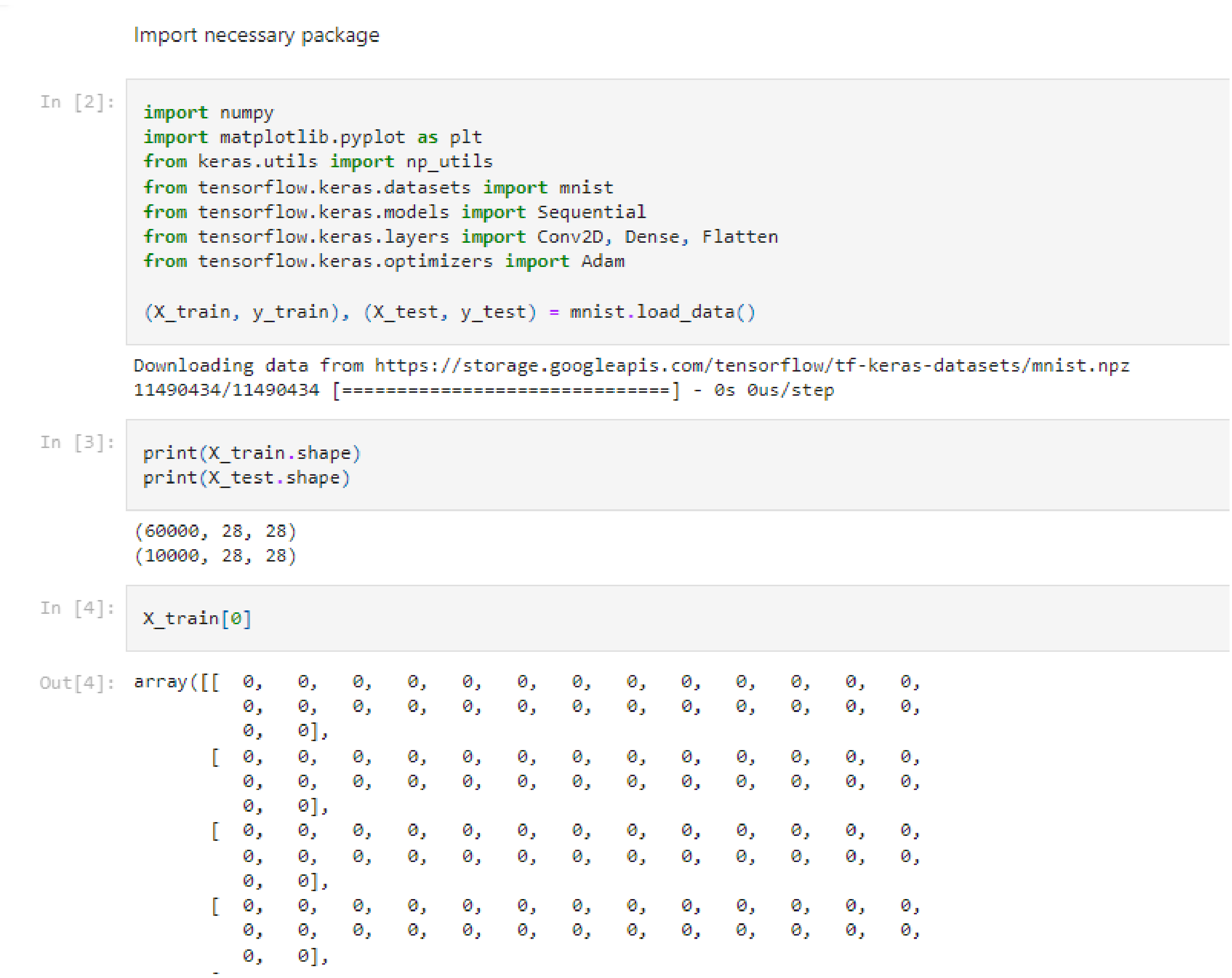
1. Add support to detect from digits multipleimages and save the results
2. Add support to detect multipledigits
3. Improve model to detect digits from complex images
4. Add support to different languages to help usersfrom all over the world

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

Appendix

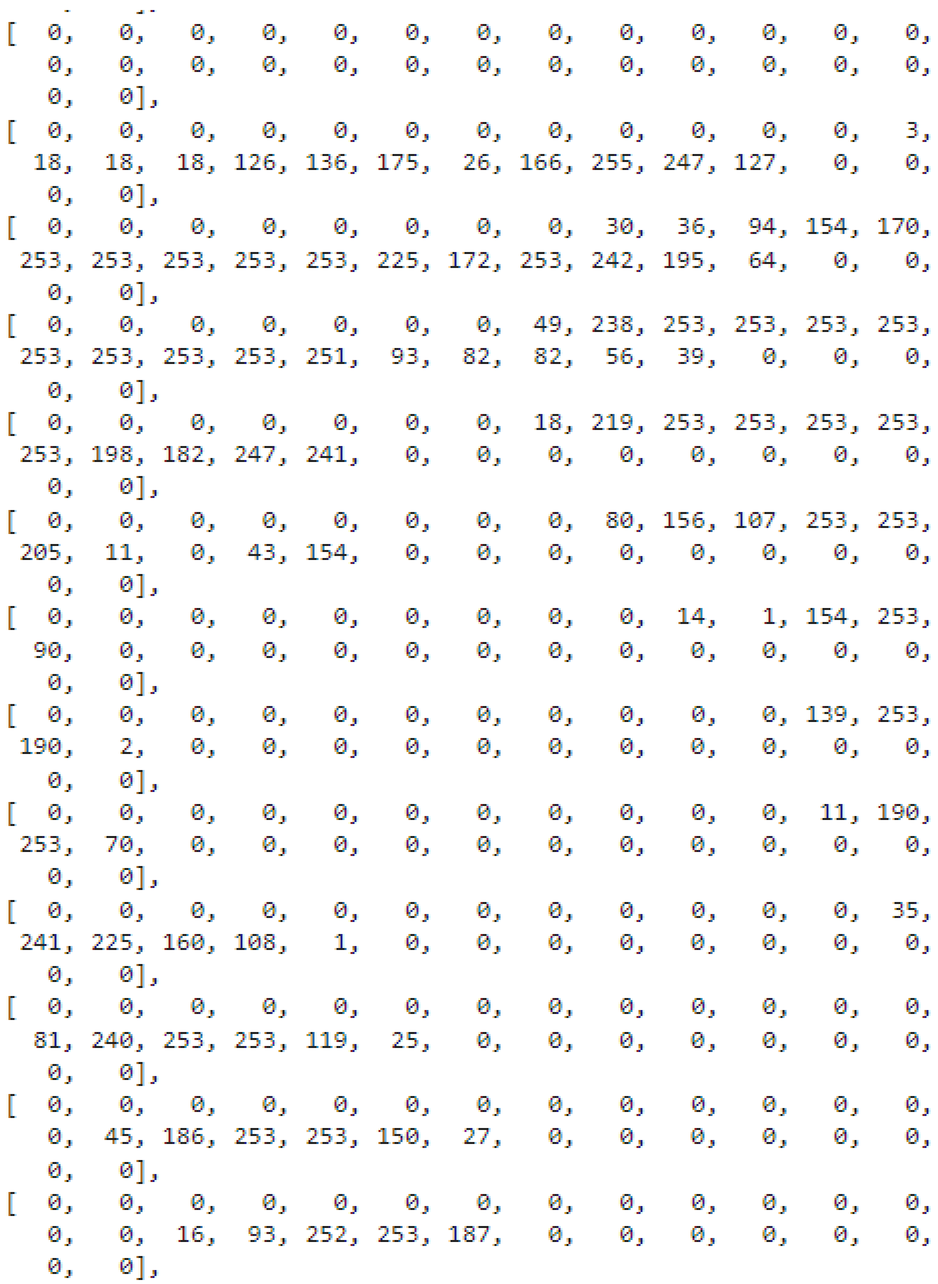
Source Code

Compiling the model

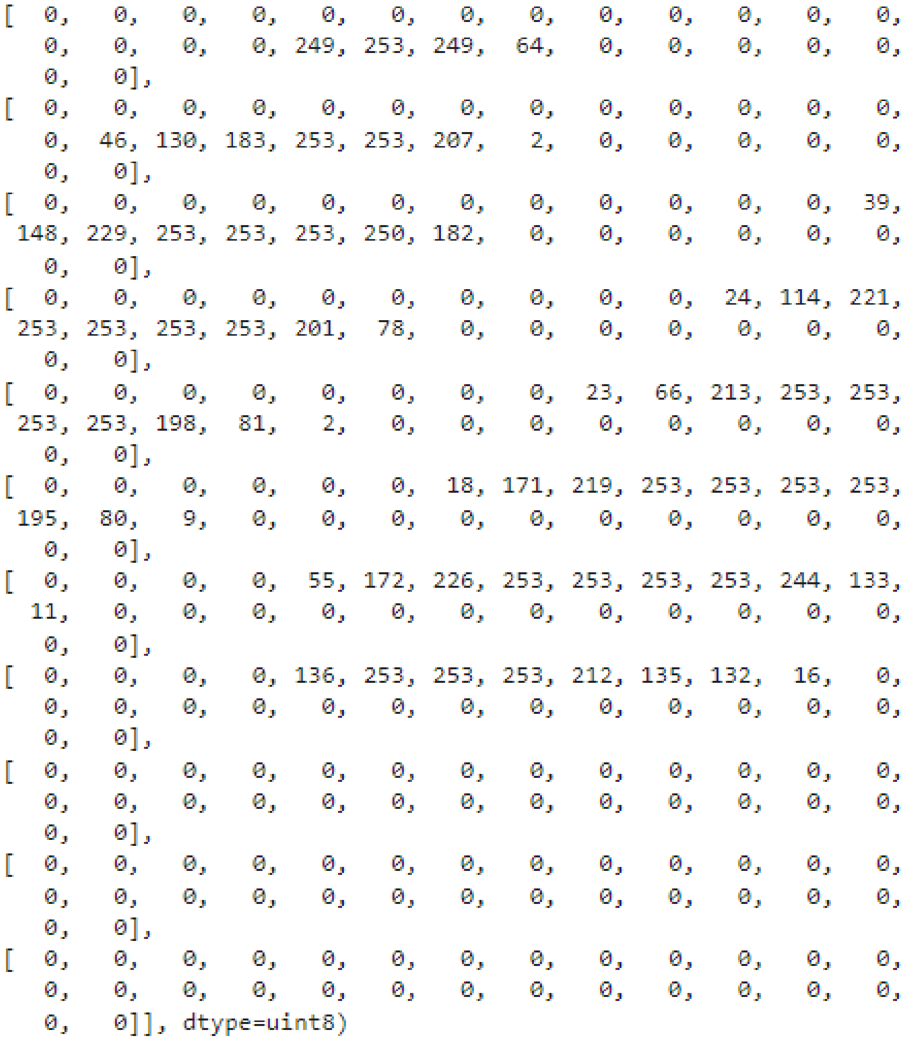


Import necessary package

```
In [2]:  import numpy
         import matplotlib.pyplot as plt
         from keras.utils import np_utils
         from tensorflow.keras.datasets import mnist
         from tensorflow.keras.models import Sequential
         from tensorflow.keras.layers import Conv2D, Dense, Flatten
         from tensorflow.keras.optimizers import Adam

         (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 [==============================] - 0s 0us/step

In [3]:  print(X_train.shape)
         print(X_test.shape)

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

In [4]:  X_train[0]

Out[4]:  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,   3,
  18,  18,  18, 126, 136, 175,  26, 166, 255, 247, 127,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,  30,  36,  94, 154, 170,
 253, 253, 253, 253, 253, 225, 172, 253, 242, 195,  64,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,  49, 238, 253, 253, 253, 253,
 253, 253, 253, 253, 251,  93,  82,  82,  56,  39,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,  18, 219, 253, 253, 253, 253,
 253, 198, 182, 247, 241,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,  80, 156, 107, 253, 253,
 205,  11,   0,  43, 154,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,   0,  14,   1, 154, 253,
  90,   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, 139, 253,
 190,   2,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,  11, 190,
 253,  70,   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,  35,
 241, 225, 160, 108,   1,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
  81, 240, 253, 253, 119,  25,   0,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,  45, 186, 253, 253, 150,  27,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0,  16,  93, 252, 253, 187,   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, 249, 253, 249,  64,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,  46, 130, 183, 253, 253, 207,   2,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,  39,
 148, 229, 253, 253, 253, 250, 182,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,  24, 114, 221,
 253, 253, 253, 253, 201,  78,   0,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,  23,  66, 213, 253, 253,
 253, 253, 198,  81,   2,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,  18, 171, 219, 253, 253, 253, 253,
 195,  80,   9,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,  55, 172, 226, 253, 253, 253, 253, 244, 133,
  11,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0, 136, 253, 253, 253, 212, 135, 132,  16,   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]], dtype=uint8)
```

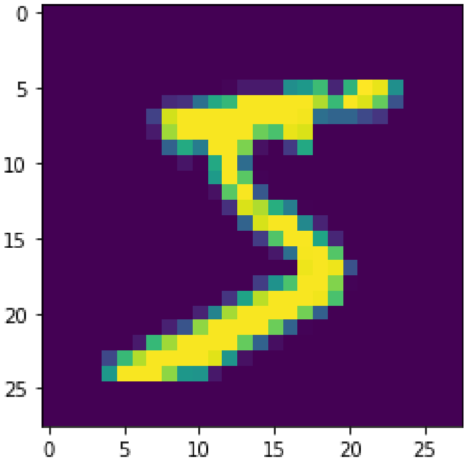
Data pre processing

```
In [5]: y_train[0]
```

```
Out[5]: 5
```

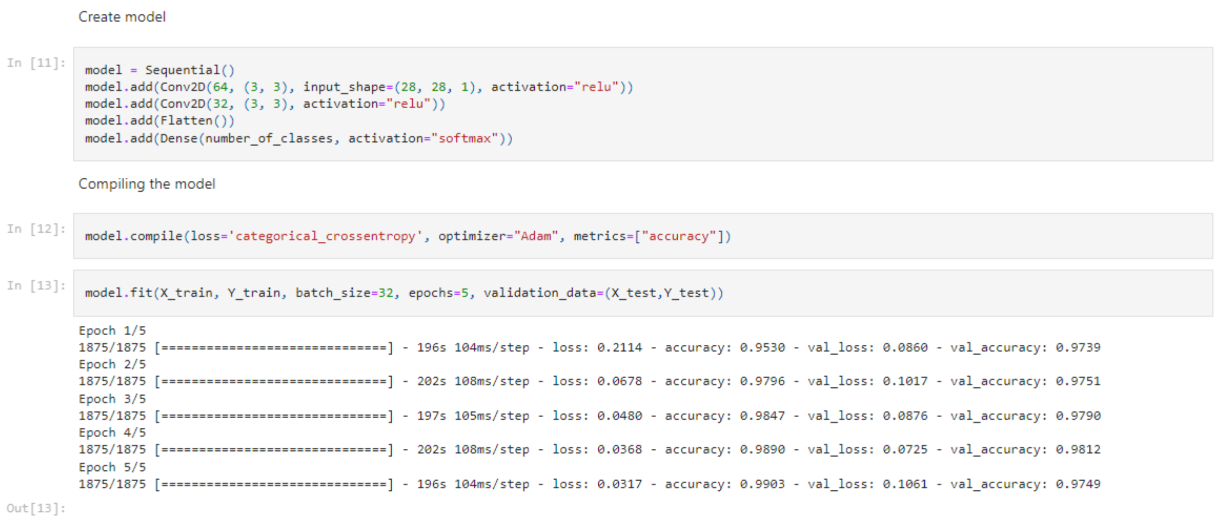
```
In [6]: plt.imshow(X_train[0])
```

```
Out[6]:
```



```
In [9]: X_train = X_train.reshape(60000, 28, 28, 1).astype('float32')
        X_test = X_test.reshape(10000, 28, 28, 1).astype('float32')
        number_of_classes = 10
        Y_train = np_utils.to_categorical(y_train, number_of_classes)
        Y_test = np_utils.to_categorical(y_test, number_of_classes)
        Y_train[0]
```

```
Out[9]: array([0., 0., 0., 0., 0., 1., 0., 0., 0., 0.], dtype=float32)
```



Create model

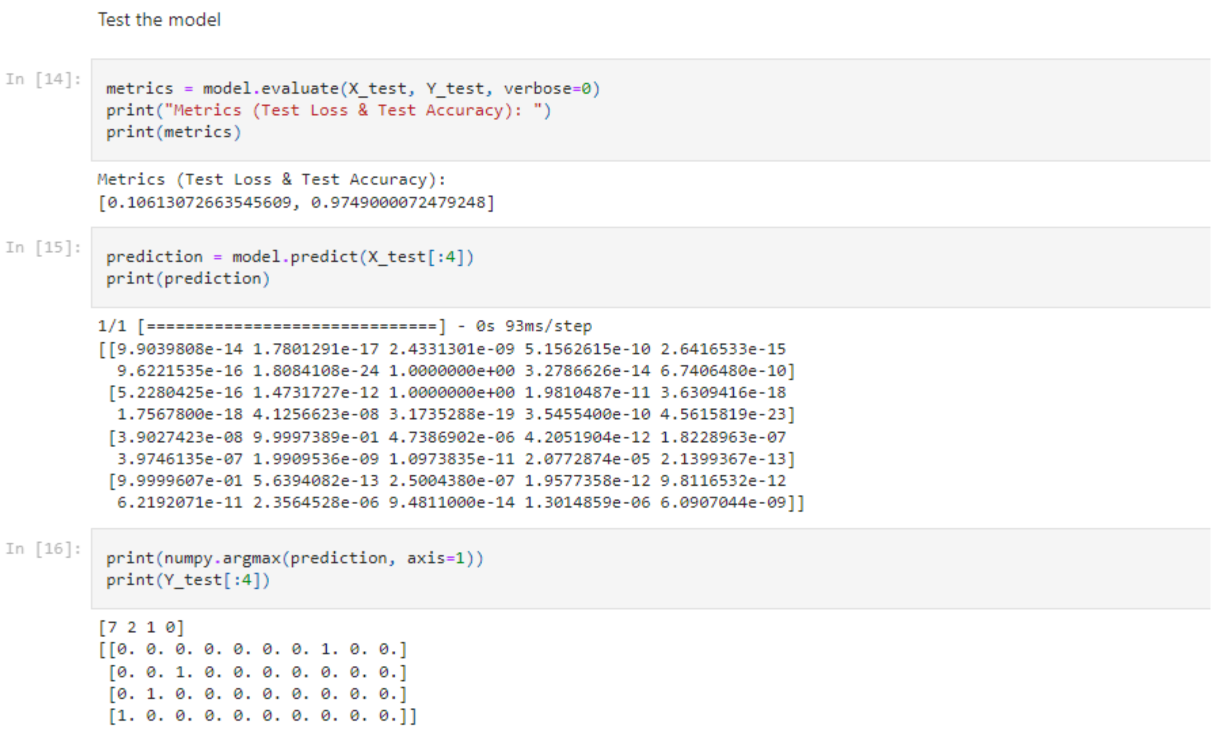
```
In [11]:
model = Sequential()
model.add(Conv2D(64, (3, 3), input_shape=(28, 28, 1), activation="relu"))
model.add(Conv2D(32, (3, 3), activation="relu"))
model.add(Flatten())
model.add(Dense(number_of_classes, activation="softmax"))
```

Compiling the model

```
In [12]:
model.compile(loss='categorical_crossentropy', optimizer="Adam", metrics=["accuracy"])
```

```
In [13]:
model.fit(X_train, Y_train, batch_size=32, epochs=5, validation_data=(X_test,Y_test))
```

```
Epoch 1/5
1875/1875 [==============================] - 196s 104ms/step - loss: 0.2114 - accuracy: 0.9530 - val_loss: 0.0860 - val_accuracy: 0.9739
Epoch 2/5
1875/1875 [==============================] - 202s 108ms/step - loss: 0.0678 - accuracy: 0.9796 - val_loss: 0.1017 - val_accuracy: 0.9751
Epoch 3/5
1875/1875 [==============================] - 197s 105ms/step - loss: 0.0480 - accuracy: 0.9847 - val_loss: 0.0876 - val_accuracy: 0.9790
Epoch 4/5
1875/1875 [==============================] - 202s 108ms/step - loss: 0.0368 - accuracy: 0.9890 - val_loss: 0.0725 - val_accuracy: 0.9812
Epoch 5/5
1875/1875 [==============================] - 196s 104ms/step - loss: 0.0317 - accuracy: 0.9903 - val_loss: 0.1061 - val_accuracy: 0.9749
Out[13]:
```



Test the model

```
In [14]:
metrics = model.evaluate(X_test, Y_test, verbose=0)
print("Metrics (Test Loss & Test Accuracy): ")
print(metrics)
```

```
Metrics (Test Loss & Test Accuracy):
[0.10613072663545609, 0.9749000072479248]
```

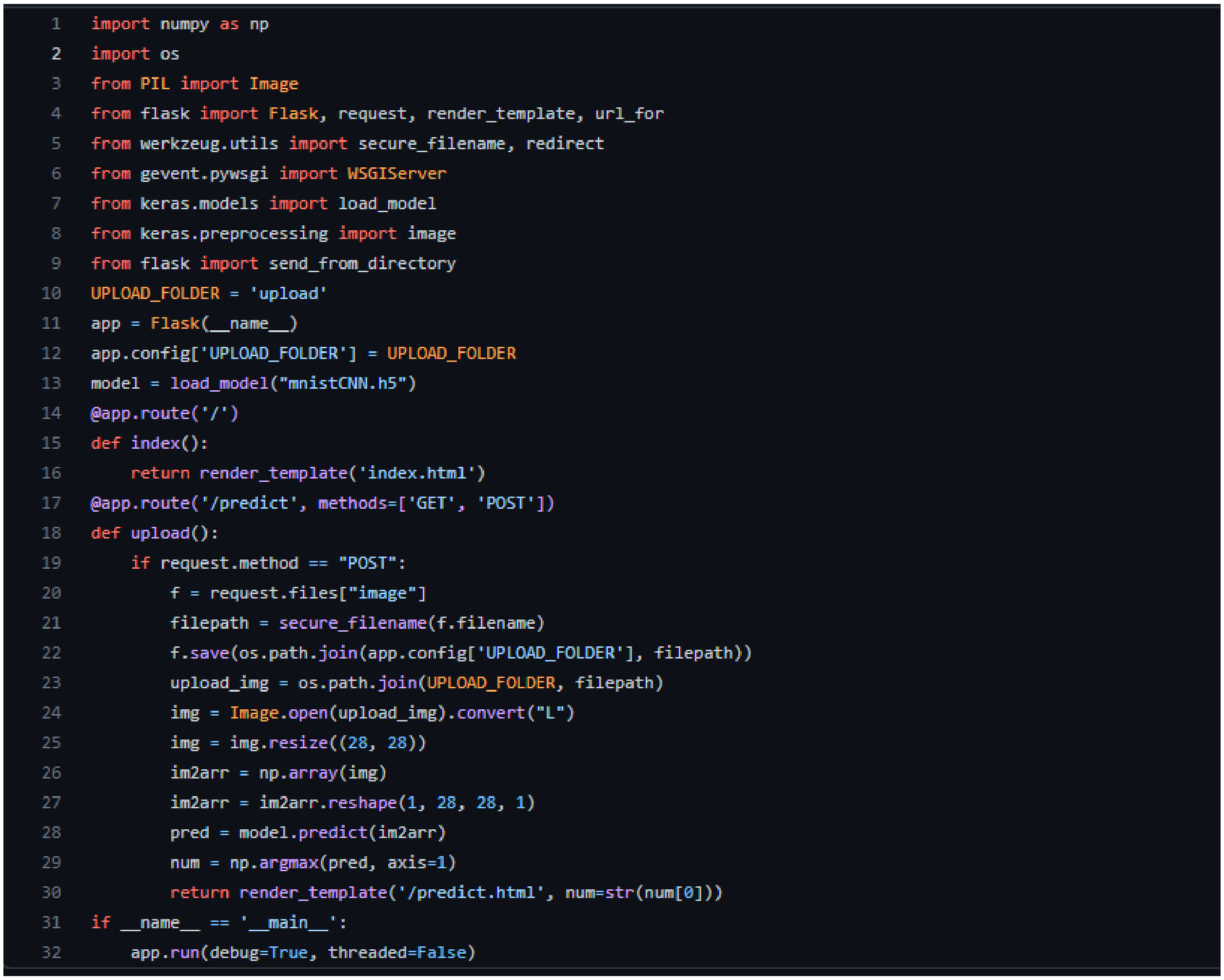
```
In [15]:
prediction = model.predict(X_test[:4])
print(prediction)
```

```
1/1 [==============================] - 0s 93ms/step
[[9.9039808e-14 1.7801291e-17 2.4331301e-09 5.1562615e-10 2.6416533e-15
  9.6221535e-16 1.8084108e-24 1.0000000e+00 3.2786626e-14 6.7406480e-10]
 [5.2280425e-16 1.4731727e-12 1.0000000e+00 1.9810487e-11 3.6309416e-18
  1.7567800e-18 4.1256623e-08 3.1735288e-19 3.5455400e-10 4.5615819e-23]
 [3.9027423e-08 9.9997389e-01 4.7386902e-06 4.2051904e-12 1.8228963e-07
  3.9746135e-07 1.9909536e-09 1.0973835e-11 2.0772874e-05 2.1399367e-13]
 [9.9999607e-01 5.6394082e-13 2.5004380e-07 1.9577358e-12 9.8116532e-12
  6.2192071e-11 2.3564528e-06 9.4811000e-14 1.3014859e-06 6.0907044e-09]]
```

```
In [16]:
print(numpy.argmax(prediction, axis=1))
print(Y_test[:4])
```

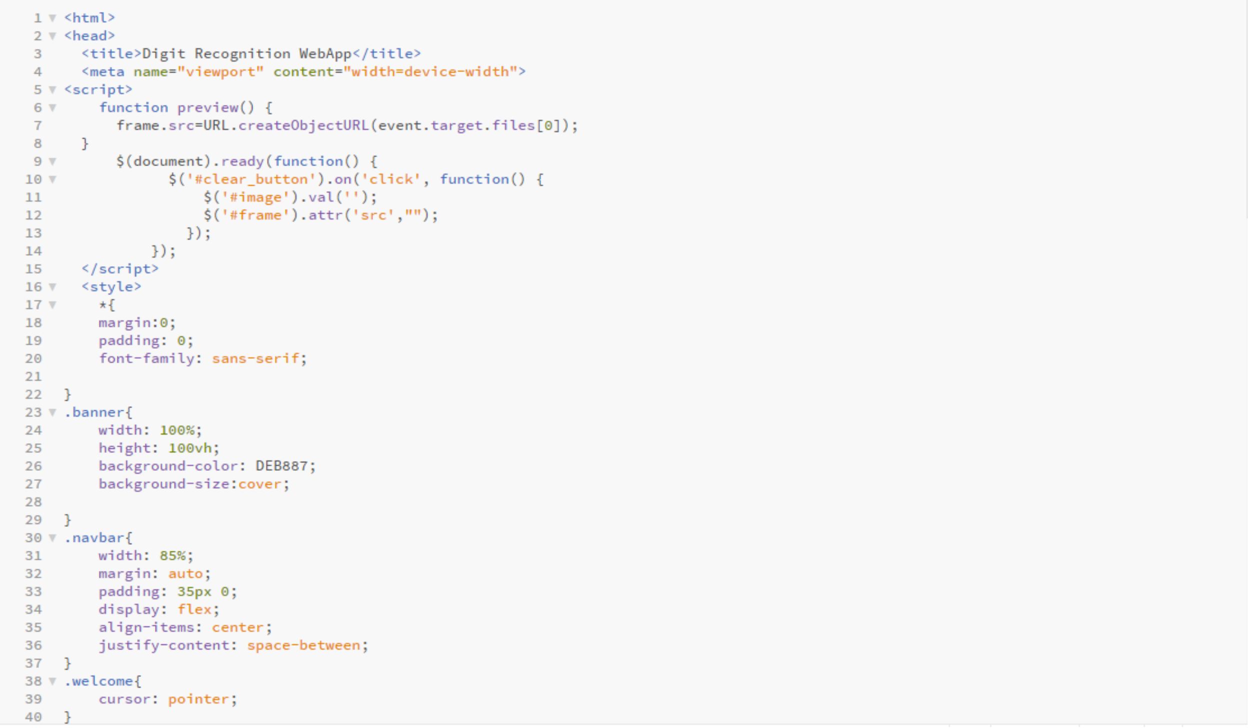
```
[7 2 1 0]
[[0. 0. 0. 0. 0. 0. 0. 1. 0. 0.]
 [0. 0. 1. 0. 0. 0. 0. 0. 0. 0.]
 [0. 1. 0. 0. 0. 0. 0. 0. 0. 0.]
 [1. 0. 0. 0. 0. 0. 0. 0. 0. 0.]]
```

App (python)

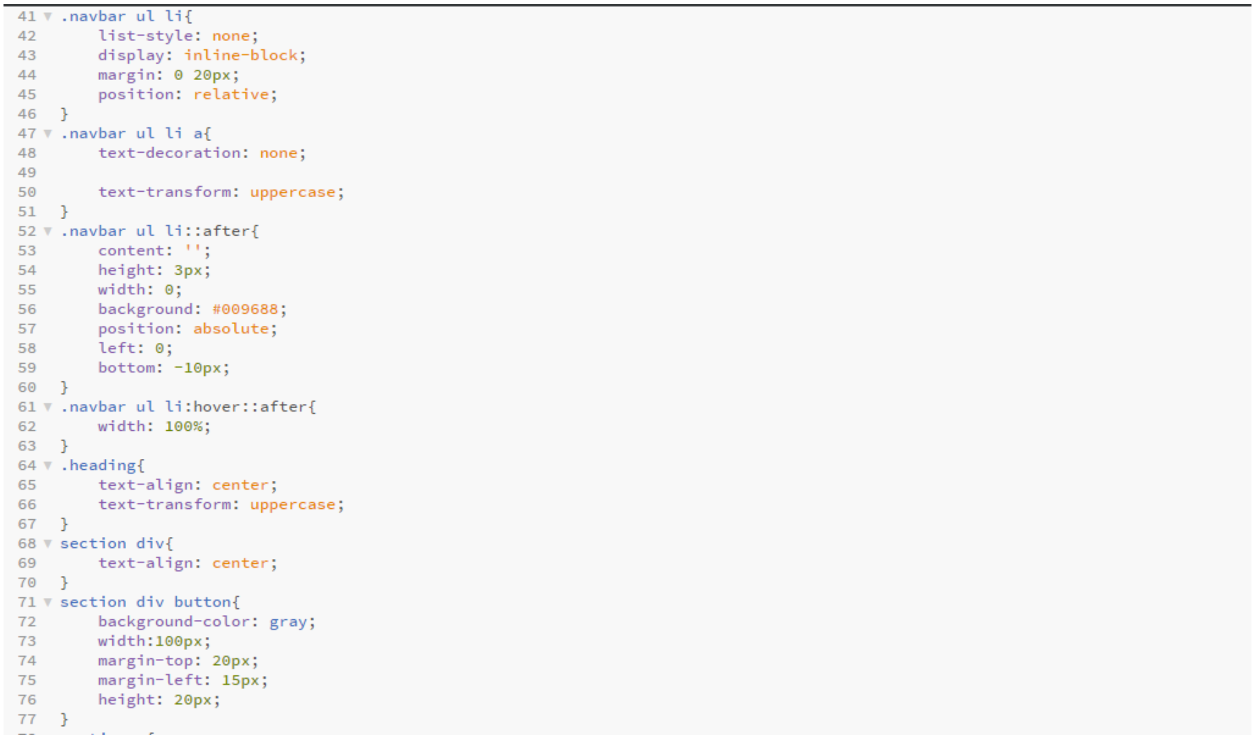


```python
import numpy as np
import os
from PIL import Image
from flask import Flask, request, render_template, url_for
from werkzeug.utils import secure_filename, redirect
from gevent.pywsgi import WSGIServer
from keras.models import load_model
from keras.preprocessing import image
from flask import send_from_directory
UPLOAD_FOLDER = 'upload'
app = Flask(__name__)
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER
model = load_model("mnistCNN.h5")
@app.route('/')
def index():
    return render_template('index.html')
@app.route('/predict', methods=['GET', 'POST'])
def upload():
    if request.method == "POST":
        f = request.files["image"]
        filepath = secure_filename(f.filename)
        f.save(os.path.join(app.config['UPLOAD_FOLDER'], filepath))
        upload_img = os.path.join(UPLOAD_FOLDER, filepath)
        img = Image.open(upload_img).convert("L")
        img = img.resize((28, 28))
        im2arr = np.array(img)
        im2arr = im2arr.reshape(1, 28, 28, 1)
        pred = model.predict(im2arr)
        num = np.argmax(pred, axis=1)
        return render_template('/predict.html', num=str(num[0]))
if __name__ == '__main__':
    app.run(debug=True, threaded=False)
```

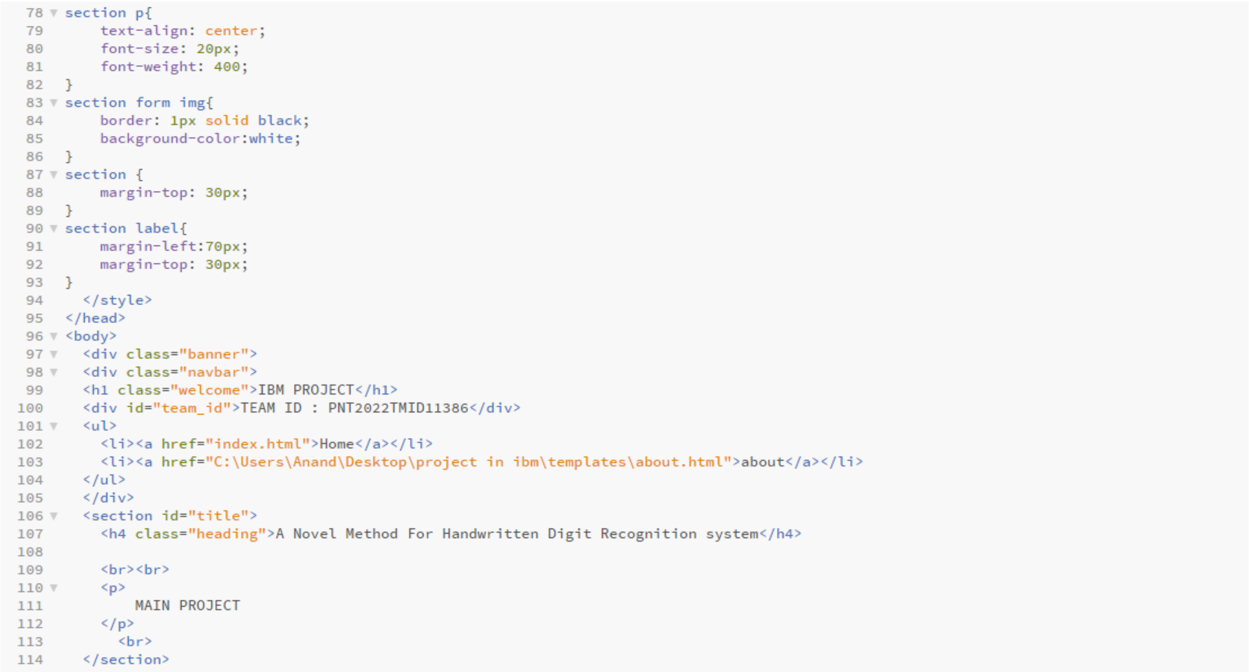
Home Page (HTML)



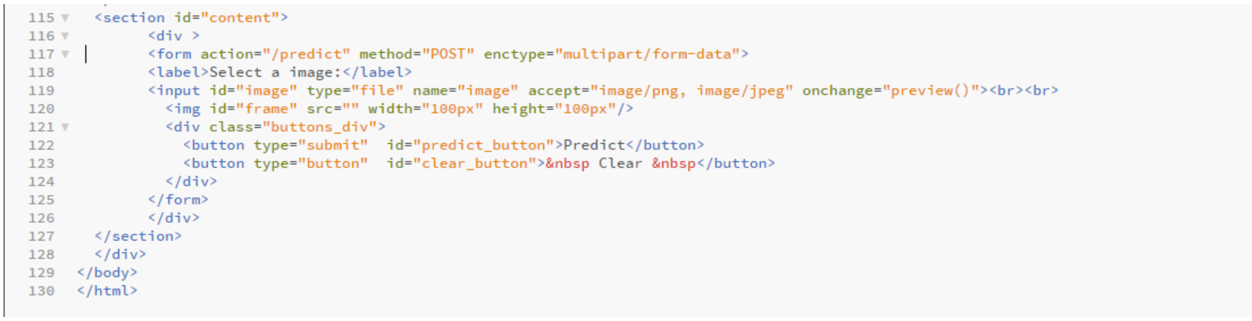
```
<html>
<head>
  <title>Digit Recognition WebApp</title>
  <meta name="viewport" content="width=device-width">
<script>
    function preview() {
      frame.src=URL.createObjectURL(event.target.files[0]);
  }
      $(document).ready(function() {
            $('#clear_button').on('click', function() {
                $('#image').val('');
                $('#frame').attr('src',"");
              });
          });
  </script>
  <style>
    *{
    margin:0;
    padding: 0;
    font-family: sans-serif;

}
.banner{
    width: 100%;
    height: 100vh;
    background-color: DEB887;
    background-size:cover;

}
.navbar{
    width: 85%;
    margin: auto;
    padding: 35px 0;
    display: flex;
    align-items: center;
    justify-content: space-between;
}
.welcome{
    cursor: pointer;
}
```



```
.navbar ul li{
    list-style: none;
    display: inline-block;
    margin: 0 20px;
    position: relative;
}
.navbar ul li a{
    text-decoration: none;

    text-transform: uppercase;
}
.navbar ul li::after{
    content: '';
    height: 3px;
    width: 0;
    background: #009688;
    position: absolute;
    left: 0;
    bottom: -10px;
}
.navbar ul li:hover::after{
    width: 100%;
}
.heading{
    text-align: center;
    text-transform: uppercase;
}
section div{
    text-align: center;
}
section div button{
    background-color: gray;
    width:100px;
    margin-top: 20px;
    margin-left: 15px;
    height: 20px;
}
```

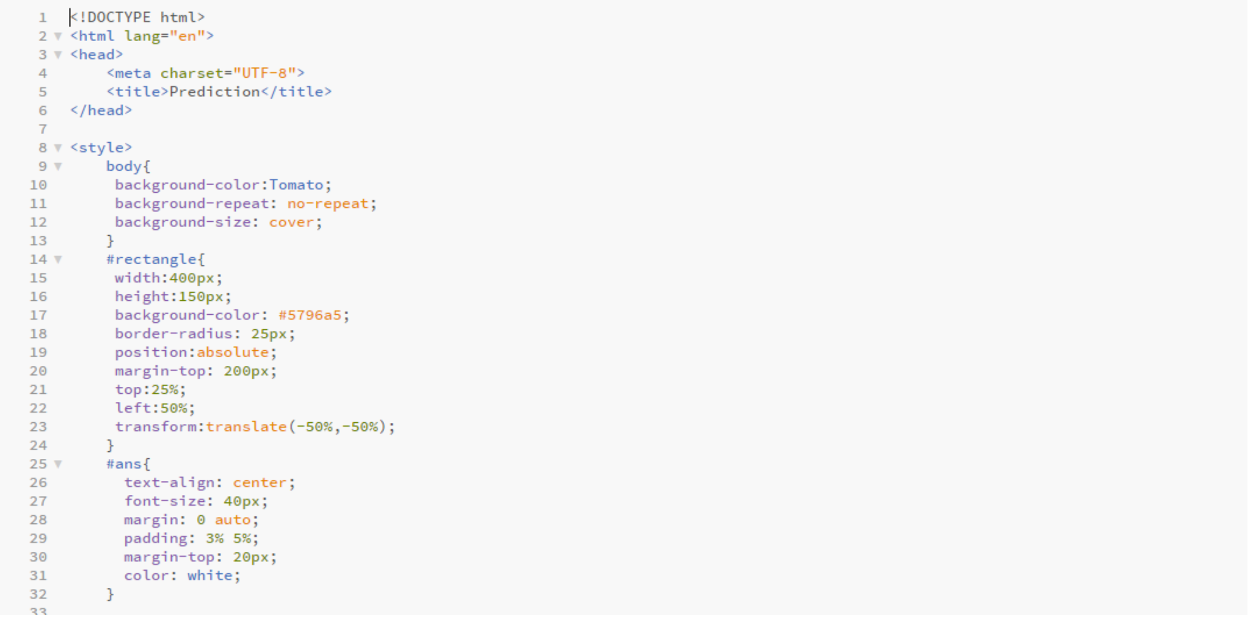


```
section p{
    text-align: center;
    font-size: 20px;
    font-weight: 400;
}
section form img{
    border: 1px solid black;
    background-color:white;
}
section {
    margin-top: 30px;
}
section label{
    margin-left:70px;
    margin-top: 30px;
}
  </style>
</head>
<body>
  <div class="banner">
  <div class="navbar">
  <h1 class="welcome">IBM PROJECT</h1>
  <div id="team_id">TEAM ID : PNT2022TMID11386</div>
  <ul>
    <li><a href="index.html">Home</a></li>
    <li><a href="C:\Users\Anand\Desktop\project in ibm\templates\about.html">about</a></li>
  </ul>
  </div>
  <section id="title">
    <h4 class="heading">A Novel Method For Handwritten Digit Recognition system</h4>

    <br><br>
    <p>
        MAIN PROJECT
    </p>
      <br>
  </section>
```

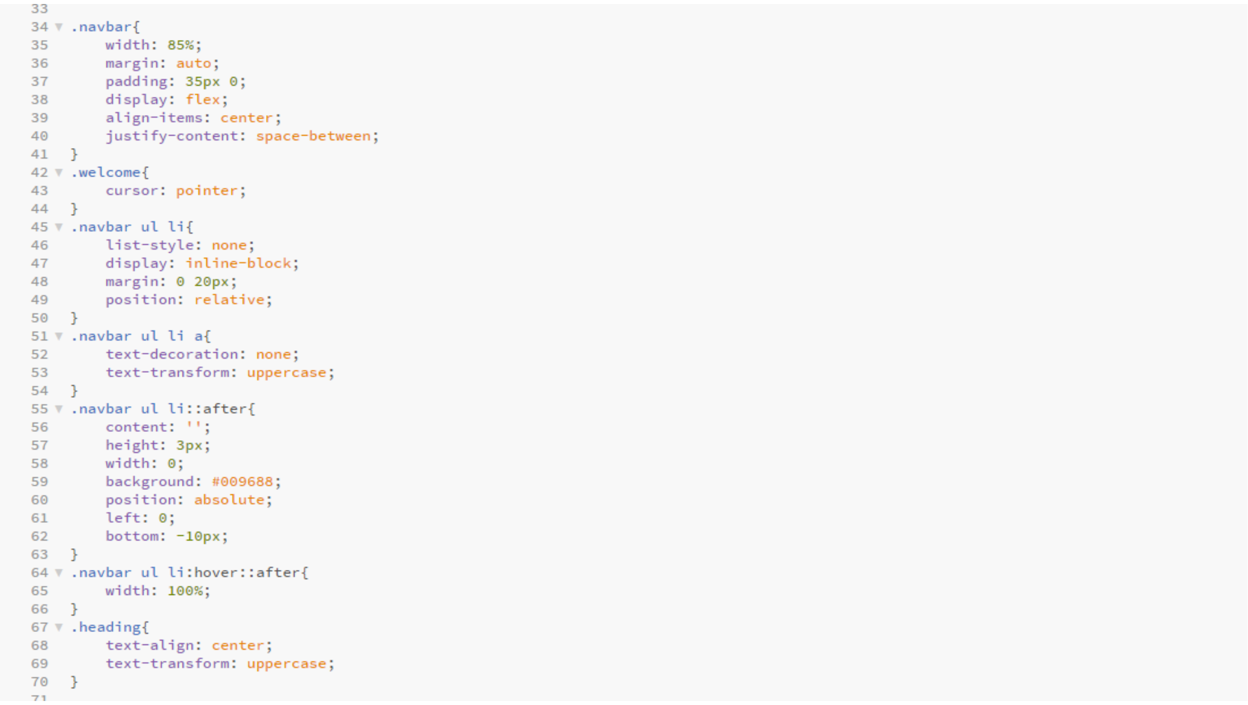


```
  <section id="content">
        <div >
        <form action="/predict" method="POST" enctype="multipart/form-data">
        <label>Select a image:</label>
        <input id="image" type="file" name="image" accept="image/png, image/jpeg" onchange="preview()"><br><br>
          <img id="frame" src="" width="100px" height="100px"/>
          <div class="buttons_div">
            <button type="submit"  id="predict_button">Predict</button>
            <button type="button"  id="clear_button">&nbsp Clear &nbsp</button>
          </div>
        </form>
        </div>
  </section>
  </div>
</body>
</html>
```

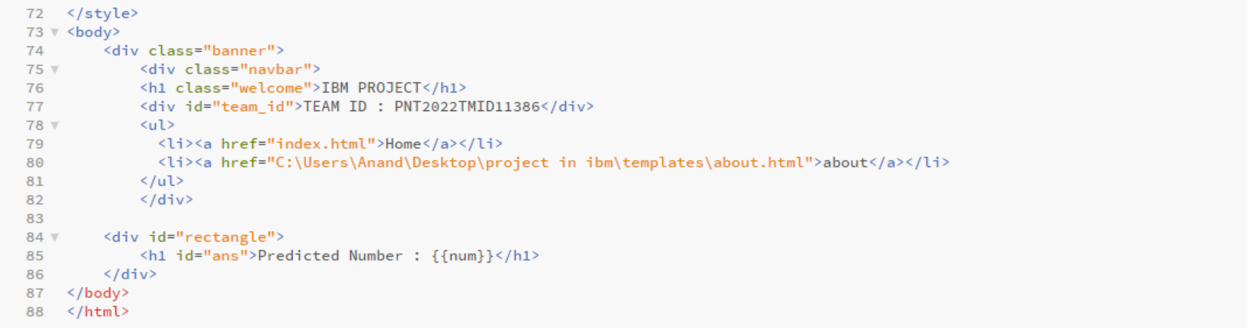
Predict Page (HTML)



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Prediction</title>
</head>

<style>
    body{
     background-color:Tomato;
     background-repeat: no-repeat;
     background-size: cover;
    }
    #rectangle{
     width:400px;
     height:150px;
     background-color: #5796a5;
     border-radius: 25px;
     position:absolute;
     margin-top: 200px;
     top:25%;
     left:50%;
     transform:translate(-50%,-50%);
    }
    #ans{
      text-align: center;
      font-size: 40px;
      margin: 0 auto;
      padding: 3% 5%;
      margin-top: 20px;
      color: white;
    }
```



```

.navbar{
    width: 85%;
    margin: auto;
    padding: 35px 0;
    display: flex;
    align-items: center;
    justify-content: space-between;
}
.welcome{
    cursor: pointer;
}
.navbar ul li{
    list-style: none;
    display: inline-block;
    margin: 0 20px;
    position: relative;
}
.navbar ul li a{
    text-decoration: none;
    text-transform: uppercase;
}
.navbar ul li::after{
    content: '';
    height: 3px;
    width: 0;
    background: #009688;
    position: absolute;
    left: 0;
    bottom: -10px;
}
.navbar ul li:hover::after{
    width: 100%;
}
.heading{
    text-align: center;
    text-transform: uppercase;
}
```



```html
</style>
<body>
    <div class="banner">
        <div class="navbar">
        <h1 class="welcome">IBM PROJECT</h1>
        <div id="team_id">TEAM ID : PNT2022TMID11386</div>
        <ul>
          <li><a href="index.html">Home</a></li>
          <li><a href="C:\Users\Anand\Desktop\project in ibm\templates\about.html">about</a></li>
        </ul>
        </div>

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

Github link :

https://github.com/IBM-EPBL/IBM-Project-34479-1660236319

Project Demo Video:

https://drive.google.com/file/d/1vn7CgIgCeIXMkxp8Q8dCQ329D_00VFCA/view?usp=sharing