PROJECT REPORT

A NOVEL METHOD FOR
HANDWRITTEN DIGIT RECOGNITION

submitted by

PNT2022TMID11565

Team Head:

Athisankaran B

Team Member:

1.Anandan N.P

2.Chitharanjan C.K

3.Darshan A.S

4.Jaya karthikeiyan C

TABLE OF CONTENTS

1 INTRODUCTION 1

1.1 PROJECT OVERVIEW 1

1.2 PURPOSE 1

2 LITERATURE SURVEY 2

2.1 EXISTING PROBLEM 2

2.2 REFERENCES 2

2.3 PROBLEM STATEMENT DEFINITION 4

3 IDEATION AND PROPOSED SOLUTION 6

3.1 EMPATHY MAP CANVAS 5

3.2 IDEATION & BRAINSTORMING 6

3.3 PROPOSED SOLUTION 7

3.4 PROBLEM SOLUTION FIT 8

4 REQUIREMENT ANALYSIS 9

4.1 FUNCTIONAL REQUIREMENTS 9

4.2 NON-FUNCTIONAL REQUIREMENTS 10

5 PROJECT DESIGN 11

5.1 DATA FLOW DIAGRAM 11

5.2 SOLUTION & TECHNICAL ARCHITECTURE 13

5.3 COMPOMENTS AND TECHNOLOGIES 14

5.4 USER STORIES 15

6 PROJECT PLANNING AND SCHEDULING 16

6.1 SPRINT PLANNING AND ESTIMATION 16

6.2 SPRINT DELIVERY SCHEDULE 18

6.3 REPORT FROM JIRA 18

5 CODING & SOLUTIONING 19

7.1 FEATURE 1 19

7.2 FEATURE 2 20

7.3 FEATURE 3 20

6 TESTING 21

8.1 TEST CASES 21

8.2 USER ACCEPTANCE TESTING 23

7 RESULTS 24

9.1 PERFORMANCE METRICS 24

8 ADVANTAGES & DISADVANTAGES 26

ADVANTAGES 26

DISADVANTAGES 26

9 CONCLUSION 27

10 FUTURE SCOPE 28

APPENDIX 29

SOURCE CODE 29

SCREENSHOTS 39

GITHUB 40

CHAPTER 1
INTRODUCTION

1.1 PROJECT OVERVIEW

Handwritten Digit Recognition is the capacity of a computer to interpret the manually written digits from various sources like messages, bank cheques, papers, pictures, and so forth and in various situations for web-based handwriting recognition on PC tablets, identifying number plates of vehicles, handling bank cheques, digits entered in any forms etc. Machine Learning provides various methods through which human efforts can be reduced in recognizing the manually written digits.

Deep Learning is a machine learning method that trains computers to do what easily falls into place for people: learning through examples. With the utilization of deep learning methods, human attempts can be diminished in perceiving, learning, recognizing and in a lot more regions. Using deep learning, the computer learns to carry out classification works from pictures or contents from any document. Deep Learning models can accomplish state-of-art accuracy, beyond the human level performance. The digit recognition model uses large datasets in order to recognize digits from distinctive sources.

1.2 PURPOSE

The main objective was to actualize a pattern characterization method to perceive the handwritten digits provided in the MINIST data set of images of handwritten digits (0-9). The goal of our work is to create a model that will be able to recognize and classify the handwritten digits from images by using concepts of Convolution Neural Network. Though the goal of our research is to create a model for digit recognition and classification, it can also be extended to letters and an individual’s handwriting. With high accuracy rates, the model can solve a lot of real life problems.

The main applications are vehicle license-plate recognition, postal letter-sorting services, Cheque truncation system (CTS) scanning and historical document preservation in archaeology departments, old documents automation in libraries and banks, etc. All these areas deal with large databases and hence demand high recognition accuracy, lesser computational complexity and consistent performance of the recognition system.

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 margins since they vary from person to person. People can struggle to read others' handwriting. The handwritten digits are not always of the same size, width, orientation as they differ from writing of person to person, so the general problem would be while classifying the digits.

Additionally, there would be issues with identifying the numbers because of similarities between numerals like 1 and 7, 5 and 6, 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

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

Handwritten digits recognition is an area of machine learning, in which a machine is trained to identify handwritten digits. One method of achieving this is with decision tree classification model. A decision tree classification is a machine learning approach that uses the predefined labels from the past known sets to determine or predict the classes of the future data sets where the class labels are unknown. In this paper we have used the standard kaggle digits dataset for recognition of handwritten digits using a decision tree classification approach. And we have evaluated the accuracy of the model against each digit from 0 to 9

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

Handwritten digit recognition is a technique or technology for automatically recognizing and detecting handwritten digital data through different Machine Learning models. In this paper we use various Machine Learning algorithms to enhance the productiveness of technique and reduce the complexity using various models. Machine Learning is an application of Artificial Intelligence that learns from previous experience and improves automatically through experience. We illustrate various Machine learning algorithms such as Support Vector Machine, Convolutional Neural Network, Quantum Computing, K-Nearest Neighbor Algorithm, Deep Learning used in Recognition technique. KEYWORDSConvolutional Neural Network, Support Vector Machine, HandWritten Digit Recognition, Artificial Intelligence, Deep Learning.

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

This paper presents a model of integrating the synergy of two superior classifiers: Convolutional Neural Network (CNN) and Random Forest Classifier (RFC), which have proven results in recognizing different types of patterns. Handwritten digit recognition is one of the practically important issues in pattern recognition applications. The applications of digit recognition include in postal mail sorting, bank check processing, form data entry, etc. The heart of the problem lies within the ability to develop an efficient algorithm that can recognize hand written digits and which is submitted by users by the way of a scanner, tablet, and other digital devices. The problem of handwritten digit recognition has long been an open problem in the field of pattern classification. Several studies have shown that Neural Network has a great performance in data classification. Ability for accurate digit recognizer modelling and prediction is critical for pattern recognition and security. A variety of classification machine learning algorithms are known to be effective for digit recognition

4.)**"Handwritten Digit String Recognition using Convolutional Neural Network"-honhjianzhan -2019**

String recognition is one of the most important tasks in computer vision applications. Recently the combinations of convolutional neural network (CNN) and recurrent neural network (RNN) have been widely applied to deal with the issue of string recognition. However RNNs are not only hard to train but also time-consuming. In this paper, we propose a new architecture which is based on CNN only, and apply it to handwritten digit string recognition (HDSR). This network is composed of three parts from bottom to top: feature extraction layers, feature dimension transposition layers and an output layer. Motivated by its super performance of DenseNet, we utilize dense blocks to conduct feature extraction. At the top of the network, a CTC (connectionist temporal classification) output layer is used to calculate the loss and decode the feature sequence, while some feature dimension transposition layers are applied to connect feature extraction and output layer.

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

Handwritten digits recognition is one of the challenging tasks and a lot of research is being carried out since many years. A remarkable work has been done for recognition of isolated handwritten characters as well as digits in many languages like English, Arabic, Devanagari, Chinese, Urdu and Pashto. However, the literature reviewed does not show any remarkable work done for Sindhi numerals recognition. The recognition of Sindhi digits is a difficult task due to the various writing styles and different font sizes. Therefore, SOM (Self-Organizing Map), a NN (Neural Network) method is used which can recognize digits with various writing styles and different font sizes. Only one sample is required to train the network for each pair of multi-digit numeral

2.3 PROBLEM STATEMENT DEFINITION

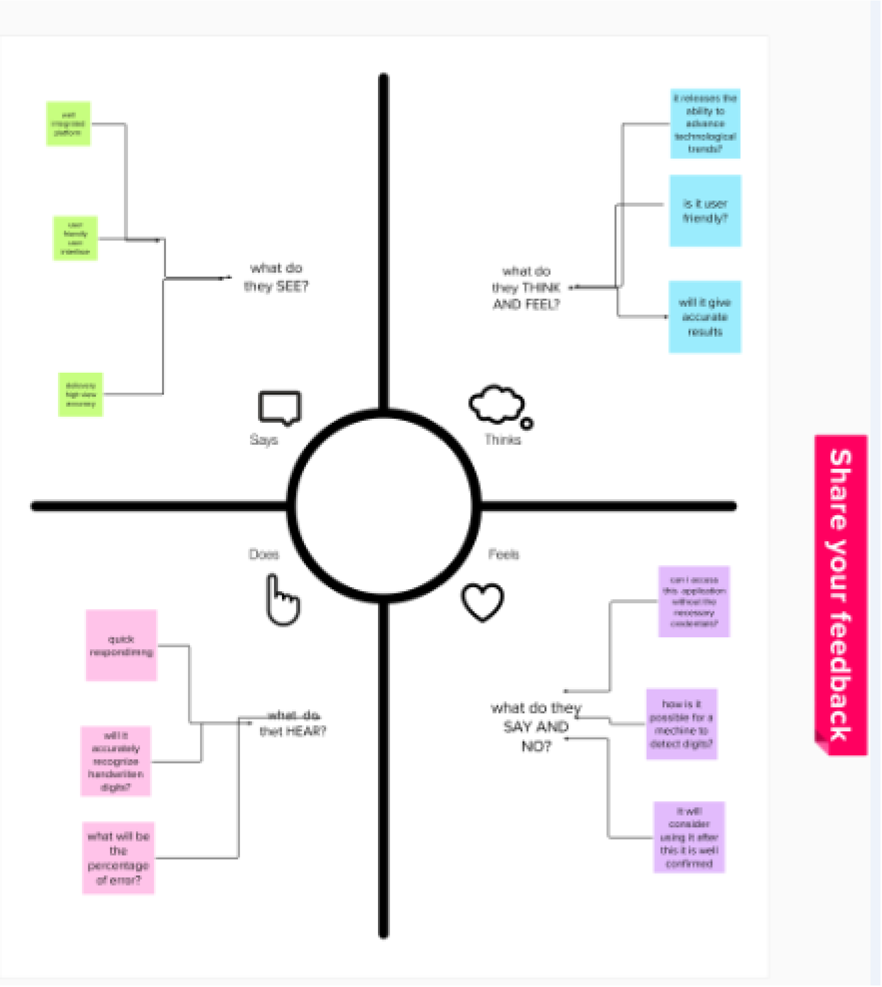
The problem statement is to classify handwritten digits. The goal is to take an image of a handwritten digit and determine what that digit and character is. It is easy for the human to perform a task accurately by practicing it repeatedly and memorizing it for the next time. Human brain can process and analyze images easily. Also, recognize the different elements present in the images.

The handwritten digit recognition is the capability of computer applications to recognize the human handwritten digits. It is a hard task for the machine because handwritten digits are not perfect and can be made with many different shapes and sizes. The handwritten digit recognition system is a way to tackle this problem which uses the image of a digit and recognizes the digit present in the image. Convolutional Neural Network model created using Python library over the MNIST dataset to recognize handwritten digits. Handwriting number recognition is a challenging problem researchers had been research into this area for so long especially in the recent years.

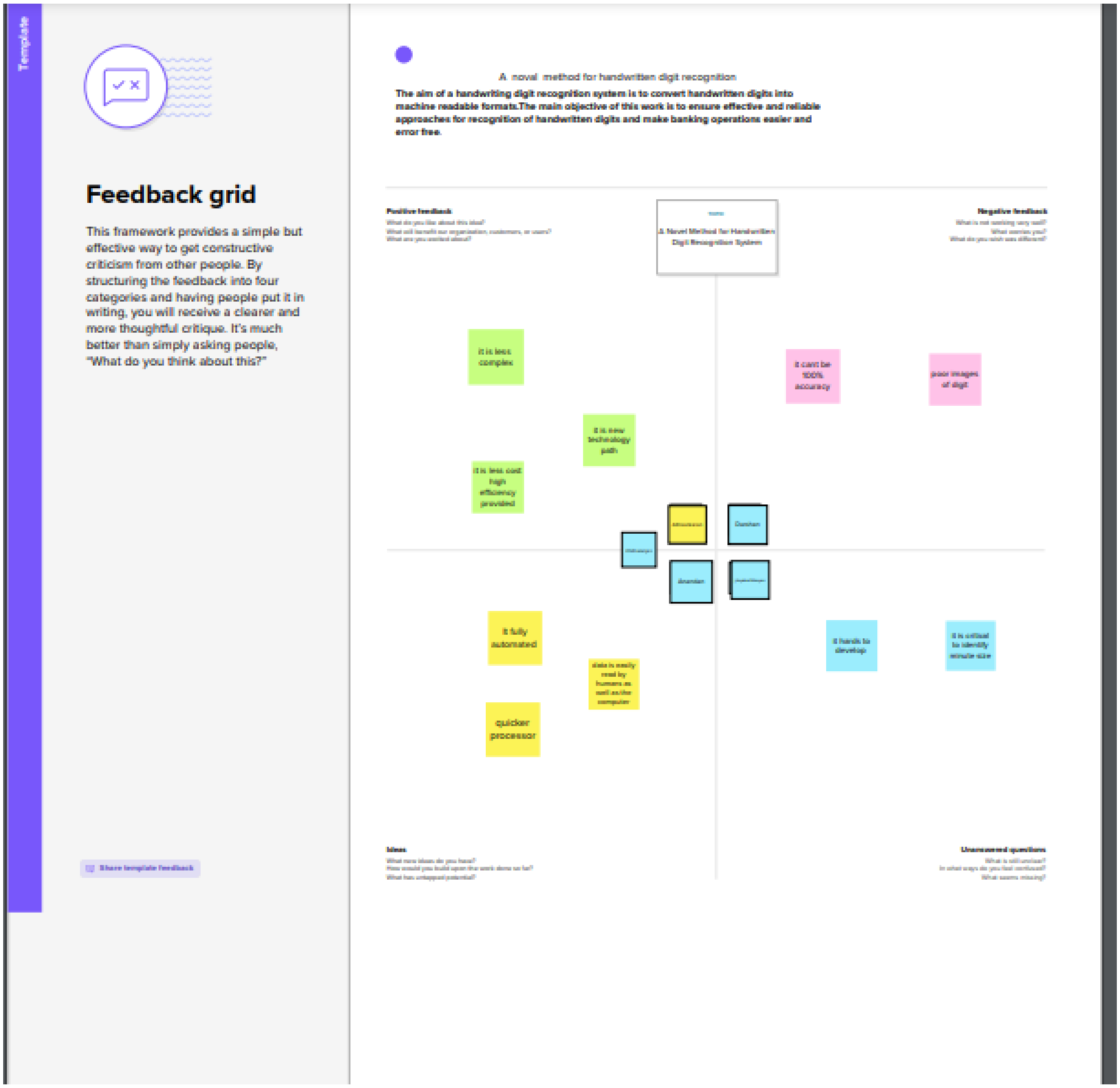
CHAPTER 3

IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS


well integrated platform
user friendly user interface
delivers high view accuracy
what do they SEE?
it releases the ability to advance technological trends?
is it user friendly?
will it give accurate results
what do they THINK AND FEEL?
Says
Thinks
Does
Feels
quick respondiming
will it accurately recognize handwritten digits?
what will be the percentage of error?
what do thet HEAR?
can i access this application without the necessary credentials?
how is it possible for a machine to detect digits?
it will consider using it after this it is well confirmed
what do they SAY AND NO?
Share your feedback

3.2 IDEATION & BRAINSTORMING



Template

Feedback grid

This framework provides a simple but effective way to get constructive criticism from other people. By structuring the feedback into four categories and having people put it in writing, you will receive a clearer and more thoughtful critique. It's much better than simply asking people, "What do you think about this?"

Share template feedback

A noval method for handwritten digit recognition

The aim of a handwriting digit recognition system is to convert handwritten digits into machine readable formats.The main objective of this work is to ensure effective and reliable approaches for recognition of handwritten digits and make banking operations easier and error free.

Positive feedback
What do you like about this idea?
What will benefit our organization, customers, or users?
What are you excited about?

A Novel Method for Handwritten Digit Recognition System

Negative feedback
What is not working very well?
What worries you?
What do you wish was different?

it is less complex

it is new technology path

it is less cost high efficiency provided

it cant be 100% accuracy

poor images of digit

it fully automated

data is easily read by humans as well as the computer

quicker processor

it hards to develop

it is critical to identify minute size

Ideas
What new ideas do you have?
How would you build upon the work done so far?
What has untapped potential?

Unanswered questions
What is still unclear?
In what ways do you feel confused?
What seems missing?

3.3 PROPOSED SOLUTION:

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	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
2.	Idea / Solution description	MNIST dats set is widely used for this recognition process and it has 70000 handwritten digits.we use Artificial neural networks to train these image sand 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 mod
3.	Novelty / Uniqueness	we can also check the accuracy of the processed data.
4.	Social Impact / Customer Satisfaction	Handwriting recognition has a variety of applications as electronic form filling, signature verification, automated music symbol notation reader, handwritten ancient document reading, write and send SMS in mother tongue and alternative to realistic keyboards
5.	Business Model (Revenue Model)	we already have many existing keyboard app Adding new feature like gesture so that it can be identified by handwritten recognition. this could increase revenue of the developers company.
6.	Scalability of the Solution	we can also create an extension for handwritten recognition that would be helpful for the user

3.4 PROBLEM SOLUTION FIT:

1.) The handwritten are not always of the same size ,width ,orientation and justified to margins as they differ from writing of person to person

2.) The similarity between digits such as 1 and 7,5 and 6,3 and 8,2 and 7 etc so, classifying between these numbers is also a major problem for computers

3.) The uniqueness and variety in the handwriting of different individuals also influence the formation and appearance of the digits

4.) A new people doesn't know language how to communicate?

5.) It using a encryption method strategy that accuracy also some problem made?

1. Also the underlying problems of not having the same size ,width, orientation, and margin always has been taken care of with the help of computer vision's opencv library's functionalities.

2. The problem of difficulty in distinguishing the difference between digits such as 1 and 7, 5 and 6, 3 and 8 etc has been resolved to a great extent with the opencv's edge detection and contour features

3. Also problems of dim lighting and blurry or unclear edges in images are corrected with the help of Gaussian blur technique. Now users can find their handwritten digits in one go without much complications

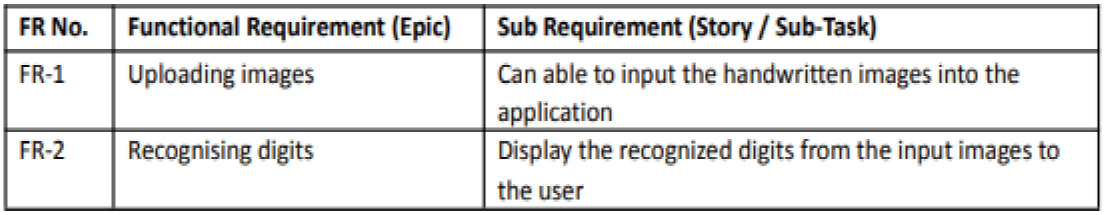
4. Using this technology easily identified what word be found out new people or another country man also easily using this technology identified what they phrase identified

5. Using this handwritten technology most of website using captcha technology so we hardly know some word so high so using method for encryption strategy

CHAPTER 4

REQUIREMENT ANALYSIS

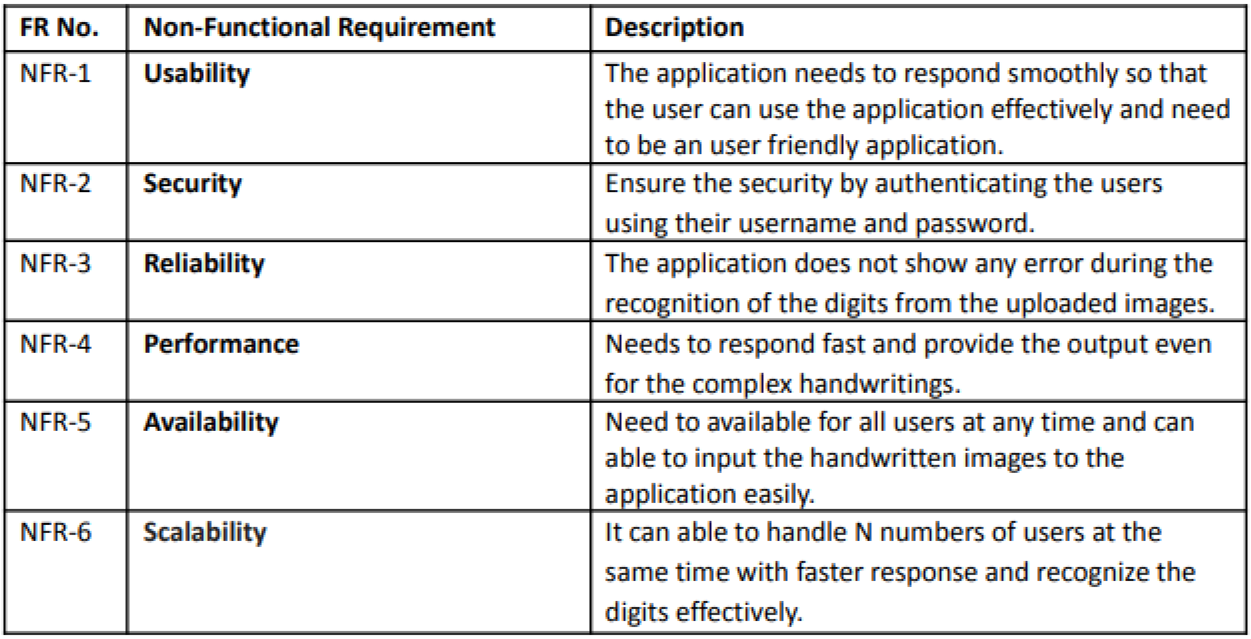
4.1 FUNCTIONAL REQUIREMENTS



FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Uploading images	Can able to input the handwritten images into the application
FR-2	Recognising digits	Display the recognized digits from the input images to the user

.

4.2 NON-FUNCTIONAL REQUIREMENTS



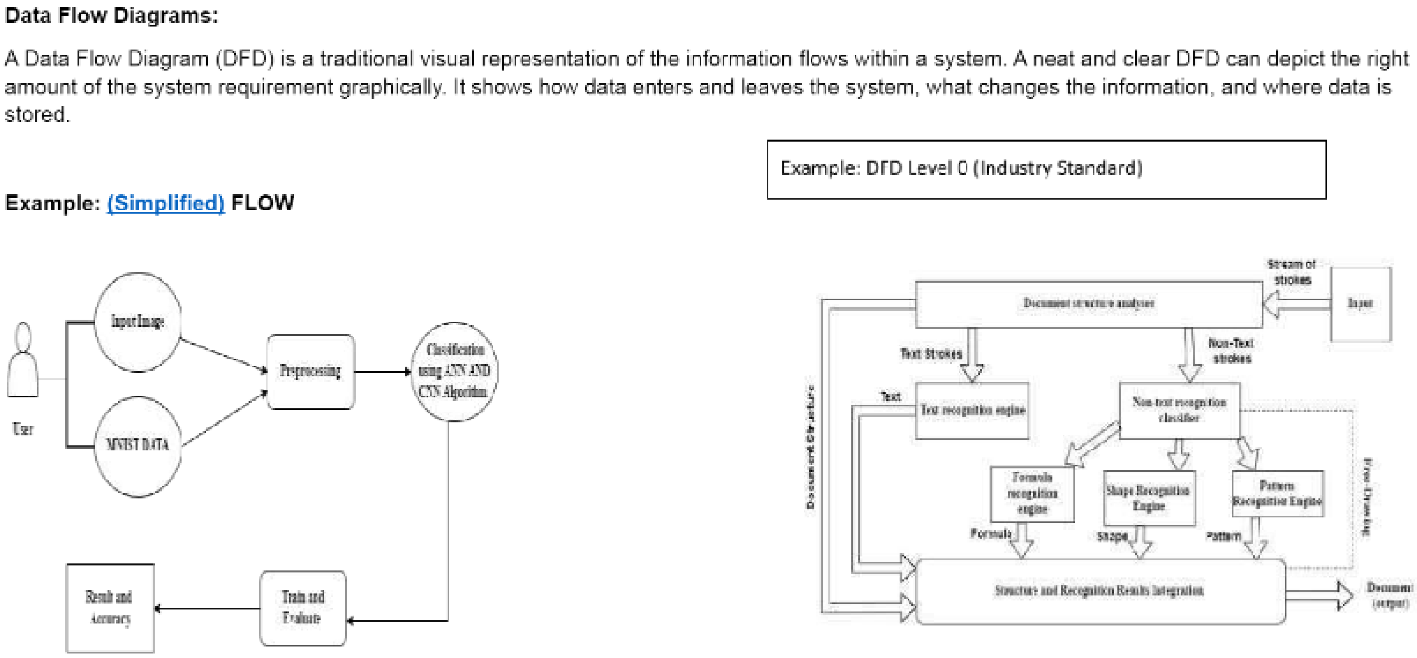
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The application needs to respond smoothly so that the user can use the application effectively and need to be an user friendly application.
NFR-2	**Security**	Ensure the security by authenticating the users using their username and password.
NFR-3	**Reliability**	The application does not show any error during the recognition of the digits from the uploaded images.
NFR-4	**Performance**	Needs to respond fast and provide the output even for the complex handwritings.
NFR-5	**Availability**	Need to available for all users at any time and can able to input the handwritten images to the application easily.
NFR-6	**Scalability**	It can able to handle N numbers of users at the same time with faster response and recognize the digits effectively.

CHAPTER 5

PROJECT DESIGN

5.1 DATA FLOW DIAGRAM

.



Data Flow Diagrams:

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.

Example: DFD Level 0 (Industry Standard)

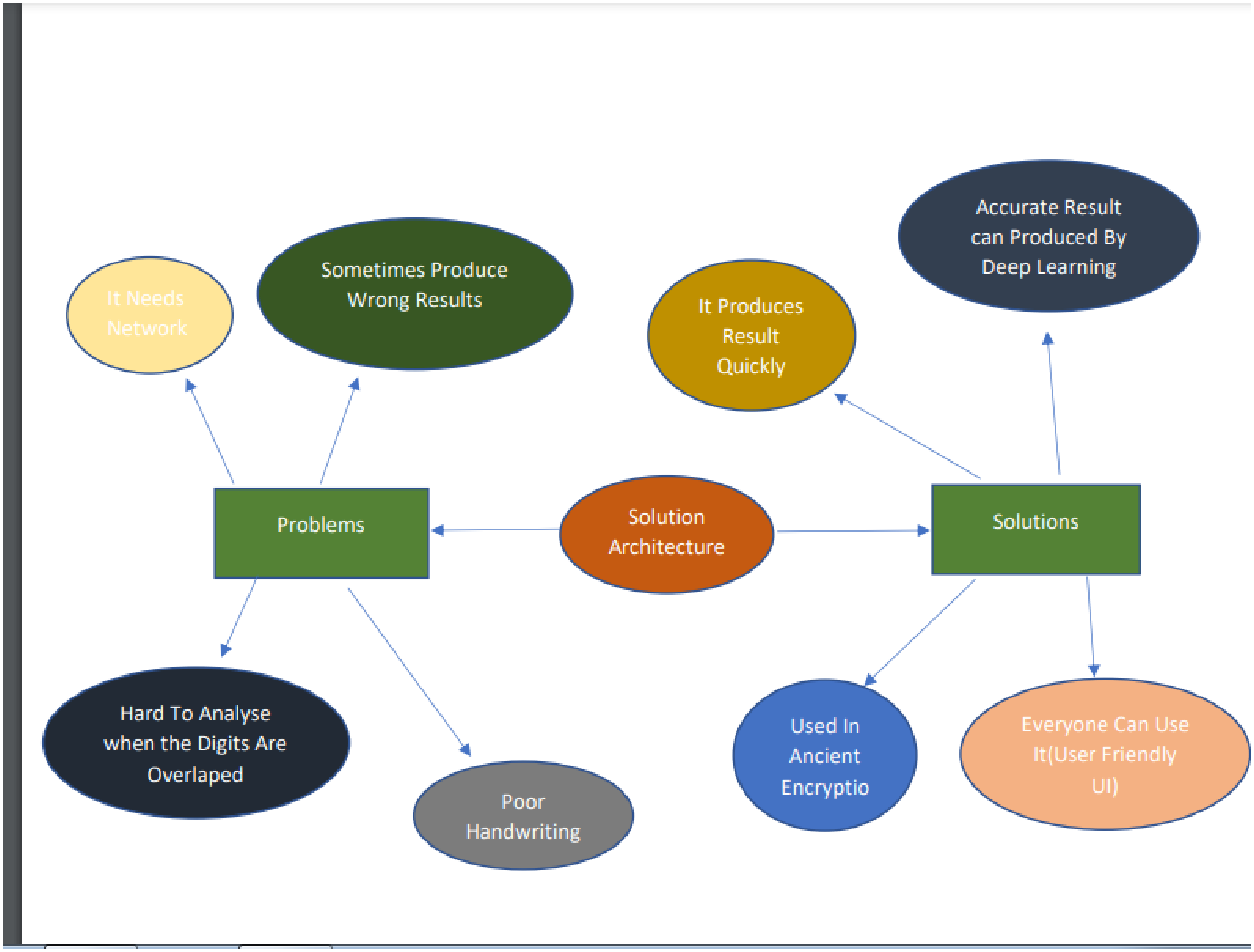
Example: (Simplified) FLOW

User Stories

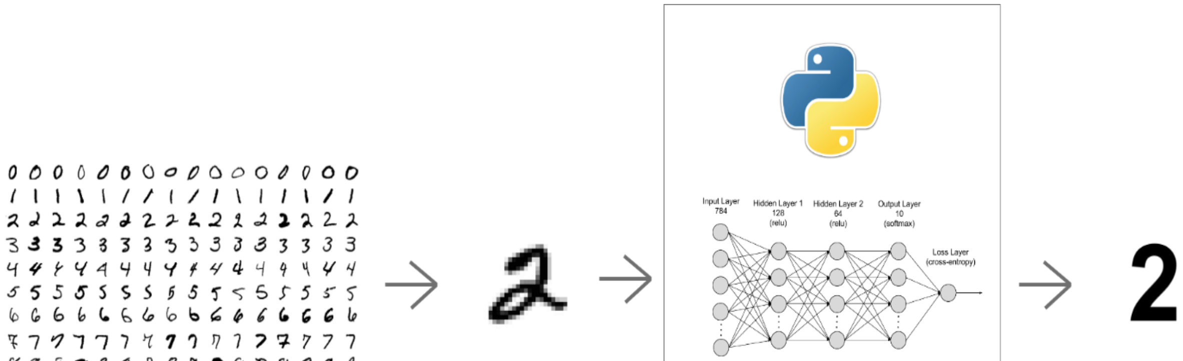
Use the below template to list all the user stories for the product.

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 view the guide and awareness to use this application.	I can view the awareness to use this application and its limitations.	Low	Sprint-1
		USN-2	As a user, I'm allowed to view the guided video to use the interface of this application.	I can gain knowledge to use this application by a practical method.	Low	Sprint-1
		USN-3	As a user, I can read the instructions to use this application.	I can read instructions also to use it in a user-friendly method.	Low	Sprint-2
	Recognize	USN-4	As a user, In this prediction page I get to choose the image.	I can choose the image 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 upload and choose the image from the system storage and also in any virtual storage.	Medium	Sprint-3
		USN-7	As a user, I will train and test the input to get the maximum accuracy of output.	I can able to train and test the application until it gets maximum accuracy of the result.	High	Sprint-4
		USN-8	As a user, I can access the MNIST data set	I can access the MNIST data set to produce the accurate result.	Medium	Sprint-3
Customer (Web user)	Home	USN-9	As a user, I can view the guide to use the web app.	I can view the awareness of this application and its limitations.	Low	Sprint-1

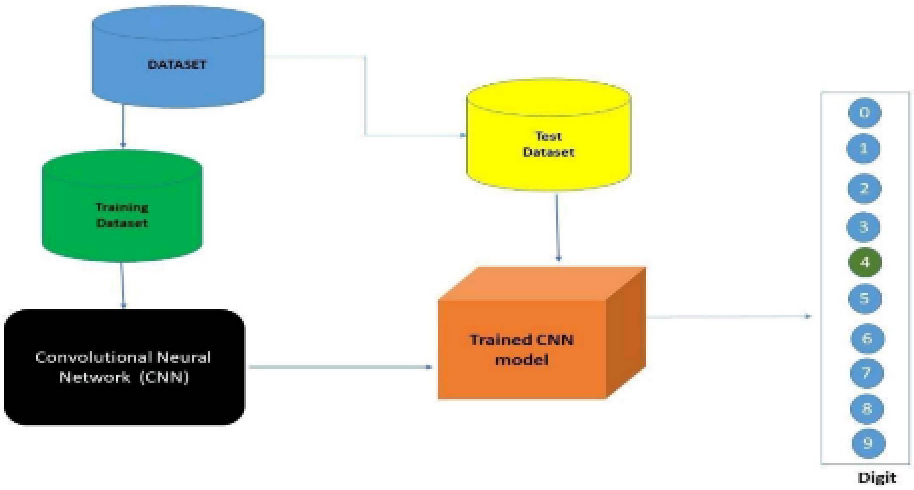
5.2 SOLUTION & TECHNICAL ARCHITECTURE


It Needs Network
Sometimes Produce Wrong Results
It Produces Result Quickly
Accurate Result can Produced By Deep Learning
Problems
Solution Architecture
Solutions
Hard To Analyse when the Digits Are Overlaped
Poor Handwriting
Used In Ancient Encryptio
Everyone Can Use It(User Friendly UI)

MNIST DATASET PROCESSING WITH PYTHON



ANACONDA
SLOGAN GOES HERE



5.3 COMPONENTS & TECHNOLOGIES:

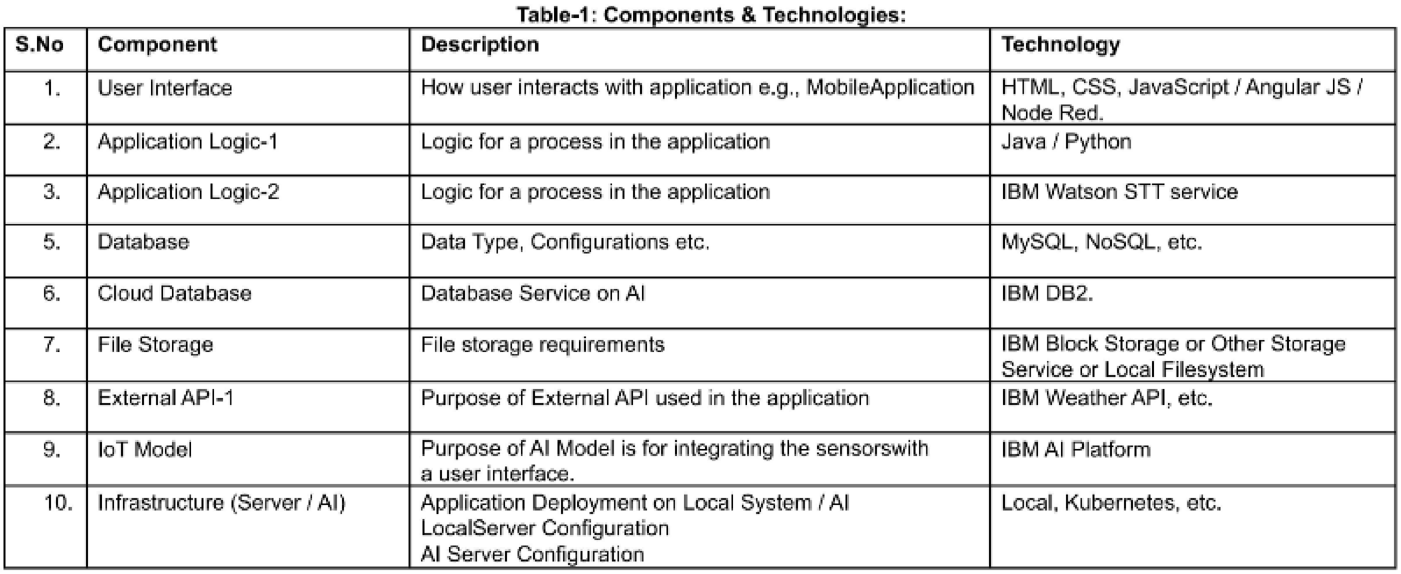


Table-1: Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g., MobileApplication	HTML, CSS, JavaScript / Angular JS / Node Red.
2.	Application Logic-1	Logic for a process in the application	Java / Python
3.	Application Logic-2	Logic for a process in the application	IBM Watson STT service
5.	Database	Data Type, Configurations etc.	MySQL, NoSQL, etc.
6.	Cloud Database	Database Service on AI	IBM DB2.
7.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem
8.	External API-1	Purpose of External API used in the application	IBM Weather API, etc.
9.	IoT Model	Purpose of AI Model is for integrating the sensorswith a user interface.	IBM AI Platform
10.	Infrastructure (Server / AI)	Application Deployment on Local System / AI LocalServer Configuration AI Server Configuration	Local, Kubernetes, etc.

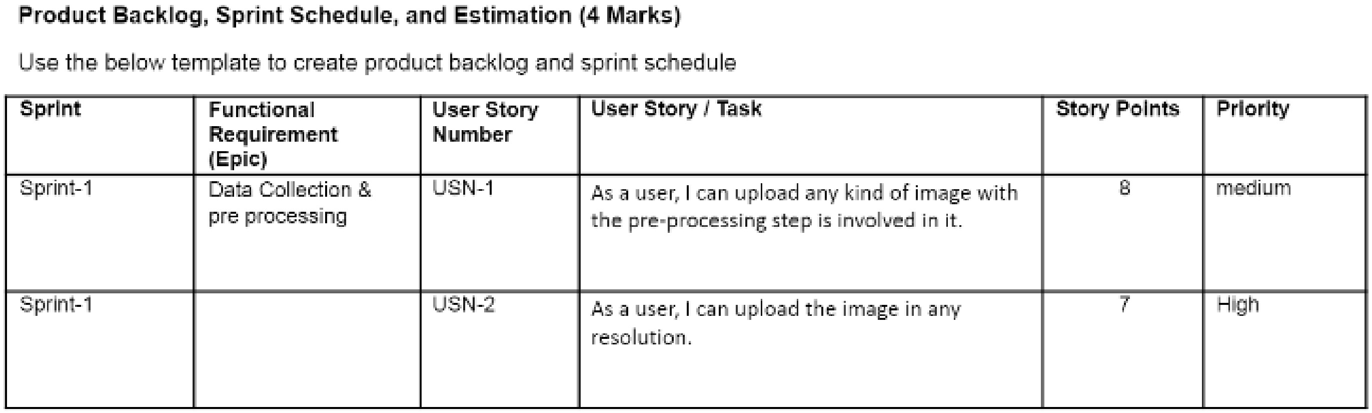
5.4 USER STORIES

User Type	Functional Requirement	User Story Number	User Story / Task	Acceptance Criteria	Priority	Release
Customer	Building the application	USN-1	As a user, I should be able to access the application from anywhere and use on any devices	User can access the application using the browser on any device	High	Sprint-4
	Uploading Image	USN-2	As a user, I should be able to upload images to predict the digits	User can upload images	low	Sprint-2
	Viewing the Results	USN-3	As a user, I should be able to view the results	The result of the prediction is displayed	High	Sprint-3
	Viewing Other Prediction	USN-4	As a user, I should be able to see other close predictions	The accuracy of other values must be displayed	low	Sprint-1
	Usage Instruction	USN-5	As a user, I should have a usage instruction to know how to use the application	The usage instruction is displayed on the home page	high	Sprint-3

CHAPTER 6

PROJECT PLANNING AND SCHEDULING

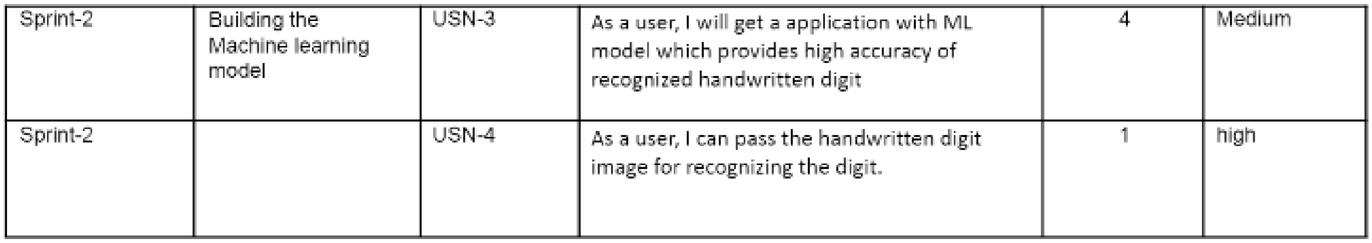
6.1 SPRINT PLANNING AND ESTIMATION



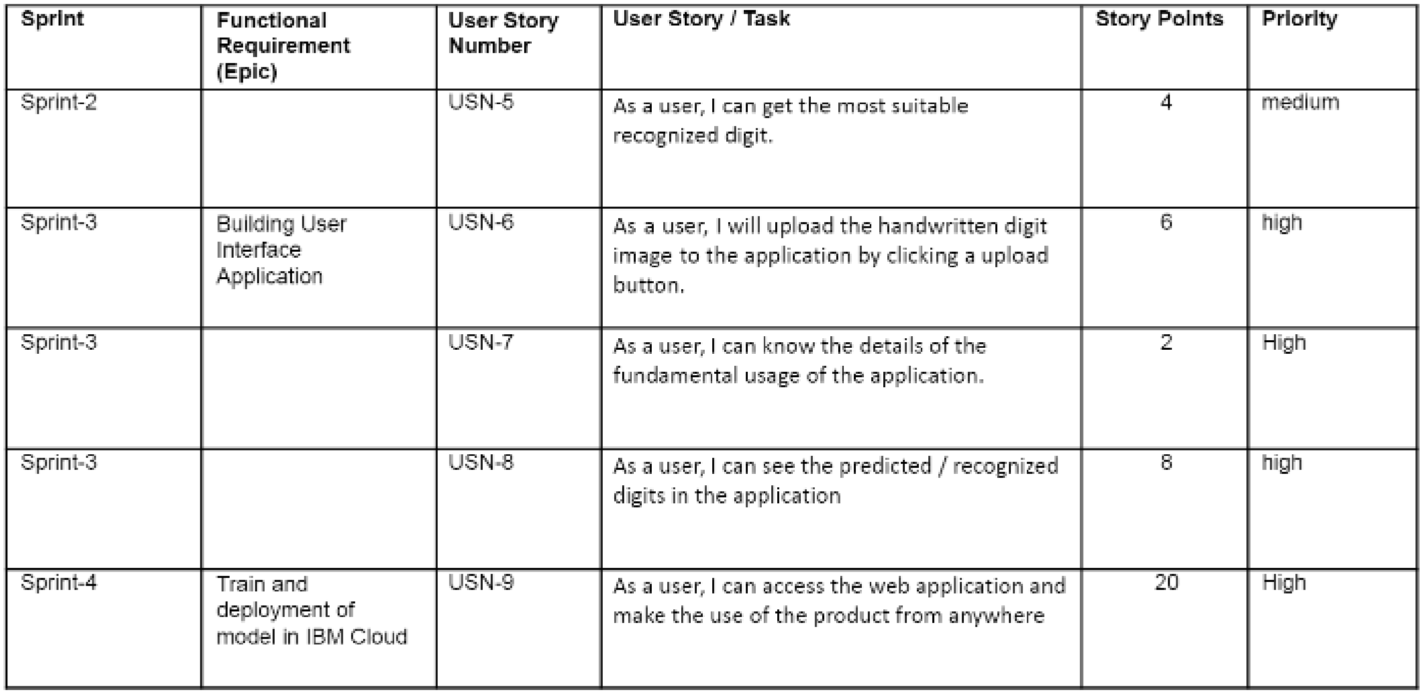
Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority
Sprint-1	Data Collection & pre processing	USN-1	As a user, I can upload any kind of image with the pre-processing step is involved in it.	8	medium
Sprint-1		USN-2	As a user, I can upload the image in any resolution.	7	High

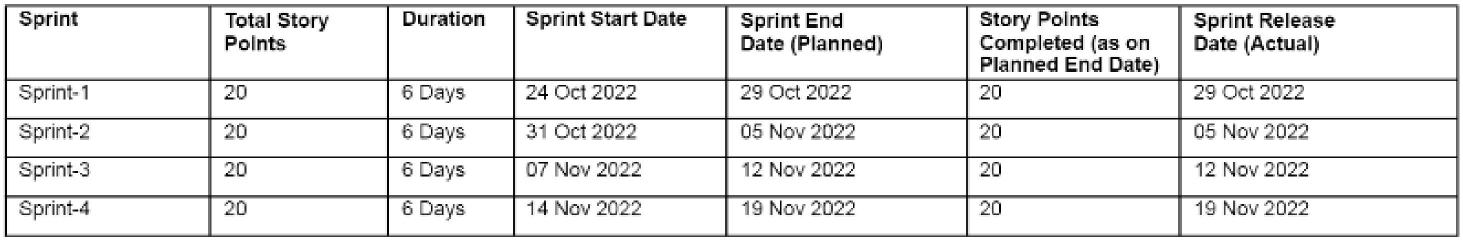


Sprint-2	Building the Machine learning model	USN-3	As a user, I will get a application with ML model which provides high accuracy of recognized handwritten digit	4	Medium
Sprint-2		USN-4	As a user, I can pass the handwritten digit image for recognizing the digit.	1	high



Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority
Sprint-2		USN-5	As a user, I can get the most suitable recognized digit.	4	medium
Sprint-3	Building User Interface Application	USN-6	As a user, I will upload the handwritten digit image to the application by clicking a upload button.	6	high
Sprint-3		USN-7	As a user, I can know the details of the fundamental usage of the application.	2	High
Sprint-3		USN-8	As a user, I can see the predicted / recognized digits in the application	8	high
Sprint-4	Train and deployment of model in IBM Cloud	USN-9	As a user, I can access the web application and make the use of the product from anywhere	20	High

6.2 SPRINT DELIVERY SCHEDULE



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

6.3 REPORT FROM JIRA

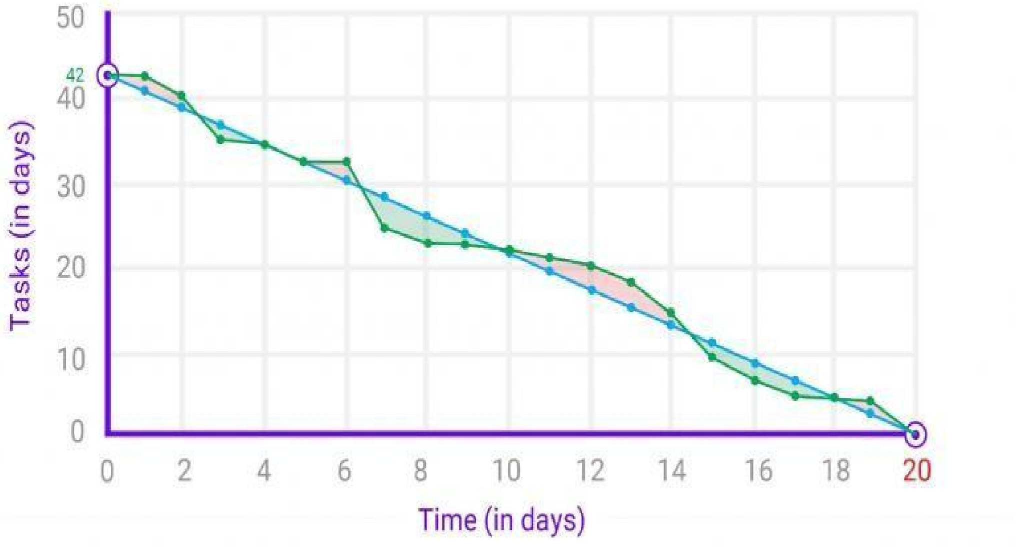
Velocity:

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

Average Velocity = 20 / 6 = 3.33

Burndown Chart:

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.

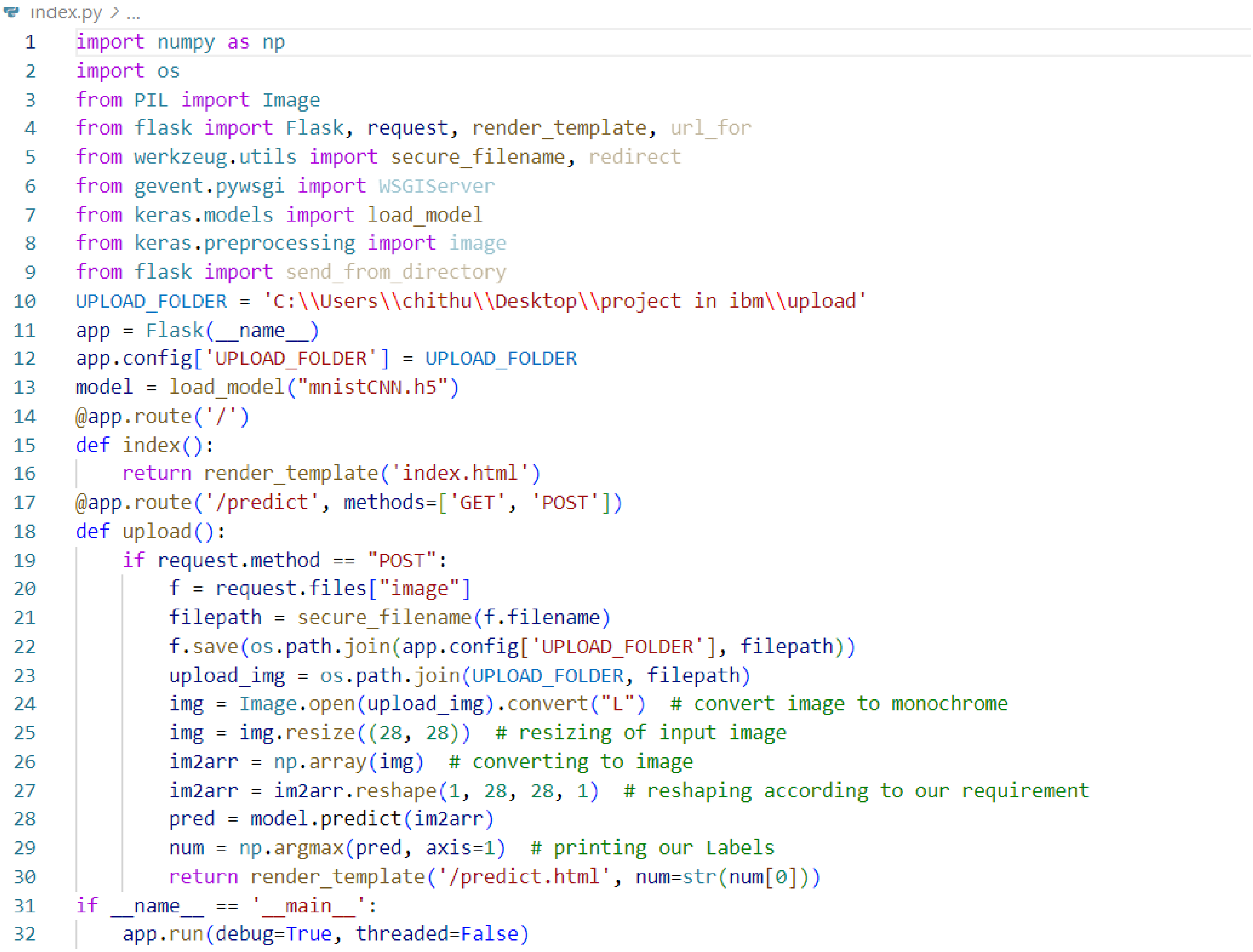


CHAPTER 7

CODING & SOLUTION

7.1 FEATURE 1 – FLASK FILE UPLOADING

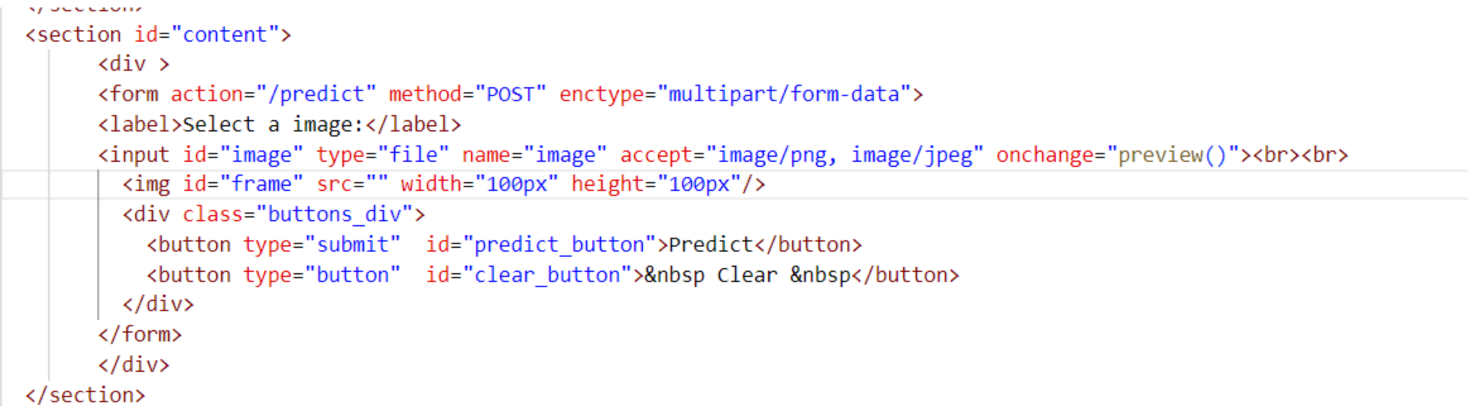
Handling file upload in Flask is very easy. It needs an HTML form with its enctype attribute set to ‘multipart/form-data’, posting the file to a URL. The URL handler fetches file from request.files[] object and saves it to the upload folder.



```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 = 'C:\\Users\\chithu\\Desktop\\project in ibm\\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")  # convert image to monochrome
        img = img.resize((28, 28))  # resizing of input image
        im2arr = np.array(img)  # converting to image
        im2arr = im2arr.reshape(1, 28, 28, 1)  # reshaping according to our requirement
        pred = model.predict(im2arr)
        num = np.argmax(pred, axis=1)  # printing our Labels
        return render_template('/predict.html', num=str(num[0]))
if __name__ == '__main__':
    app.run(debug=True, threaded=False)
```

7.2 FEATURE 2 – UPLOAD IMAGE WITH PREVIEW

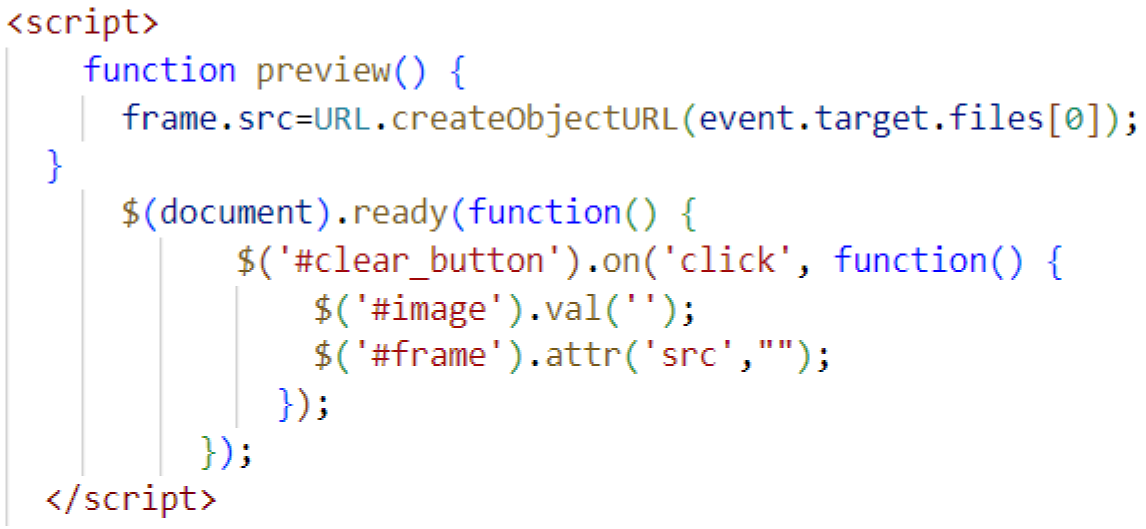
A preview refers to a feature that lets you glimpse or view something in part or whole without it being opened. A picture preview would show a small version of the picture and give you a good idea what each picture is without opening each picture it is a useful feature created using JavaScript.



```
</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>
```

7.3 FEATURE 3 – CLEAR IMAGE

This feature can be used to clear the image if we uploaded a wrong image or if we need to change the image. The clear button clears both the image value and the preview of the image in script tag.



```
<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>
```

CHAPTER 8

TESTING

8.1 TEST CASES

Test case ID	Feature Type	Component	Test Scenario	Expected 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	FAIL
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 2560 x 1801 and 768 x 630	FAIL
HP_TC_003	Functional	Home Page	Check if user can upload their file	The input image should be uploaded to the application successfully	Working as expected	PASS
HP_TC_004	Functional	Home Page	Check if user cannot upload unsupported files	The application should not allow user to select a non image file	User is able to upload any file	FAIL
HP_TC_005	Functional	Home Page	Check if the page redirects to the result page once the input is given	The page should redirect to the results page	Working as expected	PASS

BE_TC_001	Functional	Backend	Check if all the routes are working properly	All the routes should properly work	Working as expected	PASS
M_TC_001	Functional	Model	Check if the model can handle various image sizes	The model should rescale the image and predict the results	Working as expected	PASS
M_TC_002	Functional	Model	Check if the model predicts the digit	The model should predict the number	Working as expected	PASS
M_TC_003	Functional	Model	Check if the model can handle complex input image	The model should predict the number in the complex image	The model fails to identify the digit since the model is not built to handle such data	FAIL
RP_TC_001	UI	Result Page	Verify UI elements in the Result Page	The Result page must be displayed properly	Working as expected	PASS
RP_TC_002	UI	Result Page	Check if the input image is displayed properly	The input image should be displayed properly	The size of the input image exceeds the display container	FAIL
RP_TC_003	UI	Result Page	Check if the result is displayed properly	The result should be displayed properly	Working as expected	PASS
RP_TC_004	UI	Result Page	Check if the other predictions are displayed properly	The other predictions should be displayed properly	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

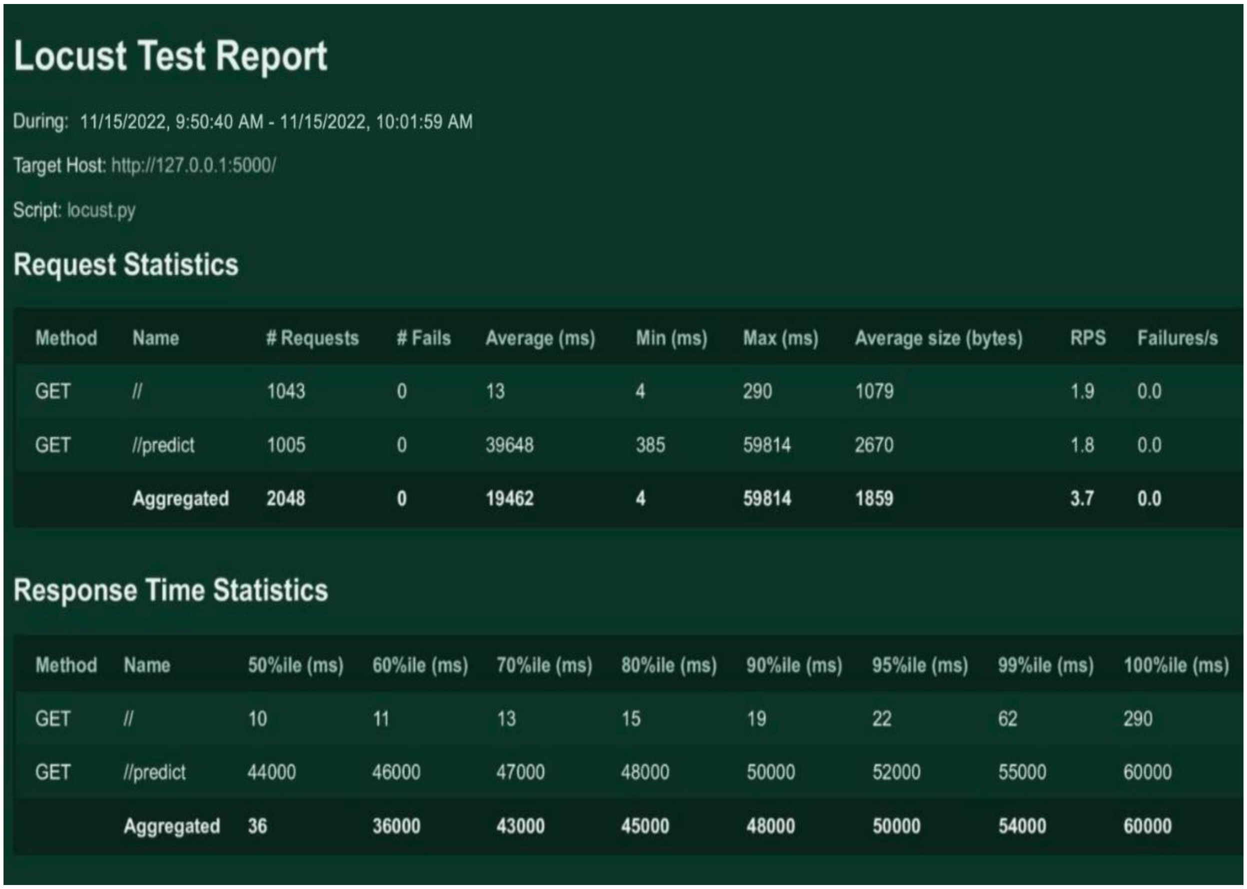
8.2.2 TEST CASE ANALYSIS

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



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/

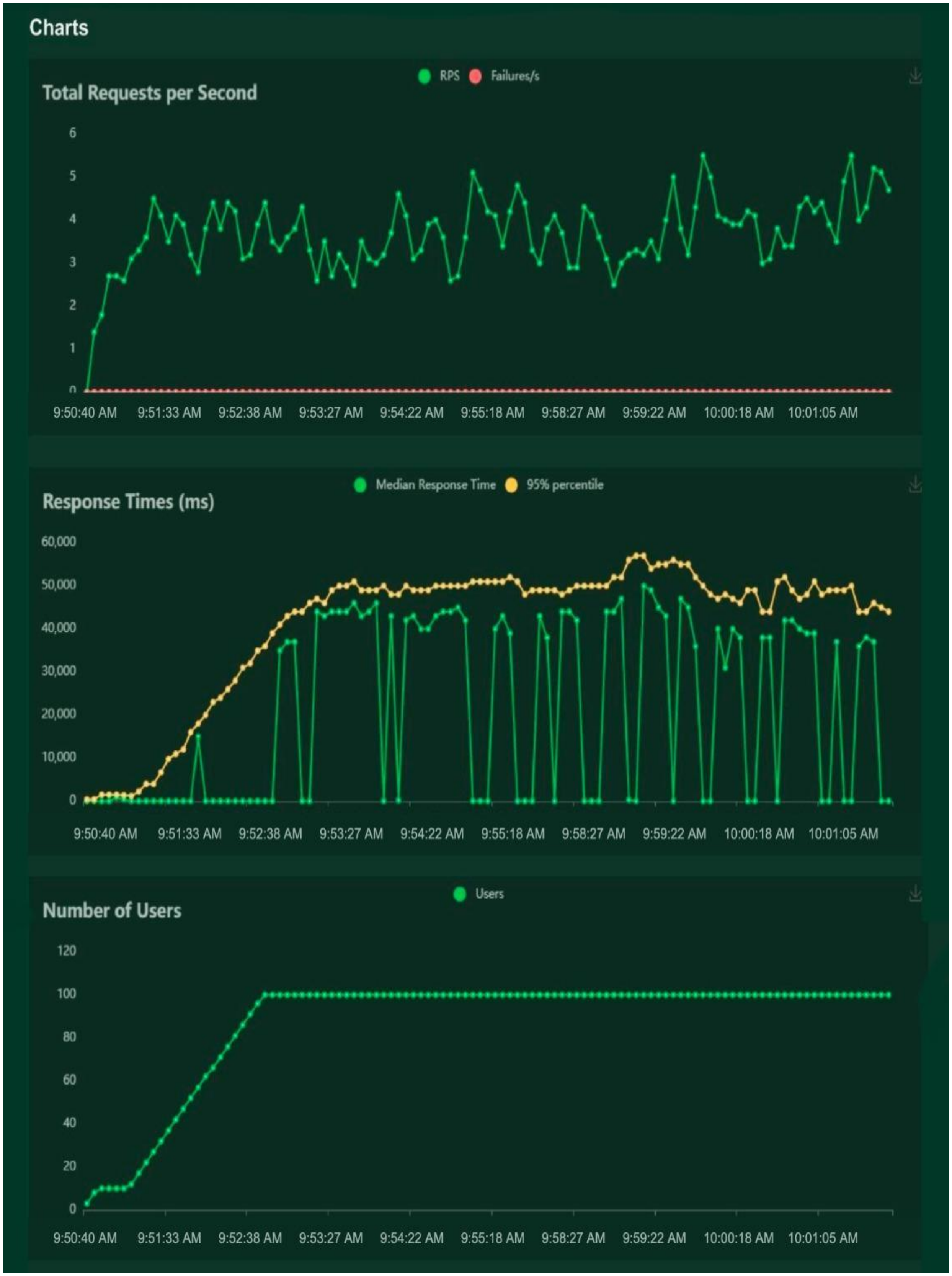
Script: locust.py

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
● RPS ● Failures/s
6
5
4
3
2
1
0
9:50:40 AM
9:51:33 AM
9:52:38 AM
9:53:27 AM
9:54:22 AM
9:55:18 AM
9:58:27 AM
9:59:22 AM
10:00:18 AM
10:01:05 AM
Response Times (ms)
● Median Response Time ● 95% percentile
60,000
50,000
40,000
30,000
20,000
10,000
0
9:50:40 AM
9:51:33 AM
9:52:38 AM
9:53:27 AM
9:54:22 AM
9:55:18 AM
9:58:27 AM
9:59:22 AM
10:00:18 AM
10:01:05 AM
Number of Users
● Users
120
100
80
60
40
20
0
9:50:40 AM
9:51:33 AM
9:52:38 AM
9:53:27 AM
9:54:22 AM
9:55:18 AM
9:58:27 AM
9:59:22 AM
10:00:18 AM
10:01:05 AM

CHAPTER 10

ADVANTAGES & DISADVANTAGES

ADVANTAGES

- Reduces manual work
- More accurate than average human
- Capable of handling a lot of data
- Can be used anywhere from any device
- Neural Network is used to train and identify written digits for greater efficiency.
- The accuracy rate is very high.
- Speed of data entry
- It is much easier to dictate the machine than to write
- Easier data retrieval

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
- There is a wide range of handwriting – good and bad.
- It is tricky for programmers to provide enough examples of how every character might look.
- Customers must try with clear image and neat handwriting to get accuracy in digits.
- Unclear image will not give accurate results.

CHAPTER 11
CONCLUSION

Convolutional Neural Network (CNN) adds its significant improvement to the Manuscript Document Recognition System. This paper tells us the effectiveness of CNN-based classification of data and pre-processing methods. Our model clearly sees handwriting and achieves outgoing predictions of up to 82.16% and accurate predictions of up to 69.16%. However the model can be continuously developed using multiple training samples. This will help the model to learn as well as the generalize better. There are many images in the training set that are completely invisible to the human eye.

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 number plates of vehicles, processing bank cheque amounts, numeric entries in forms filled up by hand (tax forms) and so on.

Through extensive evaluation using a MNIST dataset, the present work suggests the role of various hyper-parameters. Fine tuning of hyper-parameters is essential in improving the performance of CNN architecture. We achieved a recognition rate of 99.89% with the Adam optimizer for the MNIST database, which is better than all previously reported results. The effect of increasing the number of convolutional layers in CNN architecture on the performance of handwritten digit recognition is clearly presented through the experiments.

CHAPTER 12

FUTURE SCOPE

This project can be enhanced with a great field of machine learning and artificial intelligence. The world can think of a software which can recognize the text from a picture and can show it to the others, for example a shop name detector. Or this project can be extended to a greater concept of all the character sets in the world. This project has not gone for the total English alphabet because there will be more and many more training sets and testing values that the neural network model will not be enough to detect. 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.

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

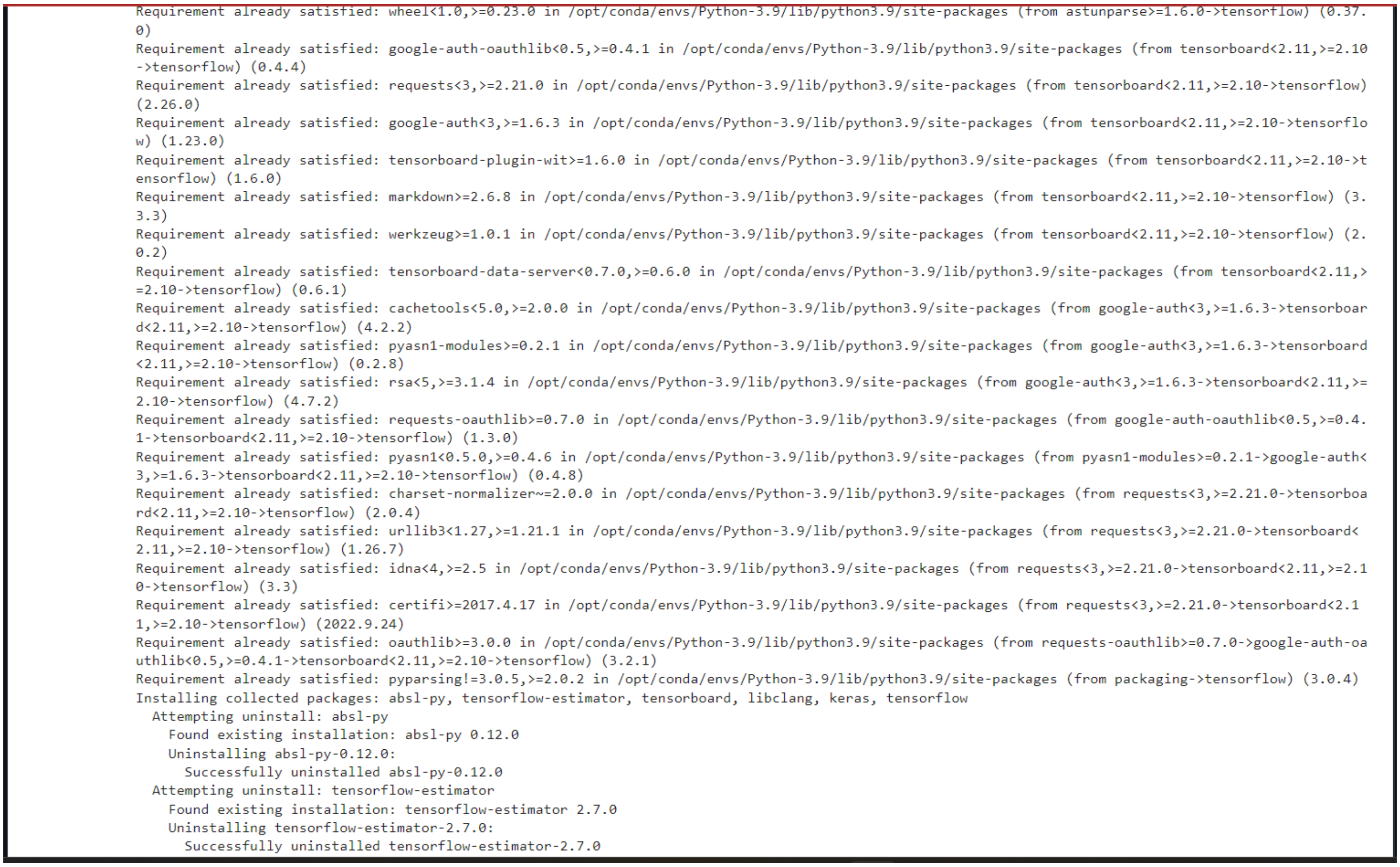
This project has endless potential and can always be enhanced to become better. Implementing this concept in the real world will benefit several industries and reduce the workload on many workers, enhancing overall work efficiency.

APPENDIX

SOURCE CODE

```
In [1]: !pip install tensorflow --upgrade
```

```
Requirement already satisfied: tensorflow in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (2.7.2)
Collecting tensorflow
  Downloading tensorflow-2.10.1-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (578.1 MB)
     |████████████████████████████████| 578.1 MB 53 kB/s s eta 0:00:012��█████████████▌     | 492.2 MB 30.9 MB/s eta 0:00:03��██████████████████████
██     | 508.8 MB 30.9 MB/s eta 0:00:03ta 0:00:02
Requirement already satisfied: astunparse>=1.6.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (1.6.3)
Requirement already satisfied: setuptools in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (58.0.4)
Collecting libclang>=13.0.0
  Downloading libclang-14.0.6-py2.py3-none-manylinux2010_x86_64.whl (14.1 MB)
     |████████████████████████████████| 14.1 MB 79.3 MB/s eta 0:00:01
Requirement already satisfied: wrapt>=1.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (1.12.1)
Requirement already satisfied: keras-preprocessing>=1.1.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (1.1.2)
Requirement already satisfied: six>=1.12.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (1.15.0)
Requirement already satisfied: numpy>=1.20 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (1.20.3)
Collecting keras<2.11,>=2.10.0
  Downloading keras-2.10.0-py2.py3-none-any.whl (1.7 MB)
     |████████████████████████████████| 1.7 MB 77.3 MB/s eta 0:00:01
Collecting tensorflow-estimator<2.11,>=2.10.0
  Downloading tensorflow_estimator-2.10.0-py2.py3-none-any.whl (438 kB)
     |████████████████████████████████| 438 kB 84.1 MB/s eta 0:00:01
Requirement already satisfied: termcolor>=1.1.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (1.1.0)
Collecting absl-py>=1.0.0
  Downloading absl_py-1.3.0-py3-none-any.whl (124 kB)
     |████████████████████████████████| 124 kB 73.3 MB/s eta 0:00:01
Requirement already satisfied: packaging in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (21.3)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (1.42.0)
Requirement already satisfied: google-pasta>=0.1.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (0.2.0)
Requirement already satisfied: flatbuffers>=2.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (2.0)
Requirement already satisfied: opt-einsum>=2.3.2 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (3.3.0)
Requirement already satisfied: protobuf<3.20,>=3.9.2 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (3.19.1)
Requirement already satisfied: gast<=0.4.0,>=0.2.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (0.4.0)
Requirement already satisfied: typing-extensions>=3.6.6 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (4.1.1)
Requirement already satisfied: h5py>=2.9.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (3.2.1)
Collecting tensorboard<2.11,>=2.10
  Downloading tensorboard-2.10.1-py3-none-any.whl (5.9 MB)
     |████████████████████████████████| 5.9 MB 83.6 MB/s eta 0:00:01
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (0.23.
1)
Requirement already satisfied: wheel<1.0,>=0.23.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from astunparse>=1.6.0->tensorflow) (0.37.
```



```
Requirement already satisfied: wheel<1.0,>=0.23.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from astunparse>=1.6.0->tensorflow) (0.37.
0)
Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorboard<2.11,>=2.10
->tensorflow) (0.4.4)
Requirement already satisfied: requests<3,>=2.21.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorboard<2.11,>=2.10->tensorflow)
(2.26.0)
Requirement already satisfied: google-auth<3,>=1.6.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorboard<2.11,>=2.10->tensorflo
w) (1.23.0)
Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorboard<2.11,>=2.10->t
ensorflow) (1.6.0)
Requirement already satisfied: markdown>=2.6.8 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorboard<2.11,>=2.10->tensorflow) (3.
3.3)
Requirement already satisfied: werkzeug>=1.0.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorboard<2.11,>=2.10->tensorflow) (2.
0.2)
Requirement already satisfied: tensorboard-data-server<0.7.0,>=0.6.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorboard<2.11,>
=2.10->tensorflow) (0.6.1)
Requirement already satisfied: cachetools<5.0,>=2.0.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from google-auth<3,>=1.6.3->tensorboar
d<2.11,>=2.10->tensorflow) (4.2.2)
Requirement already satisfied: pyasn1-modules>=0.2.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from google-auth<3,>=1.6.3->tensorboard
<2.11,>=2.10->tensorflow) (0.2.8)
Requirement already satisfied: rsa<5,>=3.1.4 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from google-auth<3,>=1.6.3->tensorboard<2.11,>=
2.10->tensorflow) (4.7.2)
Requirement already satisfied: requests-oauthlib>=0.7.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from google-auth-oauthlib<0.5,>=0.4.
1->tensorboard<2.11,>=2.10->tensorflow) (1.3.0)
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pyasn1-modules>=0.2.1->google-auth<
3,>=1.6.3->tensorboard<2.11,>=2.10->tensorflow) (0.4.8)
Requirement already satisfied: charset-normalizer~=2.0.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests<3,>=2.21.0->tensorboa
rd<2.11,>=2.10->tensorflow) (2.0.4)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests<3,>=2.21.0->tensorboard<
2.11,>=2.10->tensorflow) (1.26.7)
Requirement already satisfied: idna<4,>=2.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests<3,>=2.21.0->tensorboard<2.11,>=2.1
0->tensorflow) (3.3)
Requirement already satisfied: certifi>=2017.4.17 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests<3,>=2.21.0->tensorboard<2.1
1,>=2.10->tensorflow) (2022.9.24)
Requirement already satisfied: oauthlib>=3.0.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests-oauthlib>=0.7.0->google-auth-oa
uthlib<0.5,>=0.4.1->tensorboard<2.11,>=2.10->tensorflow) (3.2.1)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from packaging->tensorflow) (3.0.4)
Installing collected packages: absl-py, tensorflow-estimator, tensorboard, libclang, keras, tensorflow
  Attempting uninstall: absl-py
    Found existing installation: absl-py 0.12.0
    Uninstalling absl-py-0.12.0:
      Successfully uninstalled absl-py-0.12.0
  Attempting uninstall: tensorflow-estimator
    Found existing installation: tensorflow-estimator 2.7.0
    Uninstalling tensorflow-estimator-2.7.0:
      Successfully uninstalled tensorflow-estimator-2.7.0
```

```
    Uninstalling tensorboard-2.7.0:
      Successfully uninstalled tensorboard-2.7.0
  Attempting uninstall: keras
    Found existing installation: keras 2.7.0
    Uninstalling keras-2.7.0:
      Successfully uninstalled keras-2.7.0
  Attempting uninstall: tensorflow
    Found existing installation: tensorflow 2.7.2
    Uninstalling tensorflow-2.7.2:
      Successfully uninstalled tensorflow-2.7.2
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the source of the followin
g dependency conflicts.
tensorflow-text 2.7.3 requires tensorflow<2.8,>=2.7.0, but you have tensorflow 2.10.1 which is incompatible.
tensorflow-metadata 1.5.0 requires absl-py<0.13,>=0.9, but you have absl-py 1.3.0 which is incompatible.
autoai-ts-libs 1.1.9 requires tensorflow<2.8,>=2.7.0; python_version >= "3.9", but you have tensorflow 2.10.1 which is incompatible.
Successfully installed absl-py-1.3.0 keras-2.10.0 libclang-14.0.6 tensorboard-2.10.1 tensorflow-2.10.1 tensorflow-estimator-2.10.0
```

In [3]:

```
import numpy as np
import tensorflow
from tensorflow.keras.datasets import mnist
from tensorflow.keras.models import Sequential
from tensorflow.keras import layers
from tensorflow.keras.layers import Dense, Flatten
from tensorflow.keras.layers import Conv2D
from keras.utils import np_utils
import matplotlib.pyplot as plt
```

In [4]:

```
(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 [5]:

```
print (x_train.shape)
print (x_test.shape)
```

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

In [6]:

```
x_train[0]
```

In [6]:

```
x_train[0]
```

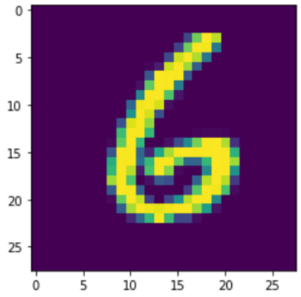
Out[6]:

```
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,
          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)
```

In [7]:

```
plt.imshow(x_train[6000])
```

Out[7]:



In [8]:

```
np.argmax(y_train[6000])
```

Out[8]:

```
0
```

In [9]:

```
number_of_classes = 10
```

In [10]:

```
y_train = np_utils.to_categorical (y_train, number_of_classes)
y_test = np_utils.to_categorical (y_test, number_of_classes)
```

In [12]:

```
model=Sequential()
```

In [13]:

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

In [14]:

```
model.add(Flatten())
```

In [15]:

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

In [16]:

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

In [17]:

```
x_train = np.asarray(x_train)
y_train = np.asarray(y_train)
```

In [19]:

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

```
Epoch 1/5
1875/1875 [==============================] - 107s 57ms/step - loss: 0.0226 - accuracy: 0.9926 - val_loss: 0.1176 - val_accuracy: 0.9770
Epoch 2/5
1875/1875 [==============================] - 108s 58ms/step - loss: 0.0196 - accuracy: 0.9937 - val_loss: 0.1708 - val_accuracy: 0.9749
Epoch 3/5
1875/1875 [==============================] - 106s 56ms/step - loss: 0.0193 - accuracy: 0.9945 - val_loss: 0.1352 - val_accuracy: 0.9755
Epoch 4/5
1875/1875 [==============================] - 107s 57ms/step - loss: 0.0160 - accuracy: 0.9952 - val_loss: 0.1255 - val_accuracy: 0.9778
Epoch 5/5
1875/1875 [==============================] - 106s 57ms/step - loss: 0.0144 - accuracy: 0.9959 - val_loss: 0.2097 - val_accuracy: 0.9754
```

Out[19]:

In [20]:

```
metrics = model.evaluate(x_test, y_test, verbose=0)
print("Metrics (Test loss &Test Accuracy) : ")
print(metrics)
```

```
Metrics (Test loss &Test Accuracy) :
[0.20970351994037628, 0.9753999710083008]
```

In [21]:

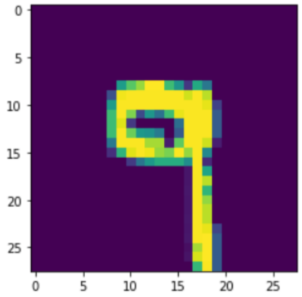
```
prediction=model.predict(x_test[6000:6001])
print(prediction)
```

```
1/1 [==============================] - 0s 82ms/step
[[0.0000000e+00 0.0000000e+00 7.8181486e-34 1.7299951e-15 4.5163800e-20
  3.4631572e-21 3.4006425e-38 2.6362448e-17 3.2771523e-20 1.0000000e+00]]
```

In [22]:

```
plt.imshow(x_test[6000])
```

Out[22]:



In [23]:

```
import numpy as np
print(np.argmax(prediction, axis=1))
```

```
[9]
```

In [24]:

```
np.argmax(y_test[6000:6001])
```

Out[24]:

```
9
```

In [25]:

```
model.save('models/mnistCNN.h5')
```

In [26]:

```
cd models
```

```
/home/wsuser/work/models
```

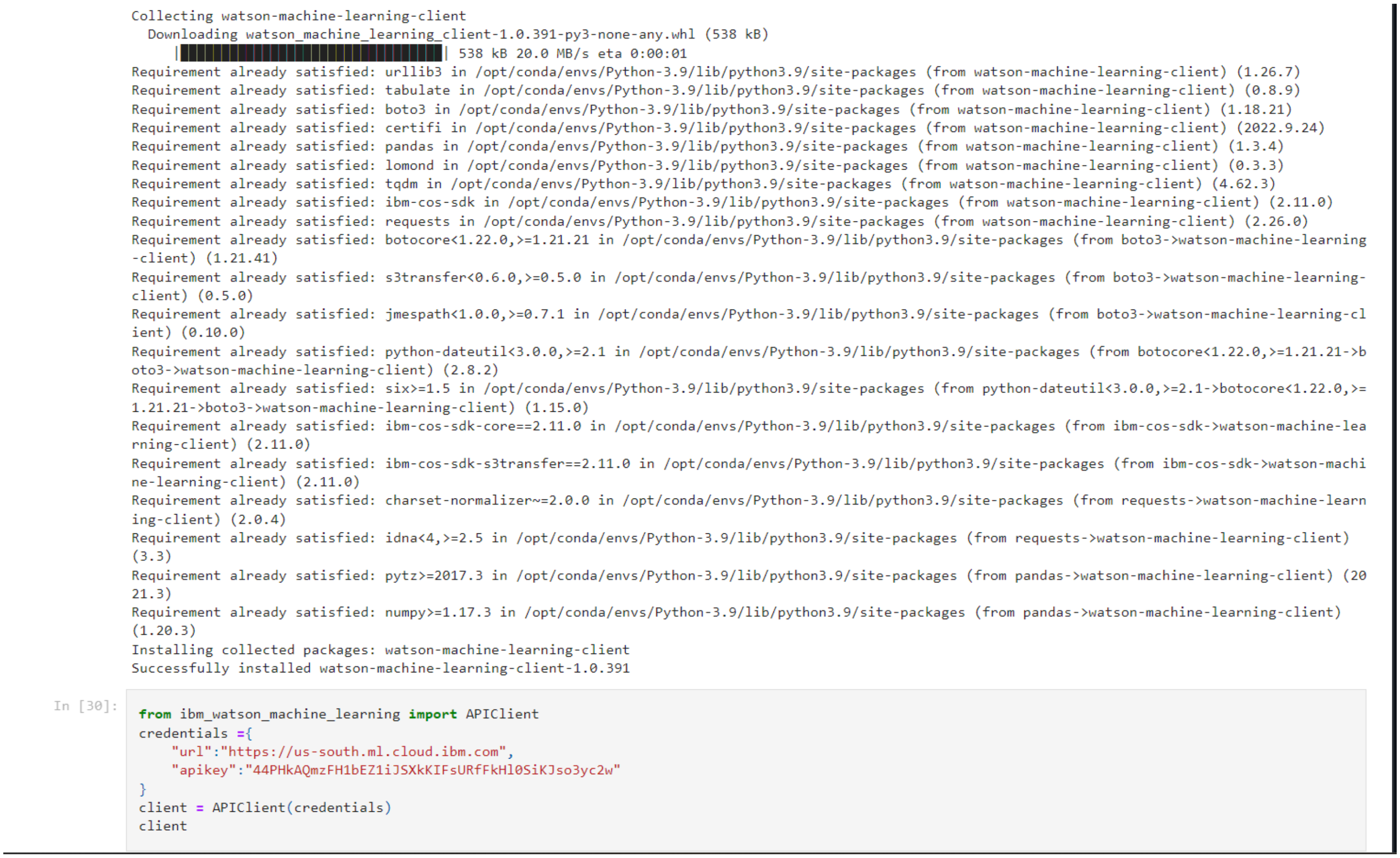
In [27]:

```
!tar -zcvf handwritten-digit-recognition-model_new.tgz mnistCNN.h5
```

```
mnistCNN.h5
```

In [28]:

```
!pip install watson-machine-learning-client --upgrade
```



```
Collecting watson-machine-learning-client
  Downloading watson_machine_learning_client-1.0.391-py3-none-any.whl (538 kB)
     |████████████████████████████████| 538 kB 20.0 MB/s eta 0:00:01
Requirement already satisfied: urllib3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (1.26.7)
Requirement already satisfied: tabulate in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (0.8.9)
Requirement already satisfied: boto3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (1.18.21)
Requirement already satisfied: certifi in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (2022.9.24)
Requirement already satisfied: pandas in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (1.3.4)
Requirement already satisfied: lomond in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (0.3.3)
Requirement already satisfied: tqdm in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (4.62.3)
Requirement already satisfied: ibm-cos-sdk in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (2.11.0)
Requirement already satisfied: requests in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (2.26.0)
Requirement already satisfied: botocore<1.22.0,>=1.21.21 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from boto3->watson-machine-learning
-client) (1.21.41)
Requirement already satisfied: s3transfer<0.6.0,>=0.5.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from boto3->watson-machine-learning-
client) (0.5.0)
Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from boto3->watson-machine-learning-cl
ient) (0.10.0)
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from botocore<1.22.0,>=1.21.21->b
oto3->watson-machine-learning-client) (2.8.2)
Requirement already satisfied: six>=1.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from python-dateutil<3.0.0,>=2.1->botocore<1.22.0,>=
1.21.21->boto3->watson-machine-learning-client) (1.15.0)
Requirement already satisfied: ibm-cos-sdk-core==2.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk->watson-machine-lea
rning-client) (2.11.0)
Requirement already satisfied: ibm-cos-sdk-s3transfer==2.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk->watson-machi
ne-learning-client) (2.11.0)
Requirement already satisfied: charset-normalizer~=2.0.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests->watson-machine-learn
ing-client) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests->watson-machine-learning-client)
(3.3)
Requirement already satisfied: pytz>=2017.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas->watson-machine-learning-client) (20
21.3)
Requirement already satisfied: numpy>=1.17.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas->watson-machine-learning-client)
(1.20.3)
Installing collected packages: watson-machine-learning-client
Successfully installed watson-machine-learning-client-1.0.391
```

In [30]:

```python
from ibm_watson_machine_learning import APIClient
credentials ={
    "url":"https://us-south.ml.cloud.ibm.com",
    "apikey":"44PHkAQmzFH1bEZ1iJSXkKIFsURfFkHl0SiKJso3yc2w"
}
client = APIClient(credentials)
client
```

In [31]:

```
client.spaces.get_details()
```

Out[31]:

```
{'resources': [{'entity': {'description': '',
    'name': 'digitclasssification',
    'scope': {'bss_account_id': 'eb25e160e3a44ea19dcb6550dbb30fed'},
    'stage': {'production': False},
    'status': {'state': 'active'},
    'storage': {'properties': {'bucket_name': '7f0ddbd1-c980-4a2d-a36d-f5ca427d3516',
      'bucket_region': 'us-south',
      'credentials': {'admin': {'access_key_id': '5a87d8278316431b8ba1dec115e5459d',
        'api_key': 'k4YGBSUaYVBbKY5-Mq6AEO1MmelUdtMb0sAJbaKjd90V',
        'secret_access_key': '02d56f22174b6f8e3b2d81df2f6637786ee79249caaa30cf',
        'service_id': 'ServiceId-7ffd7456-73c3-4706-9af9-f37c5c03a9ca'},
       'editor': {'access_key_id': 'ac09d4d47b3d44218c789dd676fe6e62',
        'api_key': 'YnkTWn1Tw8AqgcSFiMEsVTeRytXDjQ0e70E6TZwoEHAG',
        'resource_key_crn': 'crn:v1:bluemix:public:cloud-object-storage:global:a/eb25e160e3a44ea19dcb6550dbb30fed:953ecb71-6915-4c60-a924-9202c7c875d
f::',
        'secret_access_key': '1b1342075ee7e42e3ea054f460cf423b75603e4d36f70342',
        'service_id': 'ServiceId-d98d6495-1f43-43b1-aad0-5264c80b45d7'},
       'viewer': {'access_key_id': 'dc676ef8f48c45e3b09807204da48bcf',
        'api_key': 'bxFpQPjFX9D7gT7T2X7m1GH8Ir3moXeUh1X6wg5q2er3',
        'resource_key_crn': 'crn:v1:bluemix:public:cloud-object-storage:global:a/eb25e160e3a44ea19dcb6550dbb30fed:953ecb71-6915-4c60-a924-9202c7c875d
f::',
        'secret_access_key': 'cd6597636648076d40d8691d06524bafaf83d08e4dd508e2',
        'service_id': 'ServiceId-efe1354c-b1e9-48e2-a5a5-2e877c544551'}},
      'endpoint_url': 'https://s3.us-south.cloud-object-storage.appdomain.cloud',
      'guid': '953ecb71-6915-4c60-a924-9202c7c875df',
      'resource_crn': 'crn:v1:bluemix:public:cloud-object-storage:global:a/eb25e160e3a44ea19dcb6550dbb30fed:953ecb71-6915-4c60-a924-9202c7c875df::'},
     'type': 'bmcos_object_storage'}},
   'metadata': {'created_at': '2022-11-16T16:14:50.135Z',
    'creator_id': 'IBMid-6640043HNJ',
    'id': '8d67ca93-b475-44af-9f77-0d636fa5ffe7',
    'updated_at': '2022-11-16T16:15:06.534Z',
    'url': '/v2/spaces/8d67ca93-b475-44af-9f77-0d636fa5ffe7'}}]}
```

In [32]:

```
def guid_from_space_name(client,deploy):
  space = client.spaces.get_details()
  return (next(item for item in space['resources'] if item['entity']['name']==deploy)['metadata']['id'])
```

In [33]:

```
space_uid = guid_from_space_name(client,'digitclasssification')
print("Space UID = " + space_uid)
```

```
Space UID = 8d67ca93-b475-44af-9f77-0d636fa5ffe7
```

In [35]:

```
client.set.default_space(space_uid)
```

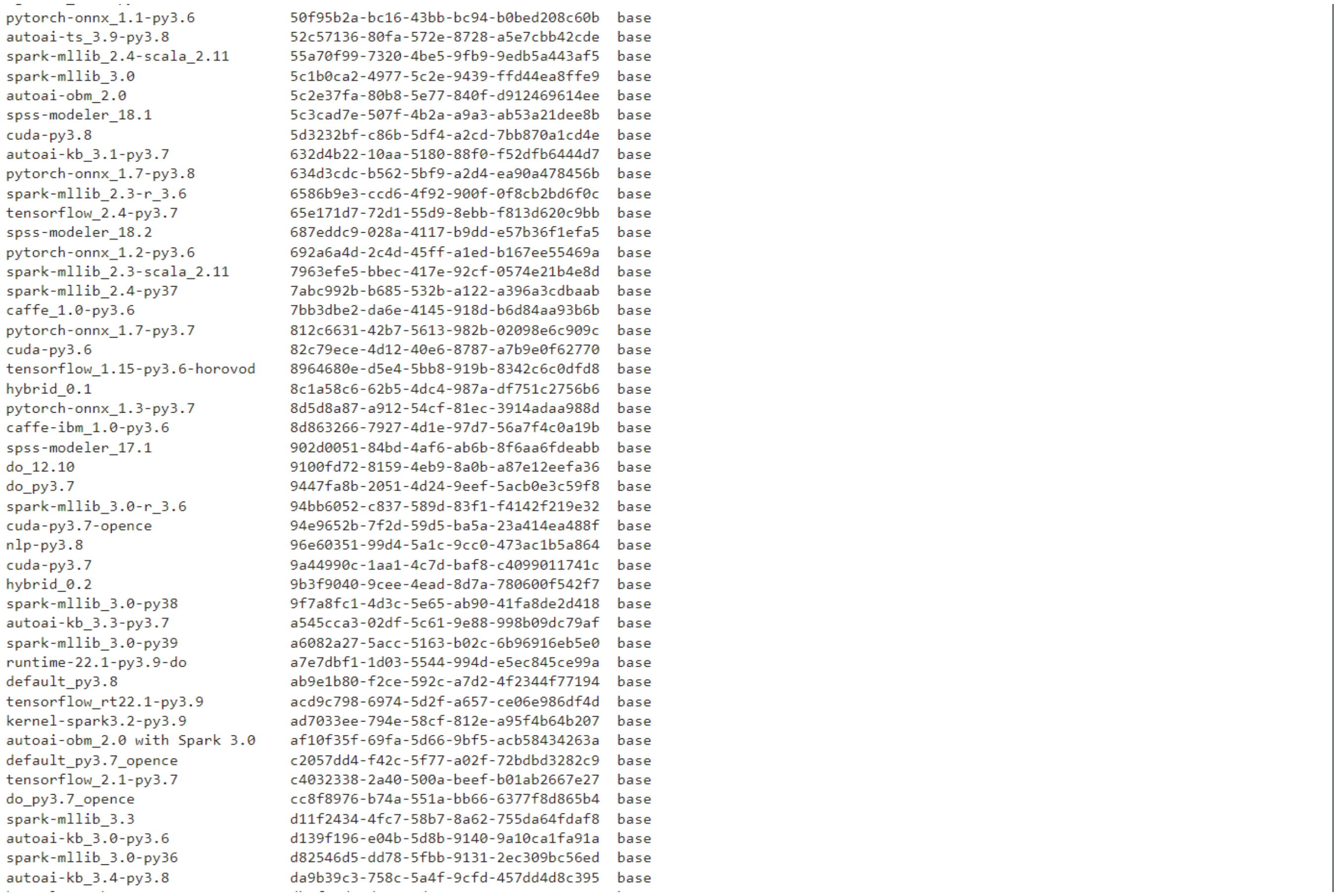
Out[35]:

```
'SUCCESS'
```

In [36]:

```
client.software_specifications.list(limit=100)
```

```
-------------------------------  ------------------------------------  ----
NAME                             ASSET_ID                              TYPE
default_py3.6                    0062b8c9-8b7d-44a0-a9b9-46c416adcbd9  base
kernel-spark3.2-scala2.12        020d69ce-7ac1-5e68-ac1a-31189867356a  base
pytorch-onnx_1.3-py3.7-edt       069ea134-3346-5748-b513-49120e15d288  base
scikit-learn_0.20-py3.6          09c5a1d0-9c1e-4473-a344-eb7b665ff687  base
spark-mllib_3.0-scala_2.12       09f4cff0-90a7-5899-b9ed-1ef348aebdee  base
pytorch-onnx_rt22.1-py3.9        0b848dd4-e681-5599-be41-b5f6fccc6471  base
ai-function_0.1-py3.6            0cdb0f1e-5376-4f4d-92dd-da3b69aa9bda  base
shiny-r3.6                       0e6e79df-875e-4f24-8ae9-62dcc2148306  base
tensorflow_2.4-py3.7-horovod     1092590a-307d-563d-9b62-4eb7d64b3f22  base
pytorch_1.1-py3.6                10ac12d6-6b30-4ccd-8392-3e922c096a92  base
tensorflow_1.15-py3.6-ddl        111e41b3-de2d-5422-a4d6-bf776828c4b7  base
autoai-kb_rt22.2-py3.10          125b6d9a-5b1f-5e8d-972a-b251688ccf40  base
runtime-22.1-py3.9               12b83a17-24d8-5082-900f-0ab31fbfd3cb  base
scikit-learn_0.22-py3.6          154010fa-5b3b-4ac1-82af-4d5ee5abbc85  base
default_r3.6                     1b70aec3-ab34-4b87-8aa0-a4a3c8296a36  base
pytorch-onnx_1.3-py3.6           1bc6029a-cc97-56da-b8e0-39c3880dbbe7  base
kernel-spark3.3-r3.6             1c9e5454-f216-59dd-a20e-474a5cdf5988  base
pytorch-onnx_rt22.1-py3.9-edt    1d362186-7ad5-5b59-8b6c-9d0880bde37f  base
tensorflow_2.1-py3.6             1eb25b84-d6ed-5dde-b6a5-3fbdf1665666  base
spark-mllib_3.2                  20047f72-0a98-58c7-9ff5-a77b012eb8f5  base
tensorflow_2.4-py3.8-horovod     217c16f6-178f-56bf-824a-b19f20564c49  base
runtime-22.1-py3.9-cuda          26215f05-08c3-5a41-a1b0-da66306ce658  base
do_py3.8                         295addb5-9ef9-547e-9bf4-92ae3563e720  base
autoai-ts_3.8-py3.8              2aa0c932-798f-5ae9-abd6-15e0c2402fb5  base
tensorflow_1.15-py3.6            2b73a275-7cbf-420b-a912-eae7f436e0bc  base
kernel-spark3.3-py3.9            2b7961e2-e3b1-5a8c-a491-482c8368839a  base
pytorch_1.2-py3.6                2c8ef57d-2687-4b7d-acce-01f94976dac1  base
spark-mllib_2.3                  2e51f700-bca0-4b0d-88dc-5c6791338875  base
pytorch-onnx_1.1-py3.6-edt       32983cea-3f32-4400-8965-dde874a8d67e  base
spark-mllib_3.0-py37             36507ebe-8770-55ba-ab2a-eafe787600e9  base
spark-mllib_2.4                  390d21f8-e58b-4fac-9c55-d7ceda621326  base
autoai-ts_rt22.2-py3.10          396b2e83-0953-5b86-9a55-7ce1628a406f  base
xgboost_0.82-py3.6               39e31acd-5f30-41dc-ae44-60233c80306e  base
```



```
pytorch-onnx_1.1-py3.6           50f95b2a-bc16-43bb-bc94-b0bed208c60b  base
autoai-ts_3.9-py3.8              52c57136-80fa-572e-8728-a5e7cbb42cde  base
spark-mllib_2.4-scala_2.11       55a70f99-7320-4be5-9fb9-9edb5a443af5  base
spark-mllib_3.0                  5c1b0ca2-4977-5c2e-9439-ffd44ea8ffe9  base
autoai-obm_2.0                   5c2e37fa-80b8-5e77-840f-d912469614ee  base
spss-modeler_18.1                5c3cad7e-507f-4b2a-a9a3-ab53a21dee8b  base
cuda-py3.8                       5d3232bf-c86b-5df4-a2cd-7bb870a1cd4e  base
autoai-kb_3.1-py3.7              632d4b22-10aa-5180-88f0-f52dfb6444d7  base
pytorch-onnx_1.7-py3.8           634d3cdc-b562-5bf9-a2d4-ea90a478456b  base
spark-mllib_2.3-r_3.6            6586b9e3-ccd6-4f92-900f-0f8cb2bd6f0c  base
tensorflow_2.4-py3.7             65e171d7-72d1-55d9-8ebb-f813d620c9bb  base
spss-modeler_18.2                687eddc9-028a-4117-b9dd-e57b36f1efa5  base
pytorch-onnx_1.2-py3.6           692a6a4d-2c4d-45ff-a1ed-b167ee55469a  base
spark-mllib_2.3-scala_2.11       7963efe5-bbec-417e-92cf-0574e21b4e8d  base
spark-mllib_2.4-py37             7abc992b-b685-532b-a122-a396a3cdbaab  base
caffe_1.0-py3.6                  7bb3dbe2-da6e-4145-918d-b6d84aa93b6b  base
pytorch-onnx_1.7-py3.7           812c6631-42b7-5613-982b-02098e6c909c  base
cuda-py3.6                       82c79ece-4d12-40e6-8787-a7b9e0f62770  base
tensorflow_1.15-py3.6-horovod    8964680e-d5e4-5bb8-919b-8342c6c0dfd8  base
hybrid_0.1                       8c1a58c6-62b5-4dc4-987a-df751c2756b6  base
pytorch-onnx_1.3-py3.7           8d5d8a87-a912-54cf-81ec-3914adaa988d  base
caffe-ibm_1.0-py3.6              8d863266-7927-4d1e-97d7-56a7f4c0a19b  base
spss-modeler_17.1                902d0051-84bd-4af6-ab6b-8f6aa6fdeabb  base
do_12.10                         9100fd72-8159-4eb9-8a0b-a87e12eefa36  base
do_py3.7                         9447fa8b-2051-4d24-9eef-5acb0e3c59f8  base
spark-mllib_3.0-r_3.6            94bb6052-c837-589d-83f1-f4142f219e32  base
cuda-py3.7-opence                94e9652b-7f2d-59d5-ba5a-23a414ea488f  base
nlp-py3.8                        96e60351-99d4-5a1c-9cc0-473ac1b5a864  base
cuda-py3.7                       9a44990c-1aa1-4c7d-baf8-c4099011741c  base
hybrid_0.2                       9b3f9040-9cee-4ead-8d7a-780600f542f7  base
spark-mllib_3.0-py38             9f7a8fc1-4d3c-5e65-ab90-41fa8de2d418  base
autoai-kb_3.3-py3.7              a545cca3-02df-5c61-9e88-998b09dc79af  base
spark-mllib_3.0-py39             a6082a27-5acc-5163-b02c-6b96916eb5e0  base
runtime-22.1-py3.9-do            a7e7dbf1-1d03-5544-994d-e5ec845ce99a  base
default_py3.8                    ab9e1b80-f2ce-592c-a7d2-4f2344f77194  base
tensorflow_rt22.1-py3.9          acd9c798-6974-5d2f-a657-ce06e986df4d  base
kernel-spark3.2-py3.9            ad7033ee-794e-58cf-812e-a95f4b64b207  base
autoai-obm_2.0 with Spark 3.0    af10f35f-69fa-5d66-9bf5-acb58434263a  base
default_py3.7_opence             c2057dd4-f42c-5f77-a02f-72bdbd3282c9  base
tensorflow_2.1-py3.7             c4032338-2a40-500a-beef-b01ab2667e27  base
do_py3.7_opence                  cc8f8976-b74a-551a-bb66-6377f8d865b4  base
spark-mllib_3.3                  d11f2434-4fc7-58b7-8a62-755da64fdaf8  base
autoai-kb_3.0-py3.6              d139f196-e04b-5d8b-9140-9a10ca1fa91a  base
spark-mllib_3.0-py36             d82546d5-dd78-5fbb-9131-2ec309bc56ed  base
autoai-kb_3.4-py3.8              da9b39c3-758c-5a4f-9cfd-457dd4d8c395  base
```

```
tensorflow_2.1-py3.7-horovod     e384fce5-fdd1-53f8-bc71-11326c9c635f  base
default_py3.7                    e4429883-c883-42b6-87a8-f419d64088cd  base
do_22.1                          e51999ba-6452-5f1f-8287-17228b88b652  base
autoai-obm_3.2                   eae86aab-da30-5229-a6a6-1d0d4e368983  base
tensorflow_rt22.2-py3.10         f65bd165-f057-55de-b5cb-f97cf2c0f393  base
do_20.1                          f686cdd9-7904-5f9d-a732-01b0d6b10dc5  base
pytorch-onnx_rt22.2-py3.10-edt   f8a05d07-e7cd-57bb-a10b-23f1d4b837ac  base
scikit-learn_0.19-py3.6          f963fa9d-4bb7-5652-9c5d-8d9289ef6ad9  base
tensorflow_2.4-py3.8             fe185c44-9a99-5425-986b-59bd1d2eda46  base
-------------------------------  ------------------------------------  ----
```

In [38]:

```
software_space_uid = client.software_specifications.get_uid_by_name('tensorflow_rt22.1-py3.9')
software_space_uid
```

Out[38]:

```
'acd9c798-6974-5d2f-a657-ce06e986df4d'
```

In [39]:

```
model_details = client.repository.store_model(model='handwritten-digit-recognition-model_new.tgz',meta_props={
    client.repository.ModelMetaNames.NAME:"CNN Digit recognition model",
    client.repository.ModelMetaNames.TYPE:"tensorflow_2.7",
    client.repository.ModelMetaNames.SOFTWARE_SPEC_UID:software_space_uid
})
```

In [40]:

```
model_details
```

Out[40]:

```
{'entity': {'hybrid_pipeline_software_specs': [],
  'software_spec': {'id': 'acd9c798-6974-5d2f-a657-ce06e986df4d',
   'name': 'tensorflow_rt22.1-py3.9'},
  'type': 'tensorflow_2.7'},
 'metadata': {'created_at': '2022-11-17T09:30:09.906Z',
  'id': '27a4d1d3-c733-4917-8fa8-6873fea963af',
  'modified_at': '2022-11-17T09:30:12.712Z',
  'name': 'CNN Digit recognition model',
  'owner': 'IBMid-6640043HNJ',
  'resource_key': '4a2dec75-86c5-4b2a-bb87-1a421b4ac3e5',
  'space_id': '8d67ca93-b475-44af-9f77-0d636fa5ffe7'},
 'system': {'warnings': []}}
```

In [41]:

```
model_id = client.repository.get_model_id(model_details)
model_id
```

Out[41]:

```
'27a4d1d3-c733-4917-8fa8-6873fea963af'
```

In [42]:

```
client.repository.download(model_id,'DigitRecog_IBM_model.tar.gz')
```

```
Successfully saved model content to file: 'DigitRecog_IBM_model.tar.gz'
```

Out[42]:

```
'/home/wsuser/work/models/DigitRecog_IBM_model.tar.gz'
```

In [43]:

```
ls
```

```
DigitRecog_IBM_model.tar.gz                  mnistCNN.h5
handwritten-digit-recognition-model_new.tgz
```

In [54]:

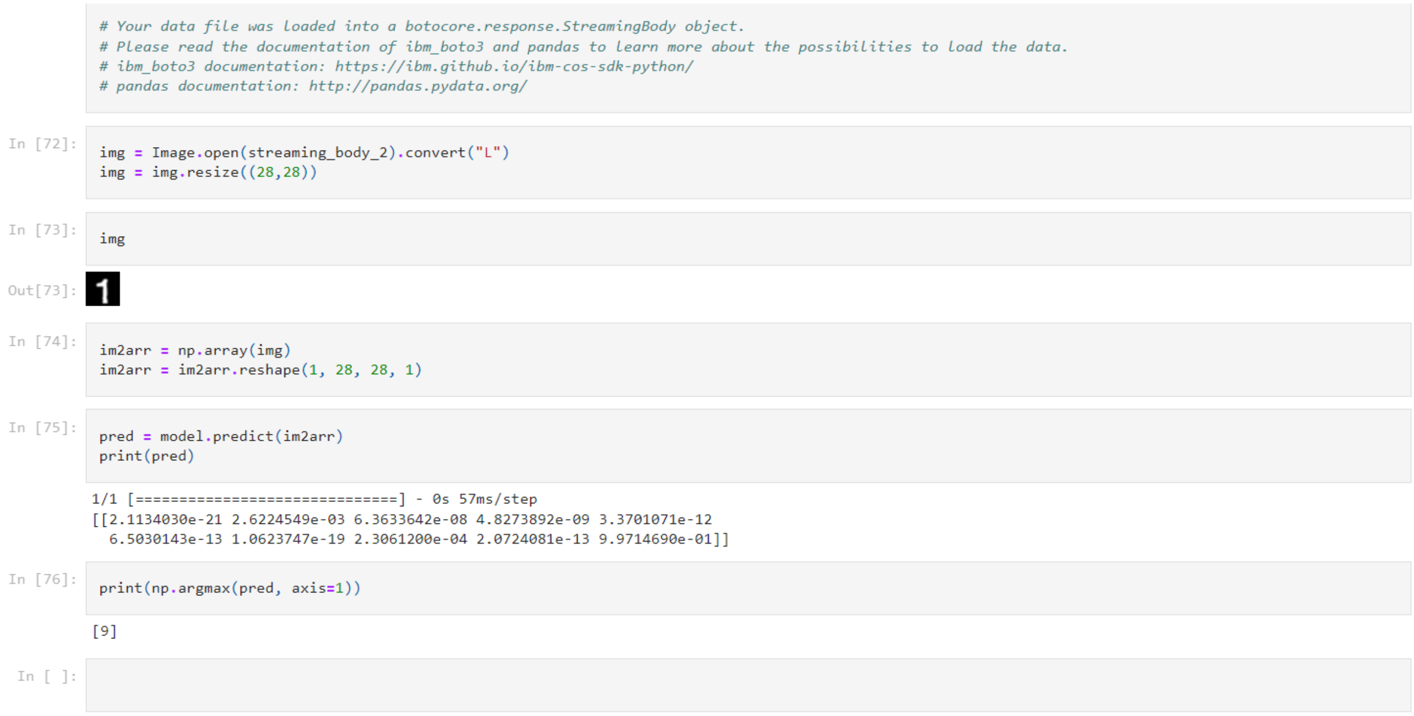
```
from tensorflow.keras.models import load_model
from keras.preprocessing import image
from PIL import Image
import numpy as np
```

In [55]:

```
model = load_model("mnistCNN.h5")
```

In [70]:

```
import os, types
import pandas as pd
from botocore.client import Config
import ibm_boto3

def __iter__(self): return 0

# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.
# You might want to remove those credentials before you share the notebook.
cos_client = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='0RQZsjyJL-OgB_JxaoYAECf5_gIBTHNrC7xqb7RR37fL',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.private.us.cloud-object-storage.appdomain.cloud')

bucket = 'digitrecognition-donotdelete-pr-2k6azacjnc2rlg'
object_key = '1.png'

streaming_body_2 = cos_client.get_object(Bucket=bucket, Key=object_key)['Body']
```



```
# Your data file was loaded into a botocore.response.StreamingBody object.
# Please read the documentation of ibm_boto3 and pandas to learn more about the possibilities to load the data.
# ibm_boto3 documentation: https://ibm.github.io/ibm-cos-sdk-python/
# pandas documentation: http://pandas.pydata.org/
```

In [72]:

```
img = Image.open(streaming_body_2).convert("L")
img = img.resize((28,28))
```

In [73]:

```
img
```

Out[73]:

In [74]:

```
im2arr = np.array(img)
im2arr = im2arr.reshape(1, 28, 28, 1)
```

In [75]:

```
pred = model.predict(im2arr)
print(pred)
```

```
1/1 [==============================] - 0s 57ms/step
[[2.1134030e-21 2.6224549e-03 6.3633642e-08 4.8273892e-09 3.3701071e-12
  6.5030143e-13 1.0623747e-19 2.3061200e-04 2.0724081e-13 9.9714690e-01]]
```

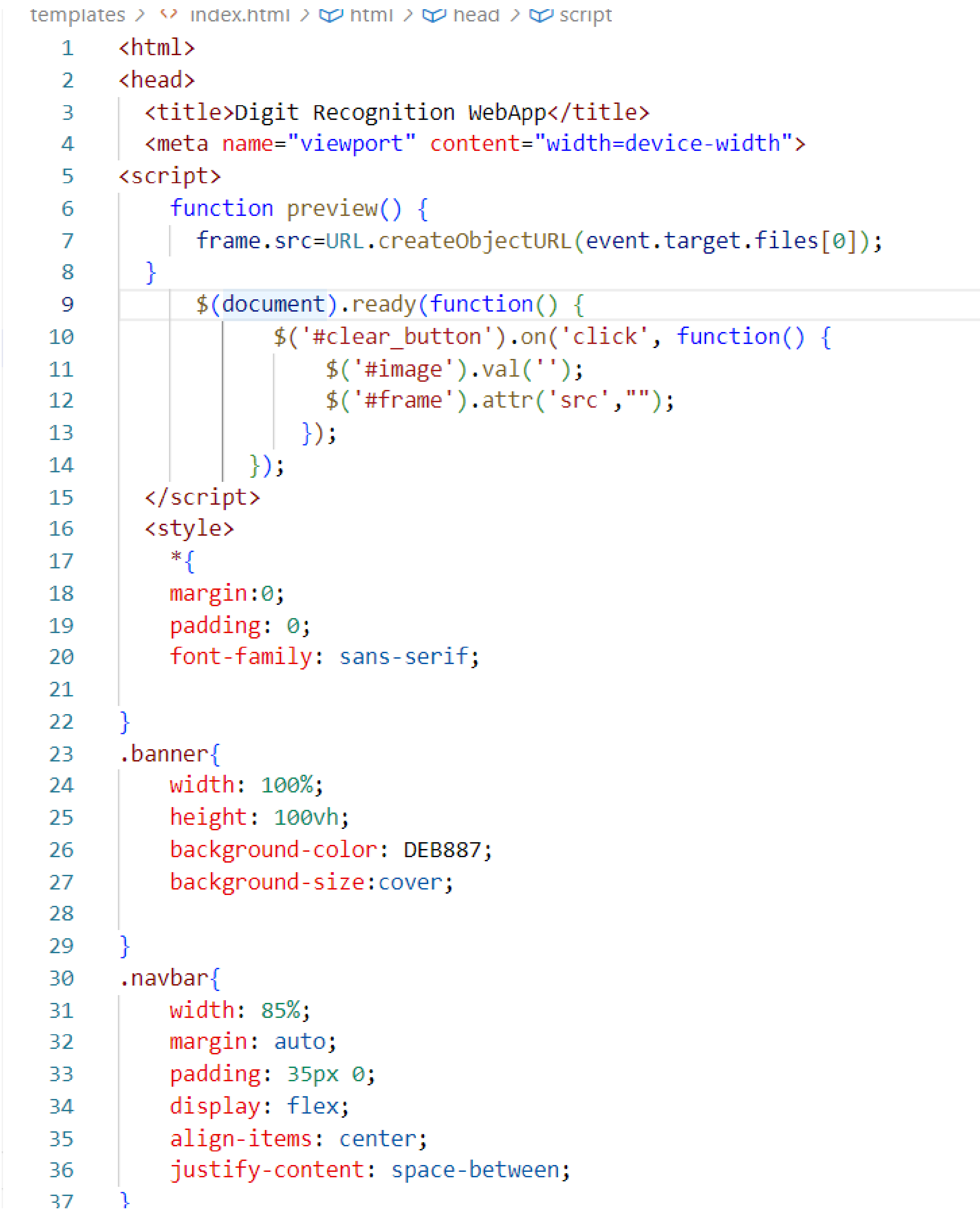
In [76]:

```
print(np.argmax(pred, axis=1))
```

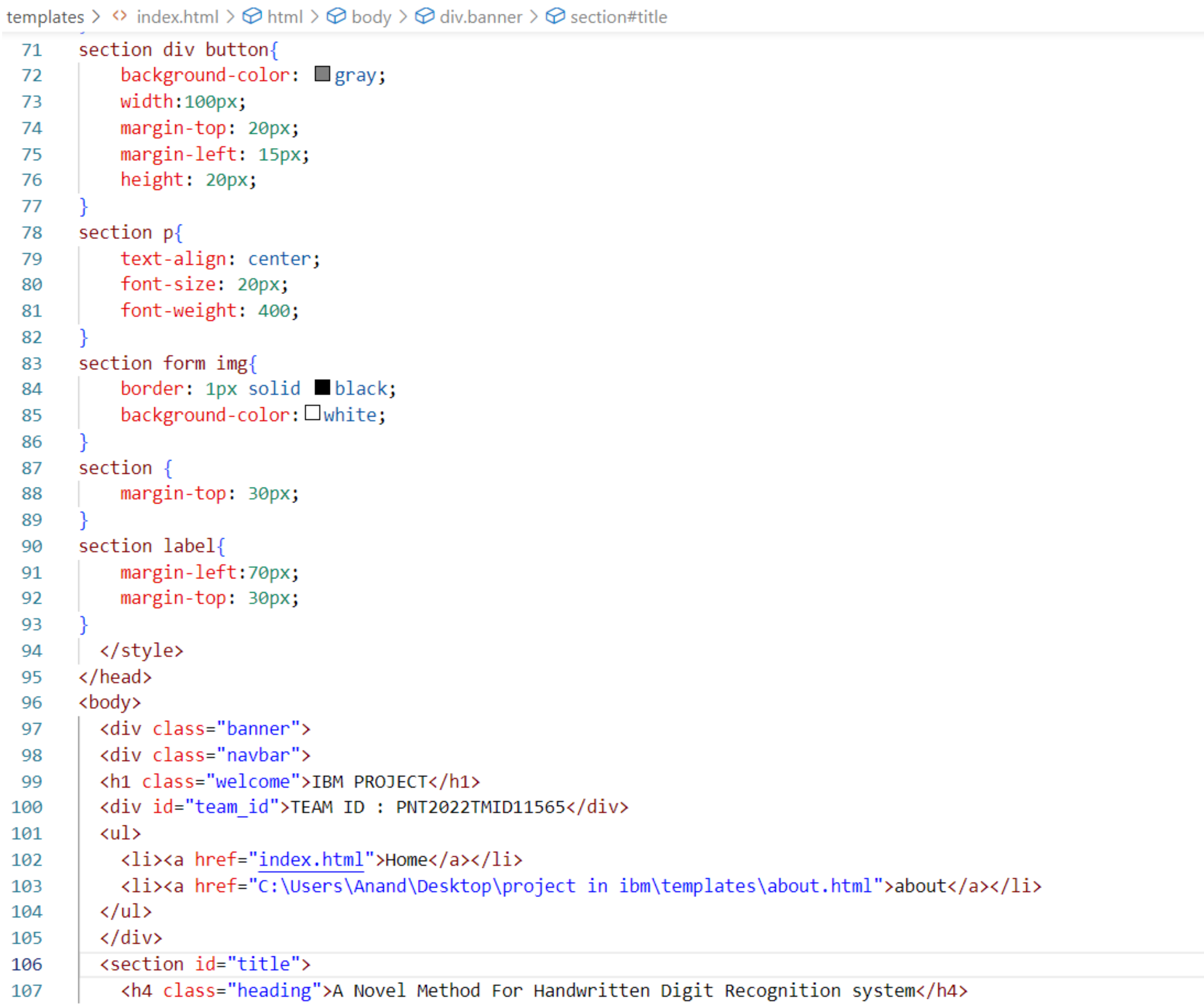
```
[9]
```

In [ ]:

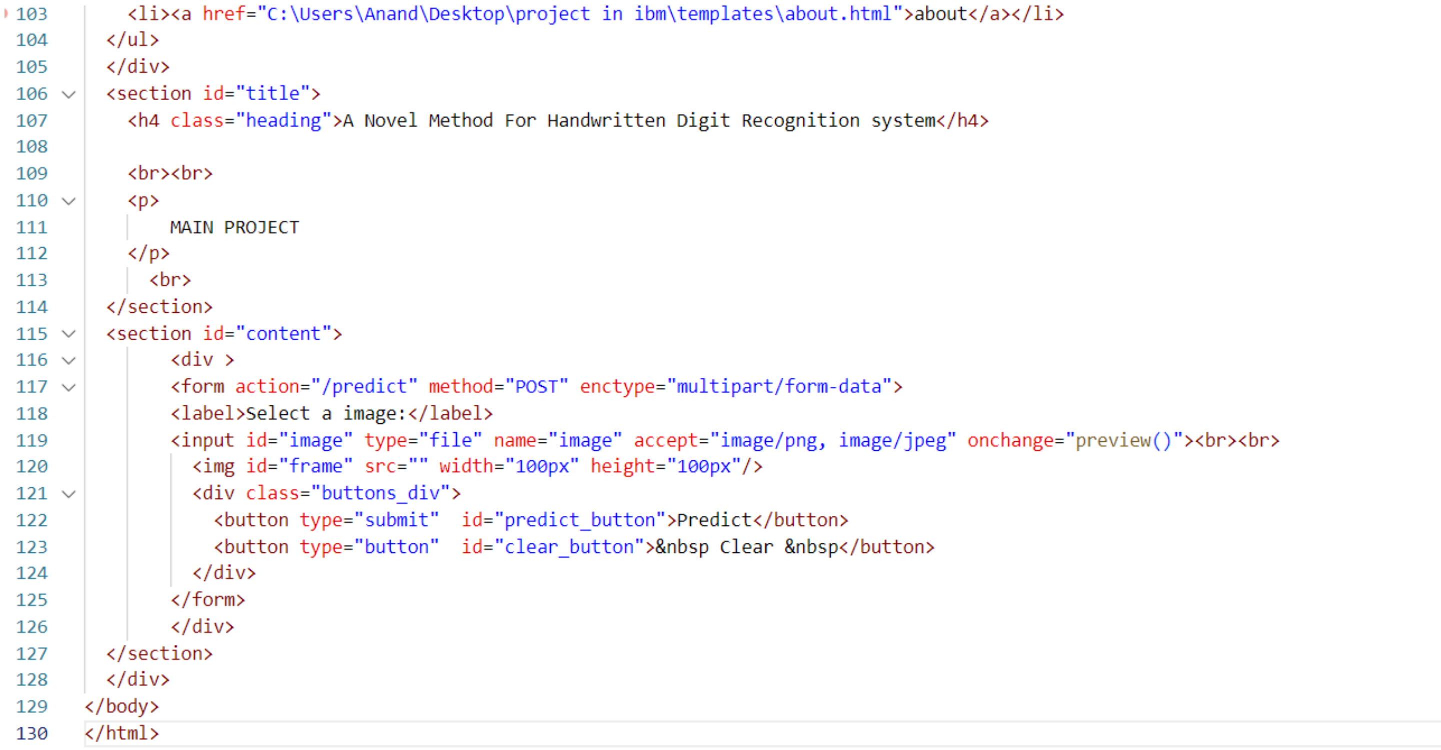
HOME PAGE(HTML) – index.html



```
<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;
}
```

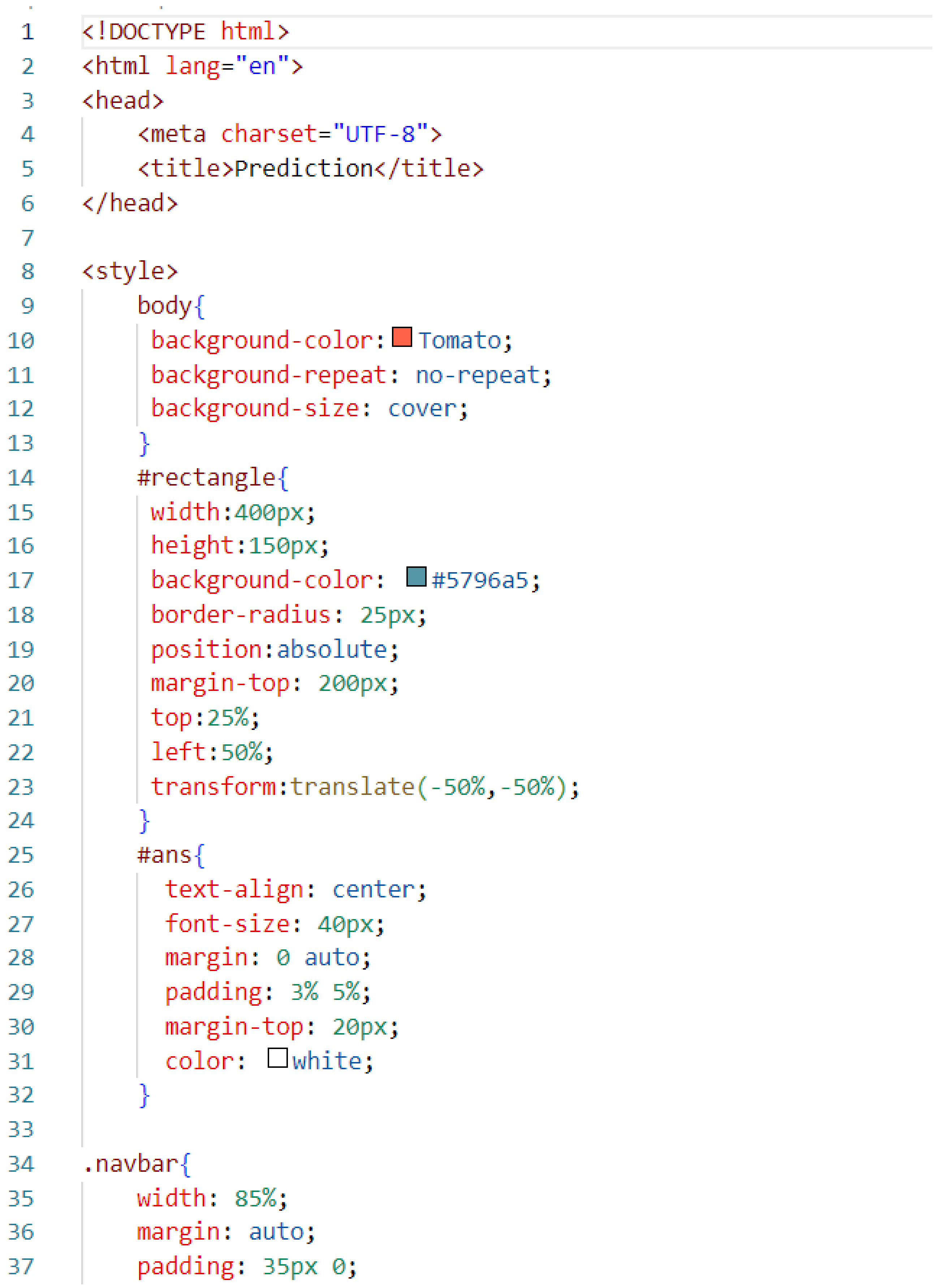

templates > index.html > html > body > div.banner > section#title

```
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 : PNT2022TMID11565</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>
```

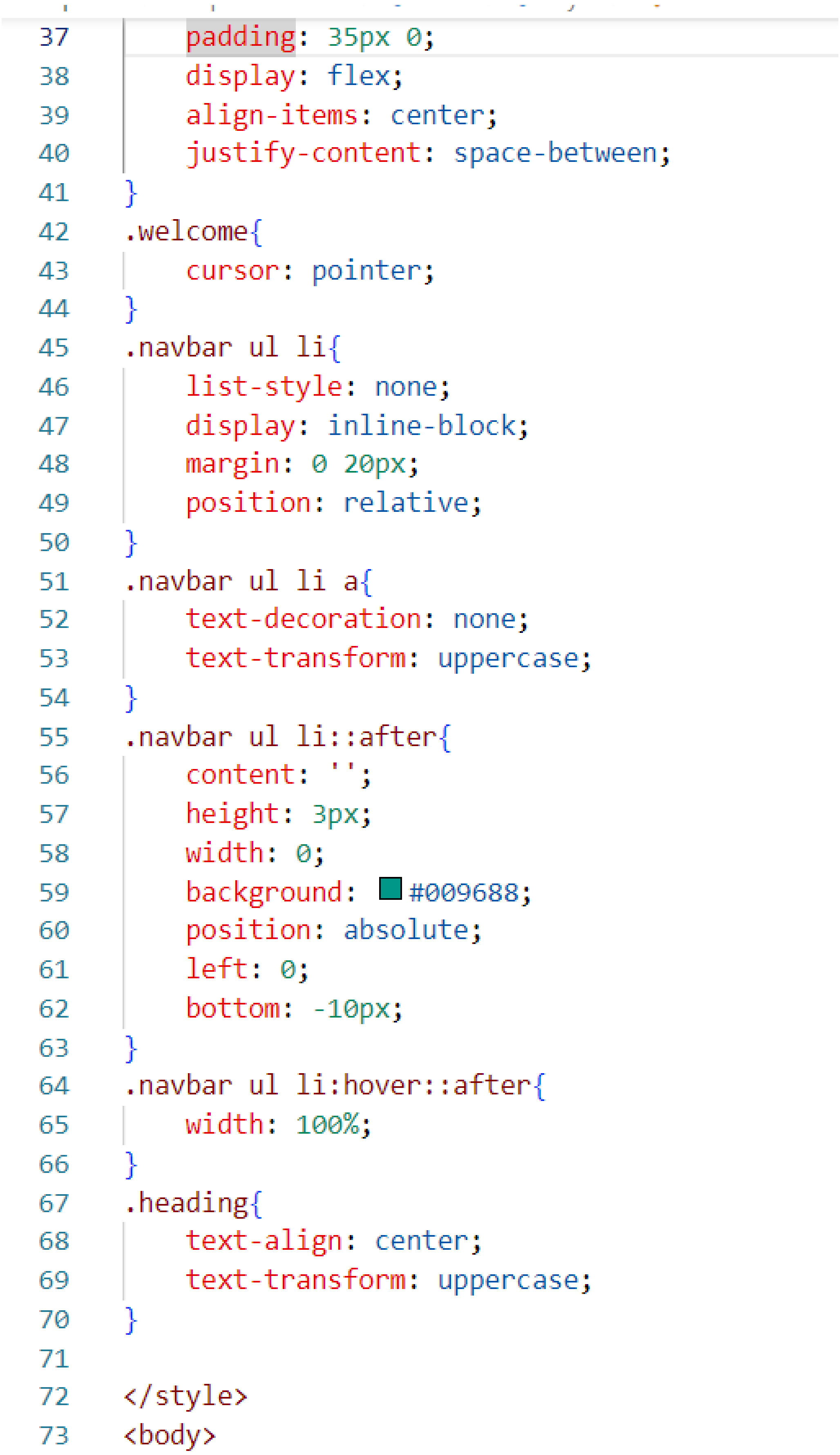


```
  <li><a href="C:\Users\Anand\Desktop\project in ibm\templates\about.html">about</a></li>
 </ul>
 </div>
 <section id="title">
  <h4 class="heading">A Novel Method For Handwritten Digit Recognition system</h4>

  <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>
```

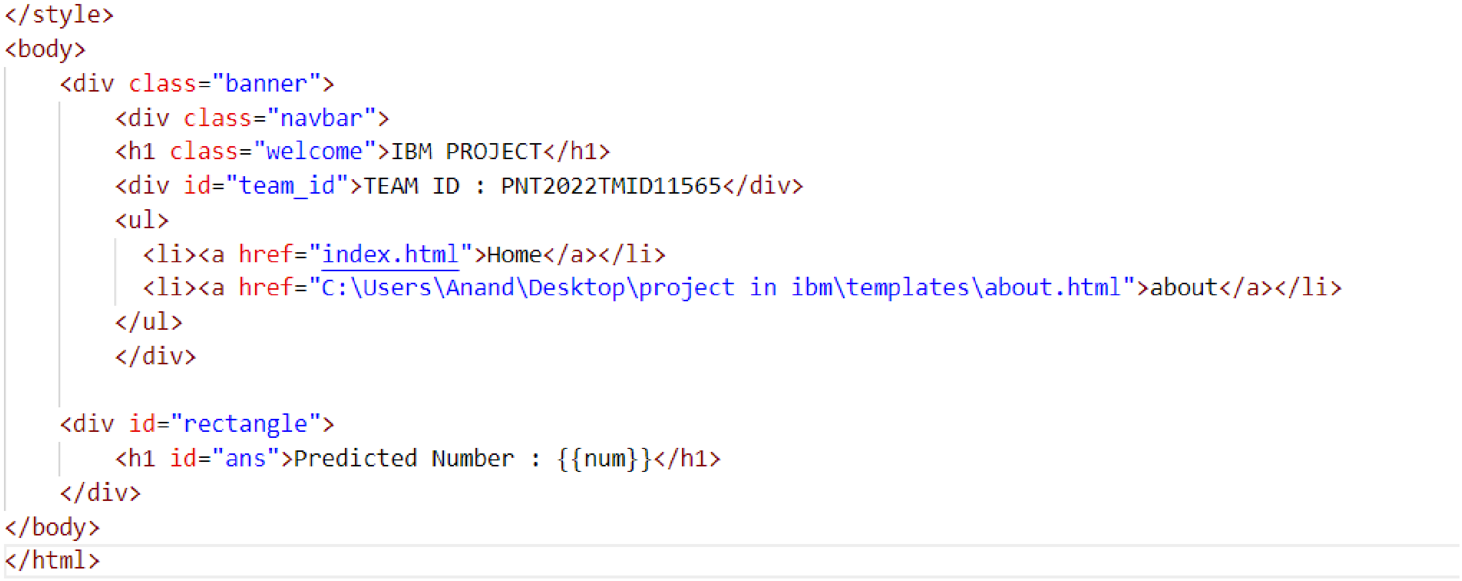
PREDICT PAGE (HTML) – predict.html



```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;
```

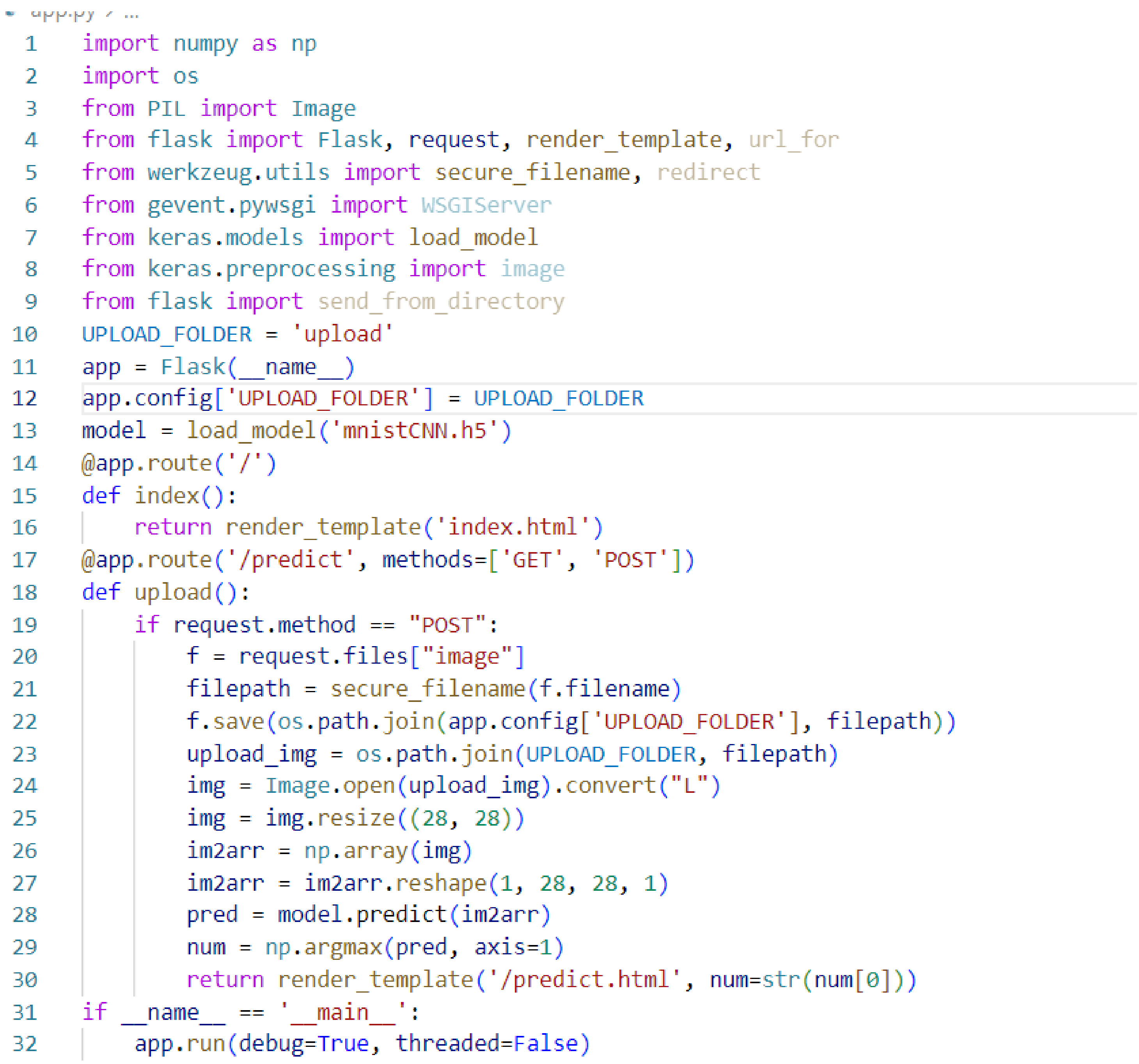


```
    padding: 35px 0;
    display: flex;
    align-items: center;
    justify-content: space-between;
}
.welcome{
    cursor: pointer;
}
.navbar ul li{
    list-style: none;
    display: inline-block;
    margin: 0 20px;
    position: relative;
}
.navbar ul li a{
    text-decoration: none;
    text-transform: 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;
}

</style>
<body>
```



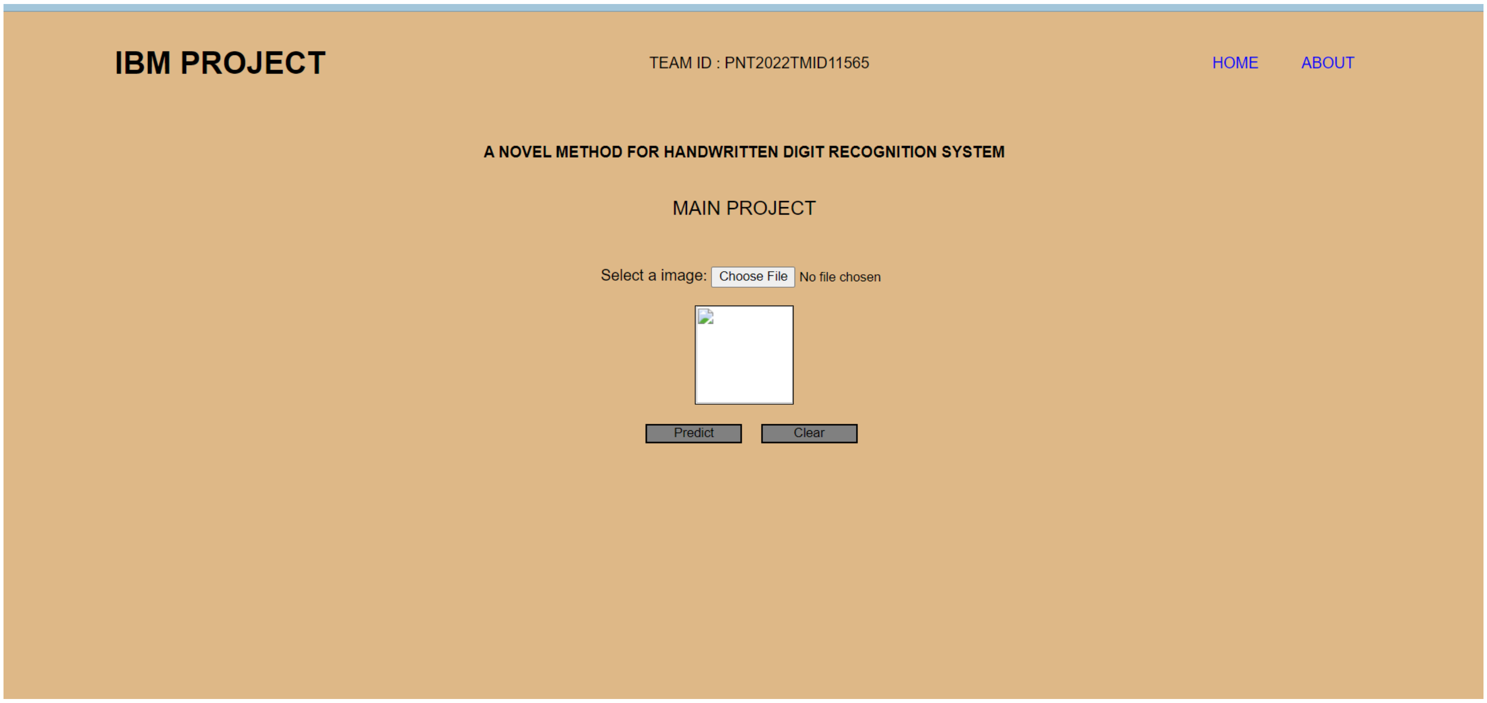
```html
</style>
<body>
    <div class="banner">
        <div class="navbar">
        <h1 class="welcome">IBM PROJECT</h1>
        <div id="team_id">TEAM ID : PNT2022TMID11565</div>
        <ul>
          <li><a href="index.html">Home</a></li>
          <li><a href="C:\Users\Anand\Desktop\project in ibm\templates\about.html">about</a></li>
        </ul>
        </div>

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

FLASK APP - app.py



```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)
```

SCREENSHOTS:



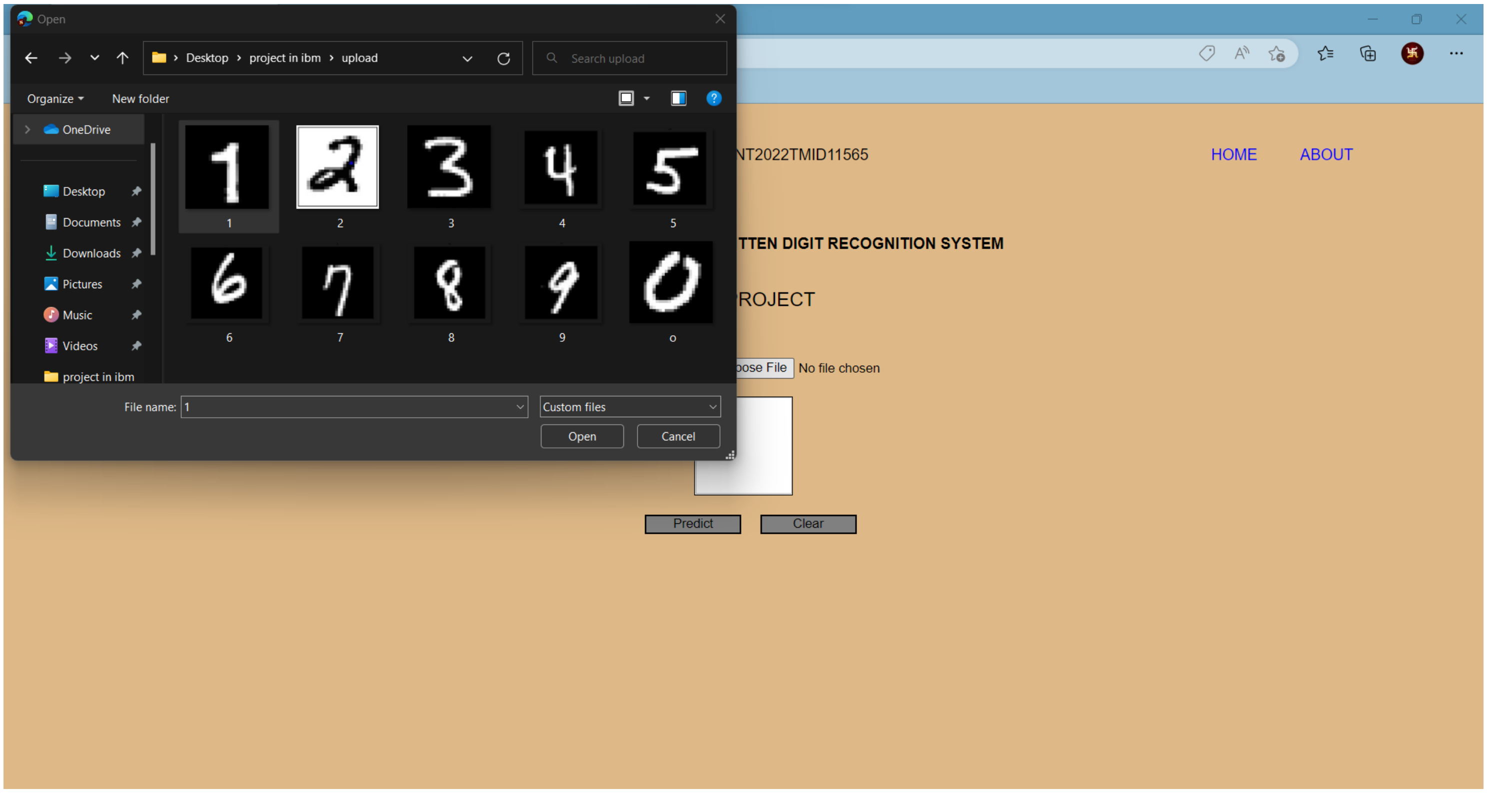
IBM PROJECT

TEAM ID : PNT2022TMID11565

A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

MAIN PROJECT

Select a image: Choose File No file chosen



File name: 1

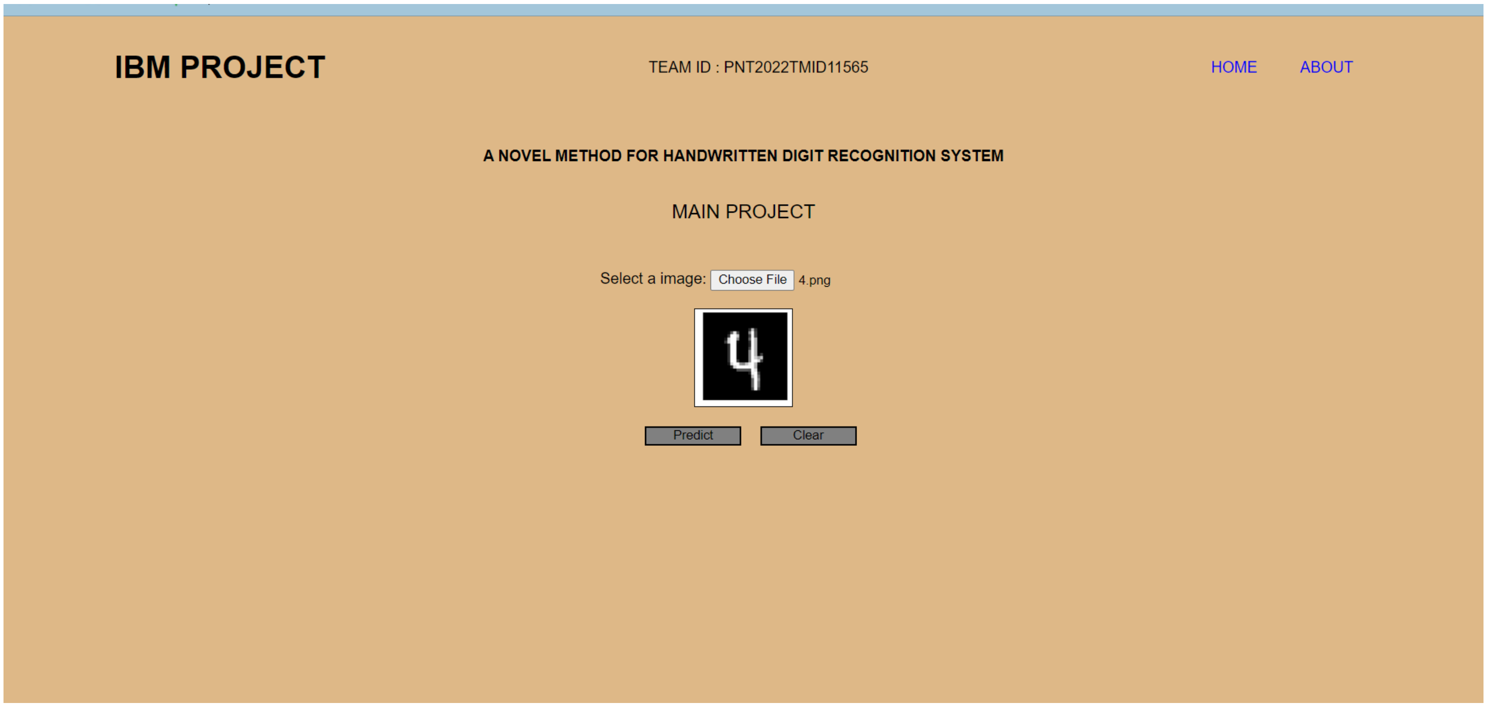
Custom files

NT2022TMID11565

TTEN DIGIT RECOGNITION SYSTEM

ROJECT

ose File No file chosen



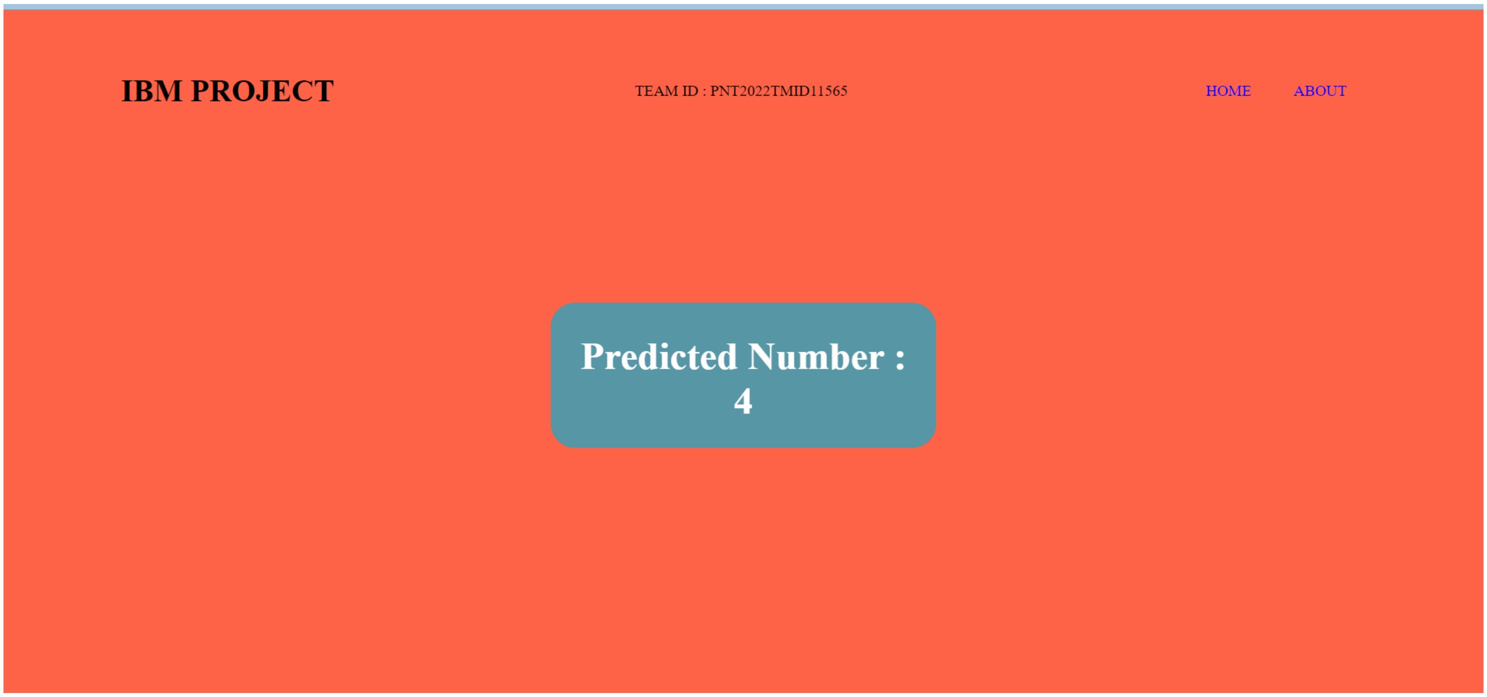
IBM PROJECT

TEAM ID : PNT2022TMID11565

A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

MAIN PROJECT

Select a image: Choose File 4.png



IBM PROJECT

TEAM ID : PNT2022TMID11565

Predicted Number : 4

GITHUB LINK:

https://github.com/IBM-EPBL/IBM-Project-34909-1660279371