IBM PROJECT

A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION

TEAM ID : PNT2022TMID07944

Submitted by

BHARANI SRI A	**AC19UCS015**
CHARUMATHI M	**AC19UCS019**
DIVYASRI V	**AC19UCS027**
JOTHISRI S	**AC19UCS044**

In the partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

ADHIYAMAAN COLLEGE OF ENGINEERING

(AUTONOMOUS), HOSUR

ANNA UNIVERSITY: CHENNAI – 600 025

NOVEMBER 2022

ABSTRACT

- In the modern world every individual has different styles of writing So it might be difficult for the system to recognise the digits.

- The purpose of this project is to build an automatic handwritten digit recognition method for the recognition of handwritten digits. The main objective of this paper is to ensure effective and reliable approaches for recognition of handwriting.

- The applications of digit recognition include 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 handwritten digits and which is submitted by users by way of a scanner, tablet, and other digital devices.

TABLE OF CONTENT

1 . INTRODUCTION

1.1.Project overview

- Handwriting recognition is one of the process to provide the ability to machines to recognize the human handwritten digits. Because of the progress in the field of science and technology, everything is being digitized to reduce human effort.
- Hence, there comes a need for handwritten digit recognition in many real-time applications. The MNIST data set is widely used for this recognition process and it has 70000 handwritten digits.
- By using 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 analyzed by the model and the detected result is returned to the UI.

1.2.Purpose

The purpose of the handwritten 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. Handwritten digit recognition system (HDR) is meant for receiving and interpreting handwritten input in the form of pictures or paper documents.

2. LITERATURE SURVEY

2.1 Existing problem

- In the existing solution, OCR(Optical Character Recognition) algorithm is used to train the model.It requires huge space to process the model while reducing the dimension the quality of image could be lost.
- OCR text works efficiently with the printed text only and not with handwritten text,Not cent percent accurate there are likely to be some mistakes made during the method.
- OCR systems are very expensive compared to CNN.

2.2 References

- Hand digit recognition using various neural network approaches -

  https://ijarcce.com/wp-content/uploads/2015/03/IJARCCE2C.pdf

- Feature extraction methods for handwritten digits recognition -

  https://www.isical.ac.in/~vlrg/sites/default/files/Pulak/Off-Line%20Handwritten%20OCR.pdf

- Hand digit recognition using image processing and neural networks -

  http://www.iaeng.org/publication/WCE2007/WCE2007_pp648-651.pdf

- Fast efficient artificial neural network for handwritten digit recognition -

  https://www.researchgate.net/publication/341284810_Effective_Handwritten_Digit_Recognition_using_Deep_Convolution_Neural_Network

- Intelligent handwritten digit recognition using artificial neural network -

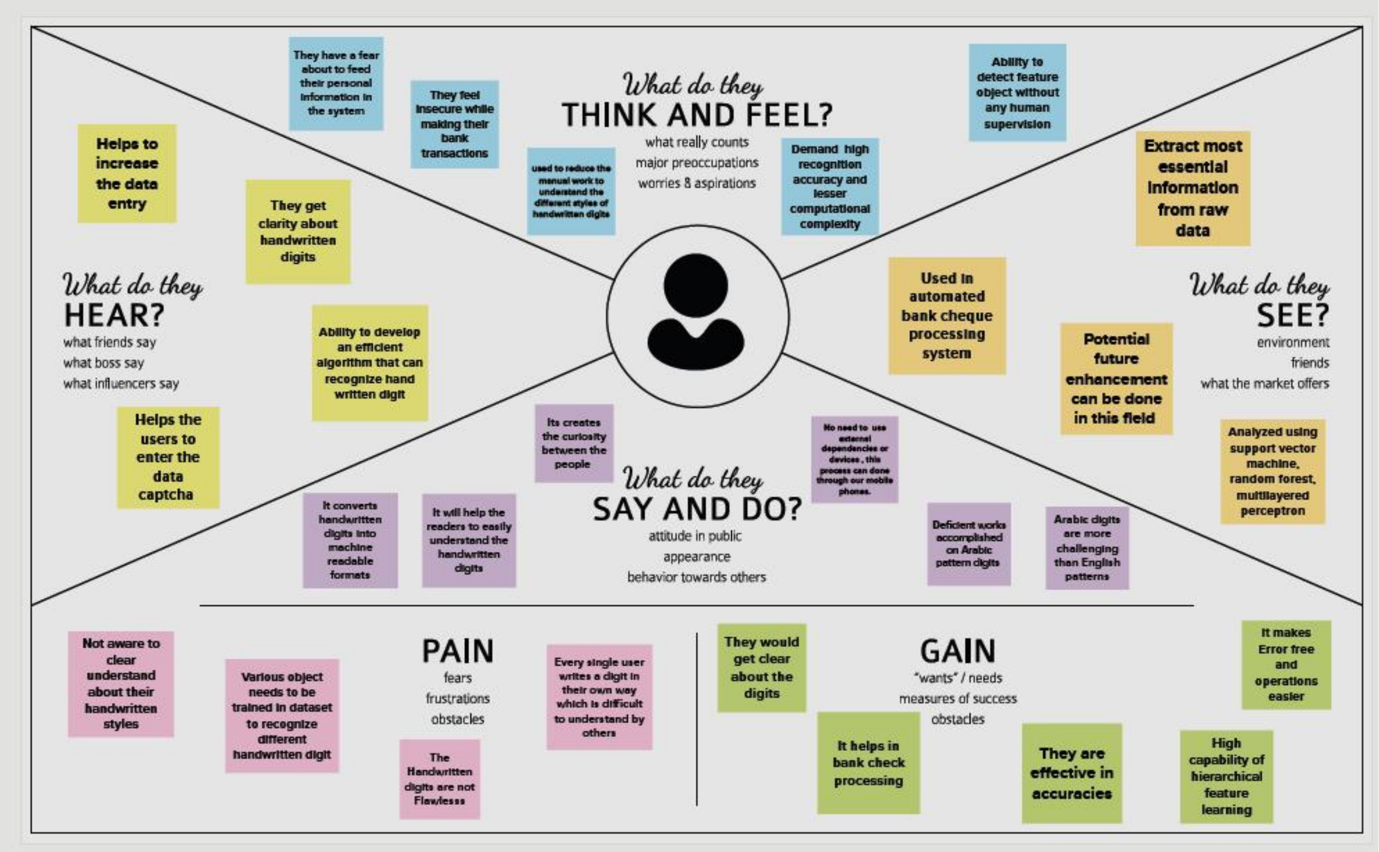
  https://www.ijera.com/papers/Vol5_issue5/Part%20-%203/H505034651.pdf

2.3 Problem Statement:

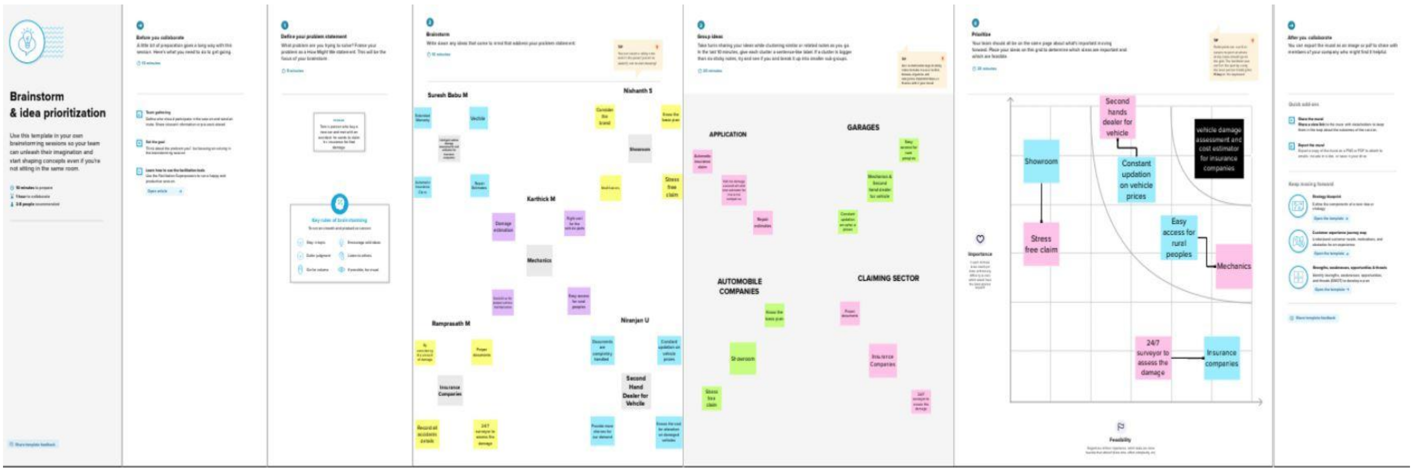
The goal of this project is to create a model that will be able to recognize and determine the handwritten digits from its image by using the concepts of Convolution Neural Network. Though the goal is to create a model which can recognize the digits of an individual's handwriting. The major goal of the proposed system is understanding Convolutional Neural Network, and applying it to the handwritten recognition system.

3. IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP


What do they
THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations
They have a fear about to feed their personal information in the system
They feel insecure while making their bank transactions
used to reduce the manual work to understand the different styles of handwritten digits
Demand high recognition accuracy and lesser computational complexity
Ability to detect feature object without any human supervision
What do they
HEAR?
what friends say
what boss say
what influencers say
Helps to increase the data entry
They get clarity about handwritten digits
Ability to develop an efficient algorithm that can recognize hand written digit
Helps the users to enter the data captcha
What do they
SEE?
environment
friends
what the market offers
Extract most essential information from raw data
Used in automated bank cheque processing system
Potential future enhancement can be done in this field
Analyzed using support vector machine, random forest, multilayered perceptron
What do they
SAY AND DO?
attitude in public
appearance
behavior towards others
Its creates the curiosity between the people
No need to use external dependencies or devices , this process can done through our mobile phones.
It converts handwritten digits into machine readable formats
It will help the readers to easily understand the handwritten digits
Deficient works accomplished on Arabic pattern digits
Arabic digits are more challenging than English patterns
PAIN
fears
frustrations
obstacles
Not aware to clear understand about their handwritten styles
Various object needs to be trained in dataset to recognize different handwritten digit
The Handwritten digits are not Flawless
Every single user writes a digit in their own way which is difficult to understand by others
GAIN
"wants" / needs
measures of success
obstacles
They would get clear about the digits
It helps in bank check processing
They are effective in accuracies
High capability of hierarchical feature learning
It makes Error free and operations easier

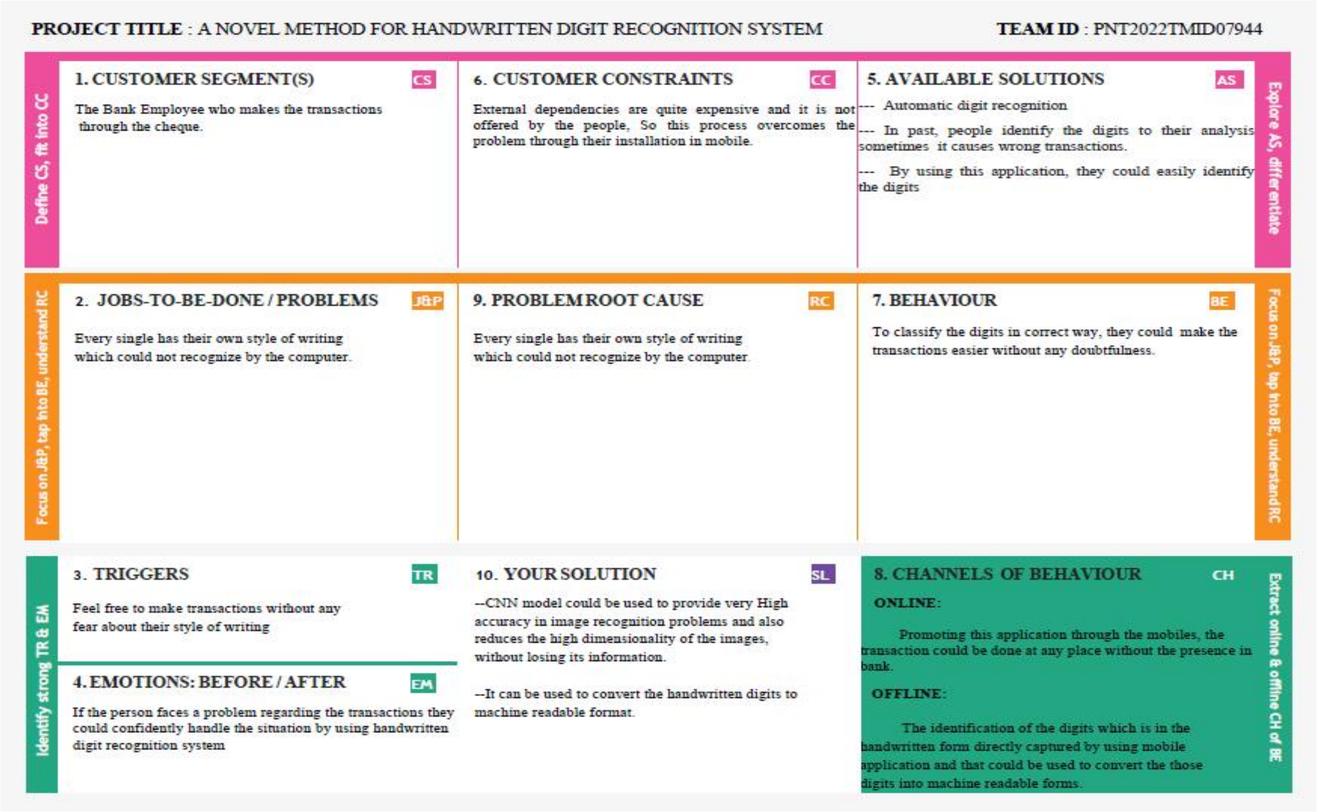
3.2 IDEATION & BRAINSTORMING



3.3 PROPOSED SOLUTION

- In the proposed solution, we use CNN algorithm to classify and further use to satisfy the needs of digits recognition.The model is trained by using MNIST dataset which holds 60,000 training samples and 10,000 test samples. The inputs are recognized by using CNN algorithm, where the CNN is used to convert the handwritten digits to machine readable format.
- CNN can be used to provide very High accuracy in image recognition problems.
- Reduces the high dimensionality of the images ,without losing its information.
- Little dependence on pre-processing ,decreasing the needs of human efforts developing functionalities.

3.4 PROBLEM SOLUTION FIT


Define CS, fit into CC
Explore AS, differentiate
Focus on J&P, tap into BE, understand RC
Focus on J&P, tap into BE, understand RC
Identify strong TR & EM
Extract online & offline CH of BE
CS
CC
AS
J&P
RC
BE
TR
EM
SL
CH

PROJECT TITLE : A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

TEAM ID : PNT2022TMID07944

1. CUSTOMER SEGMENT(S)

The Bank Employee who makes the transactions through the cheque.

6. CUSTOMER CONSTRAINTS

External dependencies are quite expensive and it is not offered by the people, So this process overcomes the problem through their installation in mobile.

5. AVAILABLE SOLUTIONS

--- Automatic digit recognition

--- In past, people identify the digits to their analysis sometimes it causes wrong transactions.

--- By using this application, they could easily identify the digits

2. JOBS-TO-BE-DONE / PROBLEMS

Every single has their own style of writing which could not recognize by the computer.

9. PROBLEM ROOT CAUSE

Every single has their own style of writing which could not recognize by the computer.

7. BEHAVIOUR

To classify the digits in correct way, they could make the transactions easier without any doubtfulness.

3. TRIGGERS

Feel free to make transactions without any fear about their style of writing

4. EMOTIONS: BEFORE / AFTER

If the person faces a problem regarding the transactions they could confidently handle the situation by using handwritten digit recognition system

10. YOUR SOLUTION

--CNN model could be used to provide very High accuracy in image recognition problems and also reduces the high dimensionality of the images, without losing its information.

--It can be used to convert the handwritten digits to machine readable format.

8. CHANNELS OF BEHAVIOUR

ONLINE:

Promoting this application through the mobiles, the transaction could be done at any place without the presence in bank.

OFFLINE:

The identification of the digits which is in the handwritten form directly captured by using mobile application and that could be used to convert the those digits into machine readable forms.

4 . REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

FR No.	**Functional Requirement (Epic)**	**Sub Requirement (Story / Sub-Task)**
FR-1	User Registration	● Registration through Web Application form ● Registration through Mobile Application form
FR-2	User Confirmation	● Confirmation via Message ● Confirmation via OTP
FR-3	Recognition Process	● The system should process the input given by the user only if it is an image file (JPG,PNG,etc.,) ● System should detect characters present in the image. ● System should retrieve characters present in the image and display them to the user.
FR-4	Display the error message	System shall show the error message to the user when the input given is not in the required format.

4.2 NON FUNCTIONAL REQUIREMENTS

FR No.	**Non-Functional Requirement**	**Description**
NFR-1	**Usability**	Handwritten character recognition is one of the practically important issues in pattern recognition applications. The applications of digit recognition include postal mail sorting, bank check processing, form data entry, etc.
NFR-2	**Security**	Provides clear ideas about handwritten digits which makes users feel free to do their activities.
NFR-3	**Reliability**	This software will work reliably for low resolution images and not for graphical images.
NFR-4	**Performance**	Handwritten characters in the input image will be recognized with an accuracy of about 98% and more.
NFR-5	**Availability**	This system will retrieve the handwritten text regions only if the image contains written text in it.

NFR-6	**Scalability**	Capability of developing the new algorithms and improve the existing algorithms is determined by the accuracy and speed factor for training and testing the models.

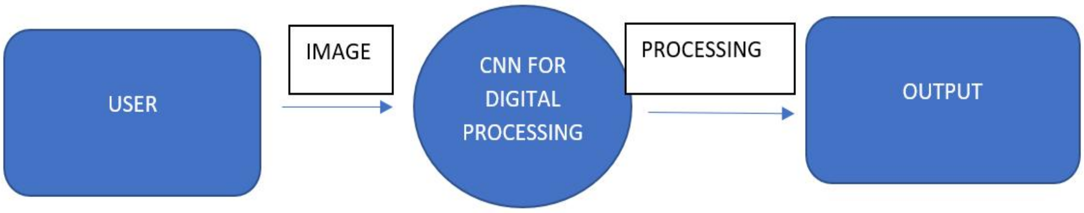
5.PROJECT DESIGN

5.1 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.

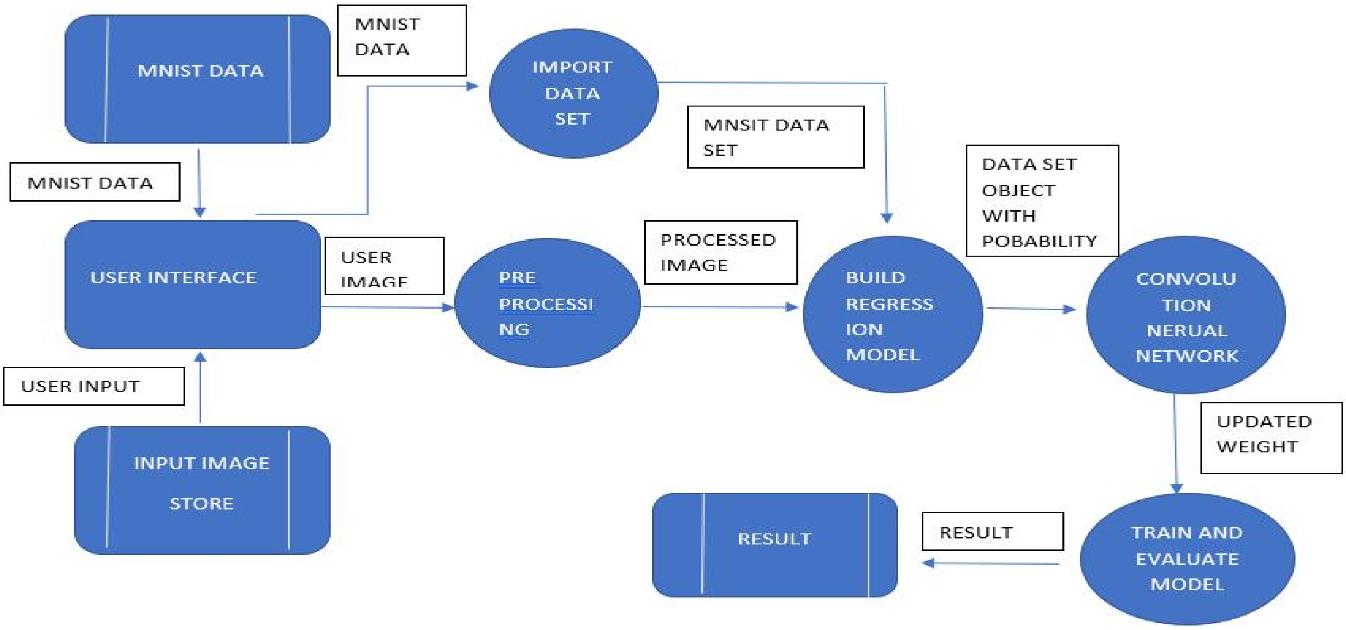
DFD Level-0

The DFD Level-0 consists of two external entities, the UI and the Output, along with a process, representing the CNN for DigitRecognition. Output is obtained after processing.



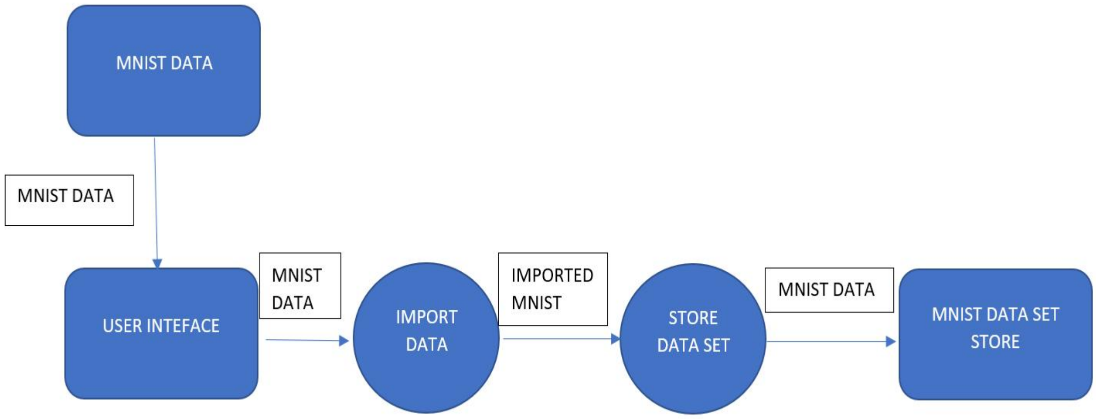
DFD Level-1

The DFD Level-1 consists of 2 external entities, the GUI and the Output, along with five process blocks and 2 data stores MNIST data and the Input image store, representing the internal workings of the CNN for Digit Recognition System. Process block imports MNIST data from library. Process block imports the image and process it and sends it to block where regression model is built. It sends objects with probabilities to CNN where weights are updated and multiple layers are built. Block trains and evaluates the model to generate output.



DFD Level-2

The DFD Level-2 for import data(figure 4) consists of two external data and one entity UI along with three process blocks, representing the three functionalities of the CNN for Digit Recognition System. It imports data from MNIST data stores and stores on the system.

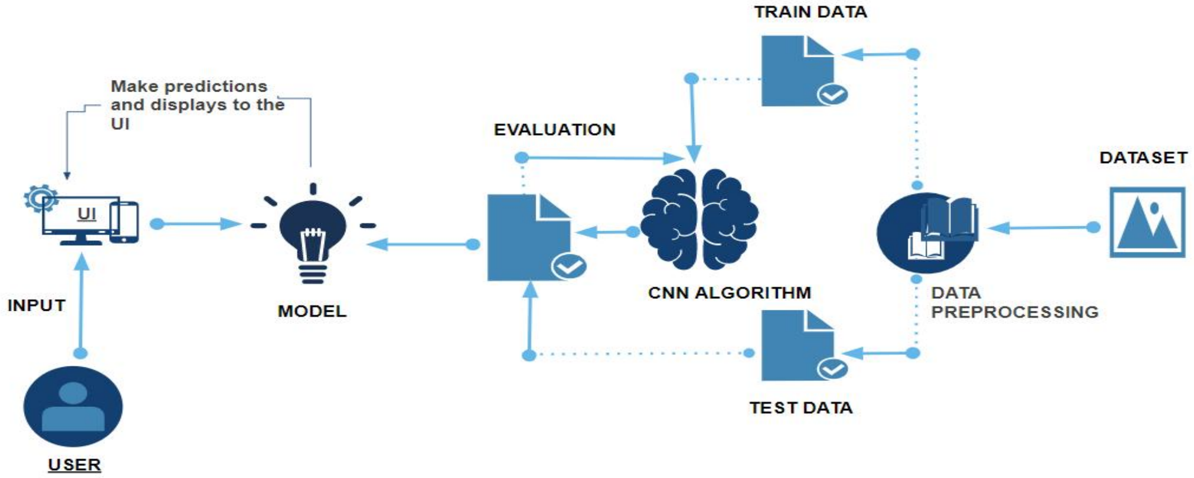


5.2 SOLUTION AND TECHNICAL ARCHITECTURE

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.
- Define features, development phases, and solution requirements.
- Provide specifications according to which the solution

is defined, managed, and delivered.



5.3 USER STORIES

User Type	Functional Requirement	User Story Number	User Story / Task	Acceptance Criteria	Priorit y	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	High	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	Mediu m	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	Mediu m	Sprint-3

6. PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNING AND ESTIMATION

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data Collection	USN-1	As a user, I can collect the dataset from various resources with different handwritings.	10	Low	Divyasri V Bharani Sri A
Sprint-1	Data Preprocessing	USN-2	As a user, I can load the dataset, handle the missing data, scaling and split data into train and test.	10	Medium	Divyasri V Bharani Sri A
Sprint-2	Model Building	USN-3	As a user, I will get an application with an ML model which provides high accuracy of recognized handwritten digit.	5	High	Divyasri V Bharani Sri A Jothisri S
Sprint-2	Add CNN layers	USN-4	Creating the model and adding the input, hidden, and output layers to it.	5	High	Divyasri V Bharani Sri A Jothisri S Charumathi M
Sprint-2	Train & test the model	USN-6	As a user, let us train our model with our image dataset.	6	Medium	Divyasri V Bharani Sri A Jothisri S Charumathi M

Sprint-2	Save the model	USN-7	As a user, the model is saved & integrated with an android application or web application in order to predict something.	2	Low	Bharani Sri A Jothisri S
Sprint-3	Building UI Application	USN-8	As a user, I will upload the handwritten digit image to the application by clicking an upload button.	5	High	Divyasri V Bharani Sri A Jothisri S Charumathi M
Sprint-3		USN-9	As a user, I can know the details of the fundamental usage of the application.	5	Low	Jothisri S Charumathi M
Sprint-3		USN-10	As a user, I can see the predicted / recognized digits in the application.	5	Medium	Divyasri V Bharani Sri A
Sprint-4	Train the model on IBM	USN-11	As a user, I train the model on IBM and integrate flask/Django with scoring end point.	10	High	Divyasri V Bharani Sri A Jothisri S Charumathi M
Sprint-4	Cloud Deployment	USN-12	As a user, I can access the web application and make the use of the product from anywhere.	10	High	Divyasri V Bharani Sri A Jothisri S Charumathi M

6.2 SPRINT DELIVERY SCHEDULE

Sprint	Total Story Points	Duration	Sprint Start	Date Sprint End (Planned)	Story Points completed(as on Planned 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.REPORTS FROM JIRA

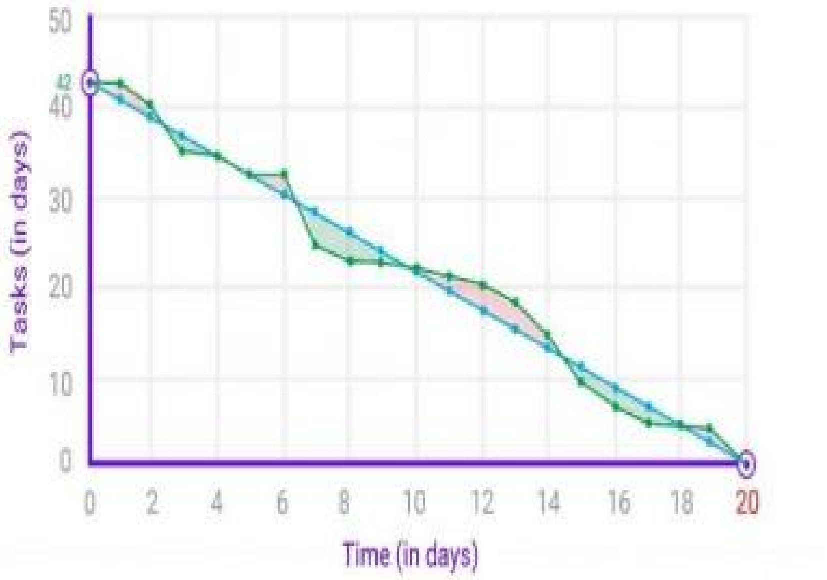
Velocity:

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:

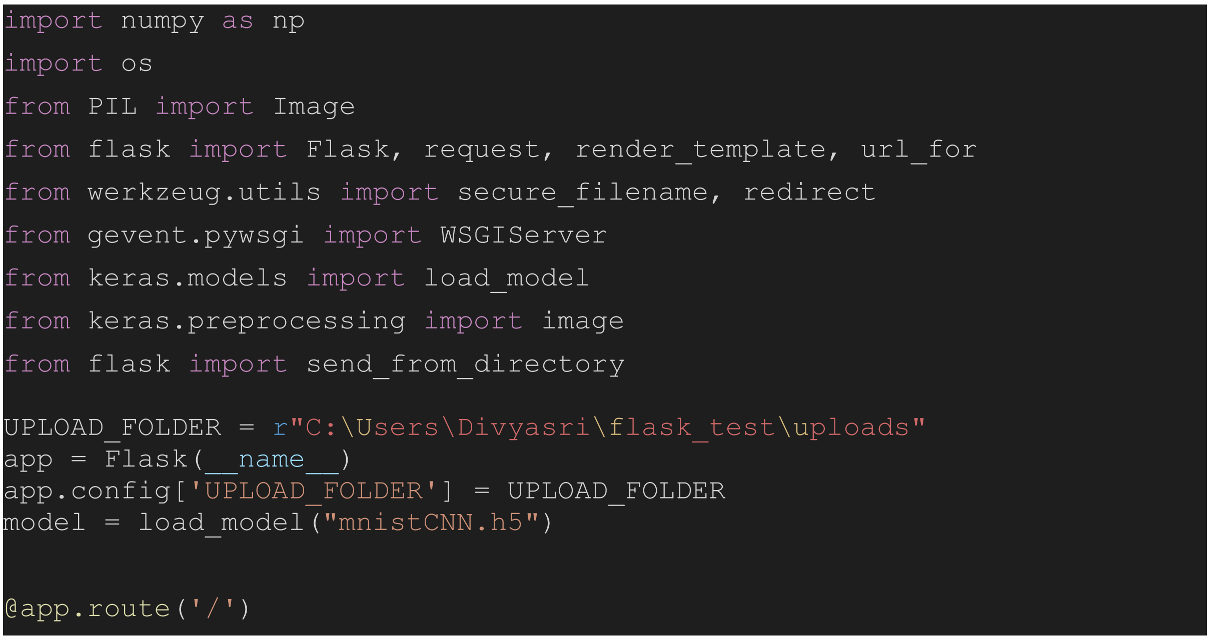
A burndown chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.



7 .CODING AND 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.



```
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 = r"C:\Users\Divyasri\flask_test\uploads"
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))  # 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 id="content">

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

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

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>

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

</scrip>
```

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

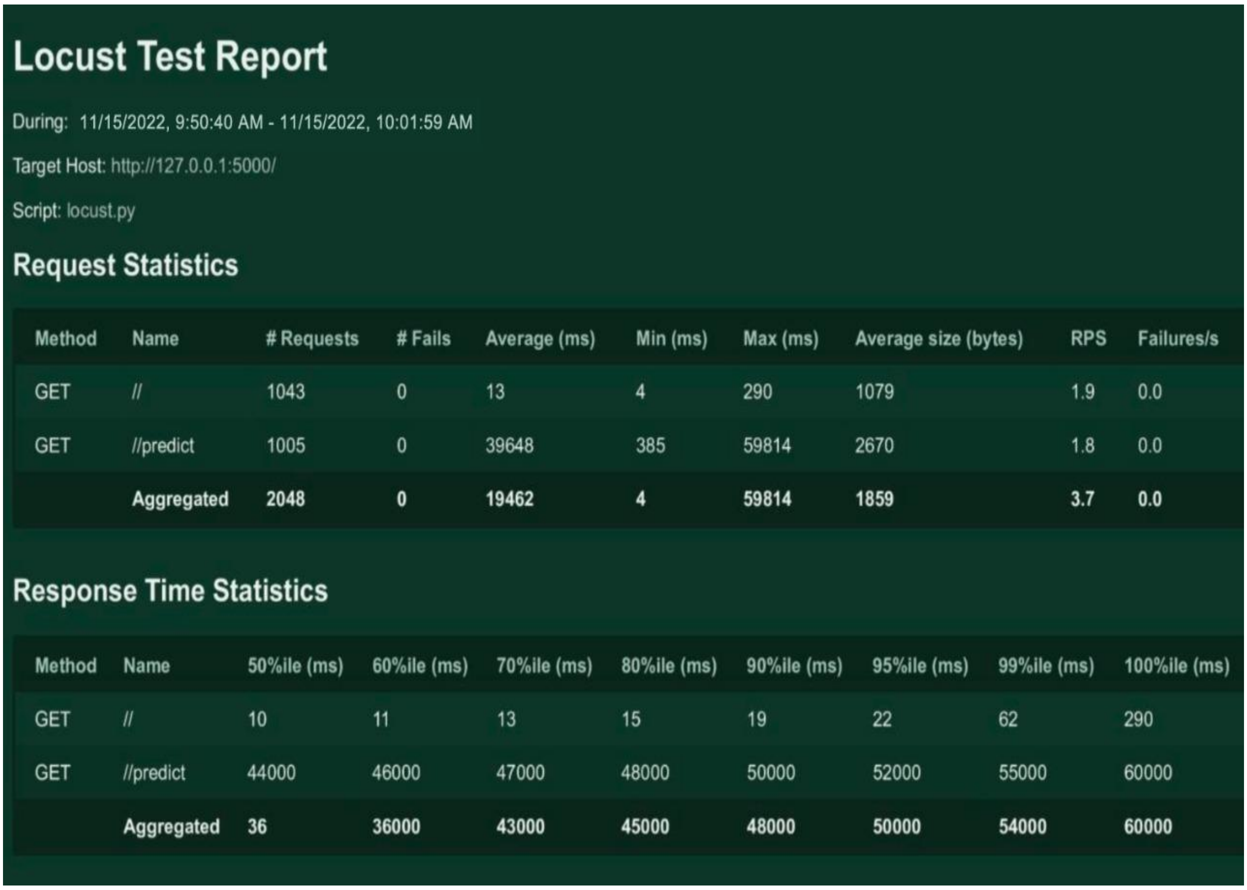
Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Total
By Design	1	0	1	0	2
Duplicate	0	0	0	0	0
External	0	0	2	0	2
Fixed	4	1	0	1	6
Not Reproduced	0	0	0	1	1
Skipped	0	0	0	1	1
Won’t Fix	1	0	1	0	2
Total	6	1	4	3	14

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

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/

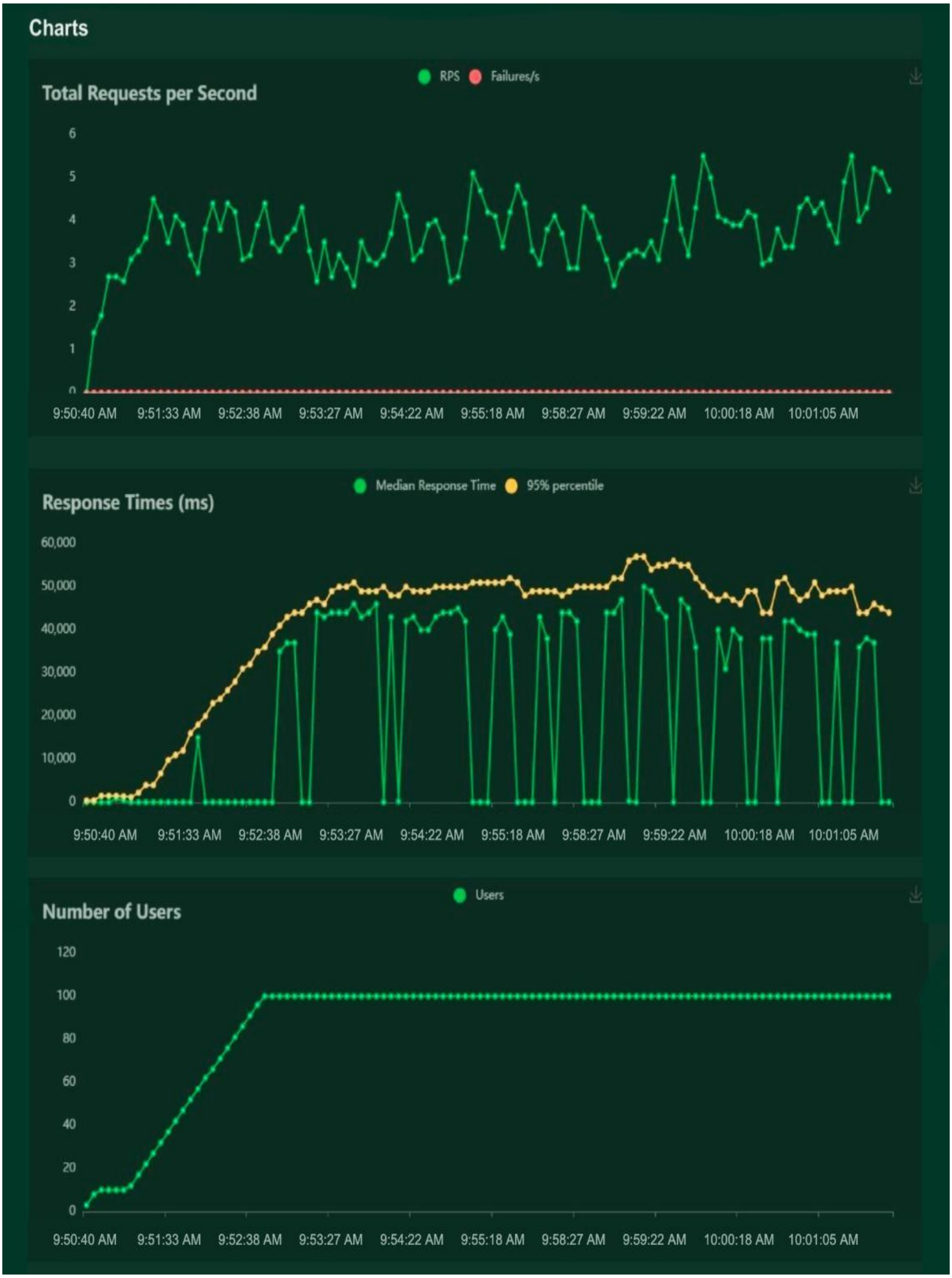
Script: locust.py

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

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.

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.

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.

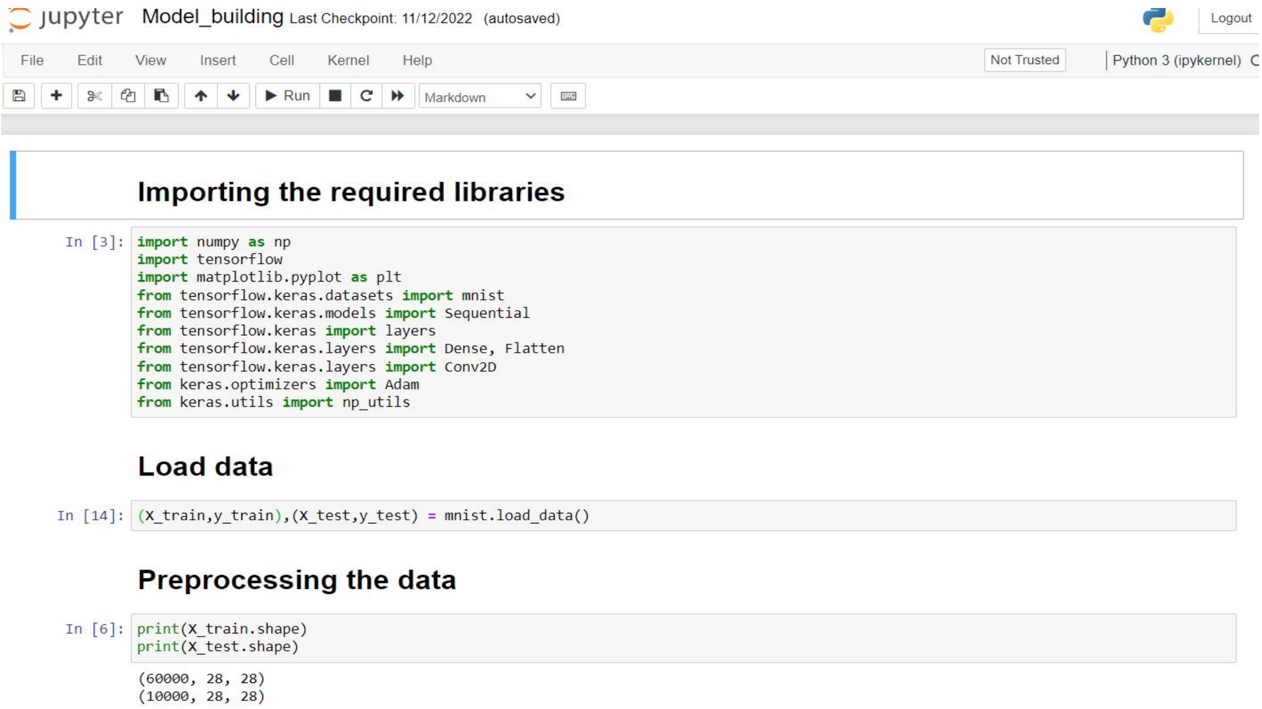
This project is far from complete and there is a lot of room for improvement. Some of the improvements that can be made to this project are as follows:

- 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.

13.SOURCE CODE

MODEL CREATION:



Importing the required libraries

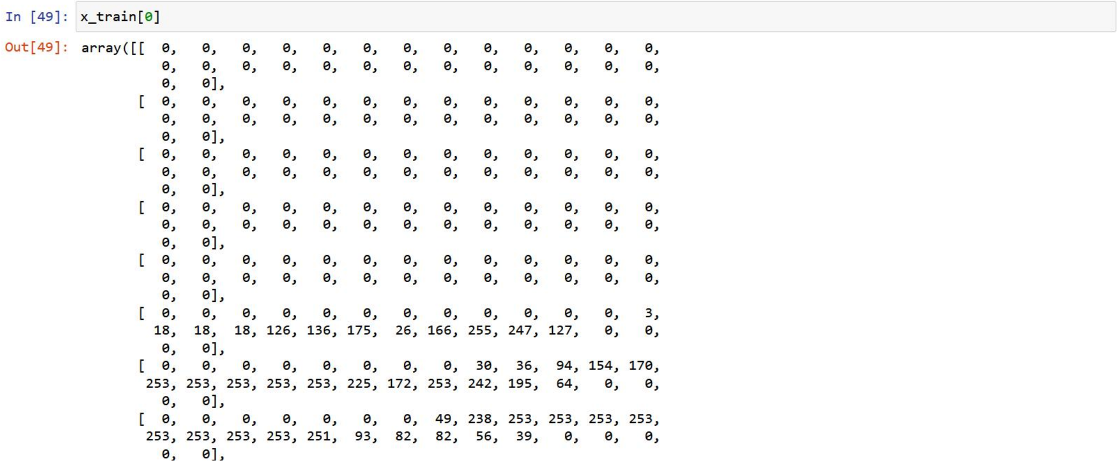
```
In [3]: import numpy as np
        import tensorflow
        import matplotlib.pyplot as plt
        from tensorflow.keras.datasets import mnist
        from tensorflow.keras.models import Sequential
        from tensorflow.keras import layers
        from tensorflow.keras.layers import Dense, Flatten
        from tensorflow.keras.layers import Conv2D
        from keras.optimizers import Adam
        from keras.utils import np_utils
```

Load data

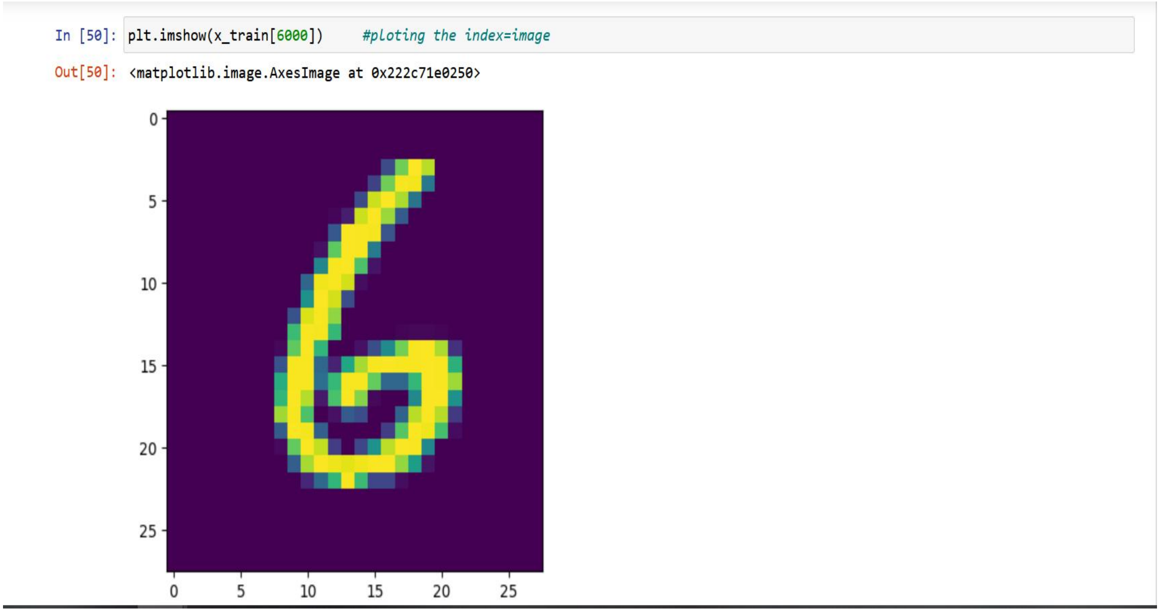
```
In [14]: (X_train,y_train),(X_test,y_test) = mnist.load_data()
```

Preprocessing the data

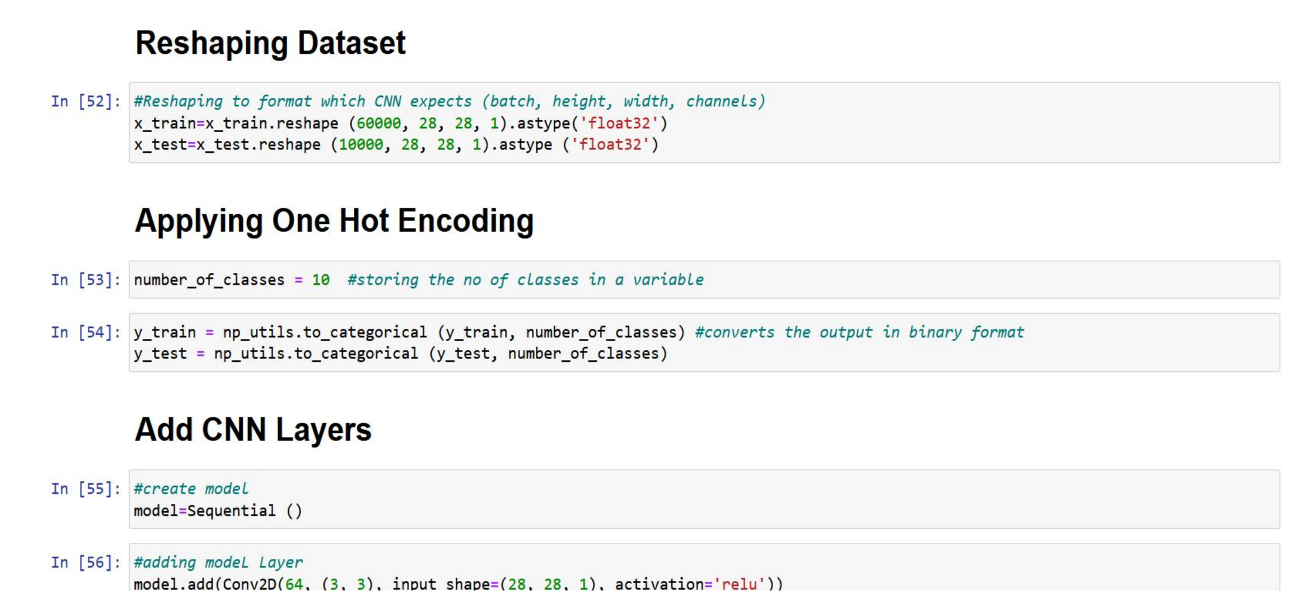
```
In [6]: print(X_train.shape)
        print(X_test.shape)

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



```
In [49]: x_train[0]

Out[49]: 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],
```



```
In [50]: plt.imshow(x_train[6000])     #ploting the index=image

Out[50]: <matplotlib.image.AxesImage at 0x222c71e0250>
```



Reshaping Dataset

```
In [52]: #Reshaping to format which CNN expects (batch, height, width, channels)
         x_train=x_train.reshape (60000, 28, 28, 1).astype('float32')
         x_test=x_test.reshape (10000, 28, 28, 1).astype ('float32')
```

Applying One Hot Encoding

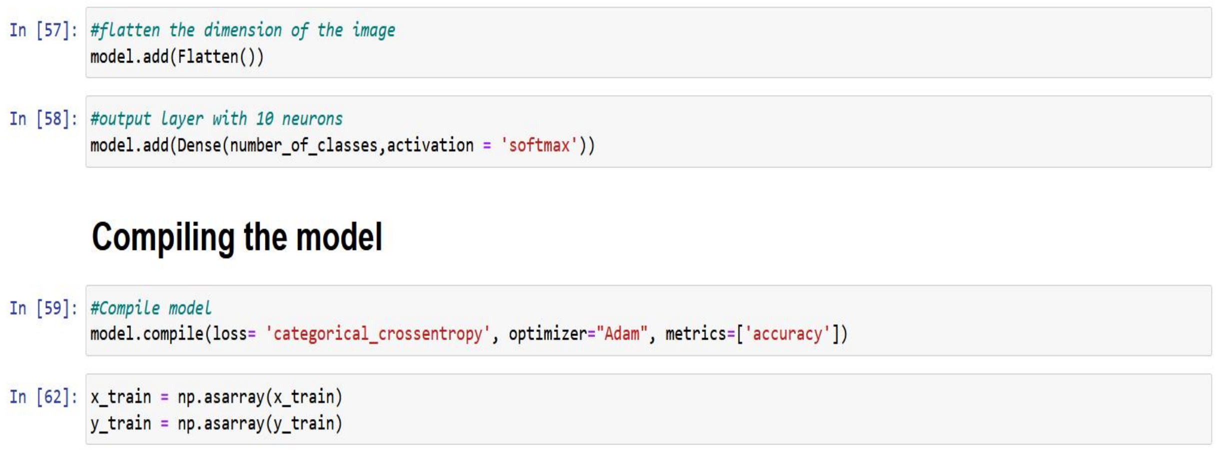
```
In [53]: number_of_classes = 10  #storing the no of classes in a variable
```

```
In [54]: y_train = np_utils.to_categorical (y_train, number_of_classes) #converts the output in binary format
         y_test = np_utils.to_categorical (y_test, number_of_classes)
```

Add CNN Layers

```
In [55]: #create model
         model=Sequential ()
```

```
In [56]: #adding model Layer
         model.add(Conv2D(64, (3, 3), input_shape=(28, 28, 1), activation='relu'))
```



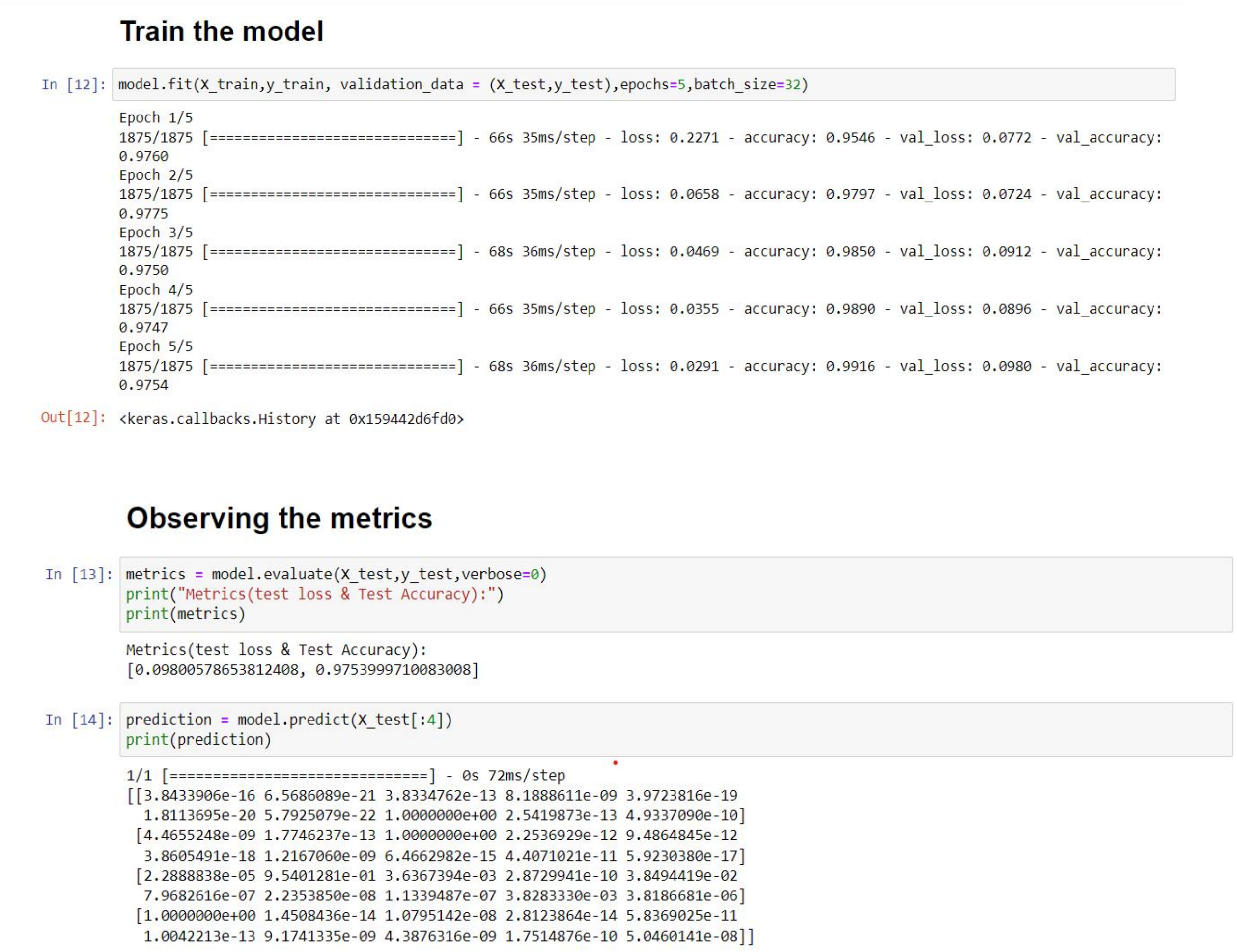
```
In [57]: #flatten the dimension of the image
         model.add(Flatten())
```

```
In [58]: #output layer with 10 neurons
         model.add(Dense(number_of_classes,activation = 'softmax'))
```

Compiling the model

```
In [59]: #Compile model
         model.compile(loss= 'categorical_crossentropy', optimizer="Adam", metrics=['accuracy'])
```

```
In [62]: x_train = np.asarray(x_train)
         y_train = np.asarray(y_train)
```



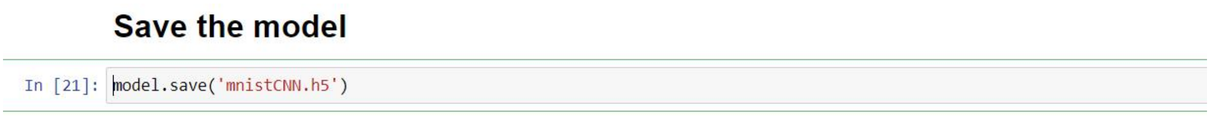
Train the model

```
In [12]: model.fit(X_train,y_train, validation_data = (X_test,y_test),epochs=5,batch_size=32)

         Epoch 1/5
         1875/1875 [==============================] - 66s 35ms/step - loss: 0.2271 - accuracy: 0.9546 - val_loss: 0.0772 - val_accuracy:
         0.9760
         Epoch 2/5
         1875/1875 [==============================] - 66s 35ms/step - loss: 0.0658 - accuracy: 0.9797 - val_loss: 0.0724 - val_accuracy:
         0.9775
         Epoch 3/5
         1875/1875 [==============================] - 68s 36ms/step - loss: 0.0469 - accuracy: 0.9850 - val_loss: 0.0912 - val_accuracy:
         0.9750
         Epoch 4/5
         1875/1875 [==============================] - 66s 35ms/step - loss: 0.0355 - accuracy: 0.9890 - val_loss: 0.0896 - val_accuracy:
         0.9747
         Epoch 5/5
         1875/1875 [==============================] - 68s 36ms/step - loss: 0.0291 - accuracy: 0.9916 - val_loss: 0.0980 - val_accuracy:
         0.9754

Out[12]: <keras.callbacks.History at 0x159442d6fd0>
```

Observing the metrics

```
In [13]: metrics = model.evaluate(X_test,y_test,verbose=0)
         print("Metrics(test loss & Test Accuracy):")
         print(metrics)

         Metrics(test loss & Test Accuracy):
         [0.09800578653812408, 0.9753999710083008]
```

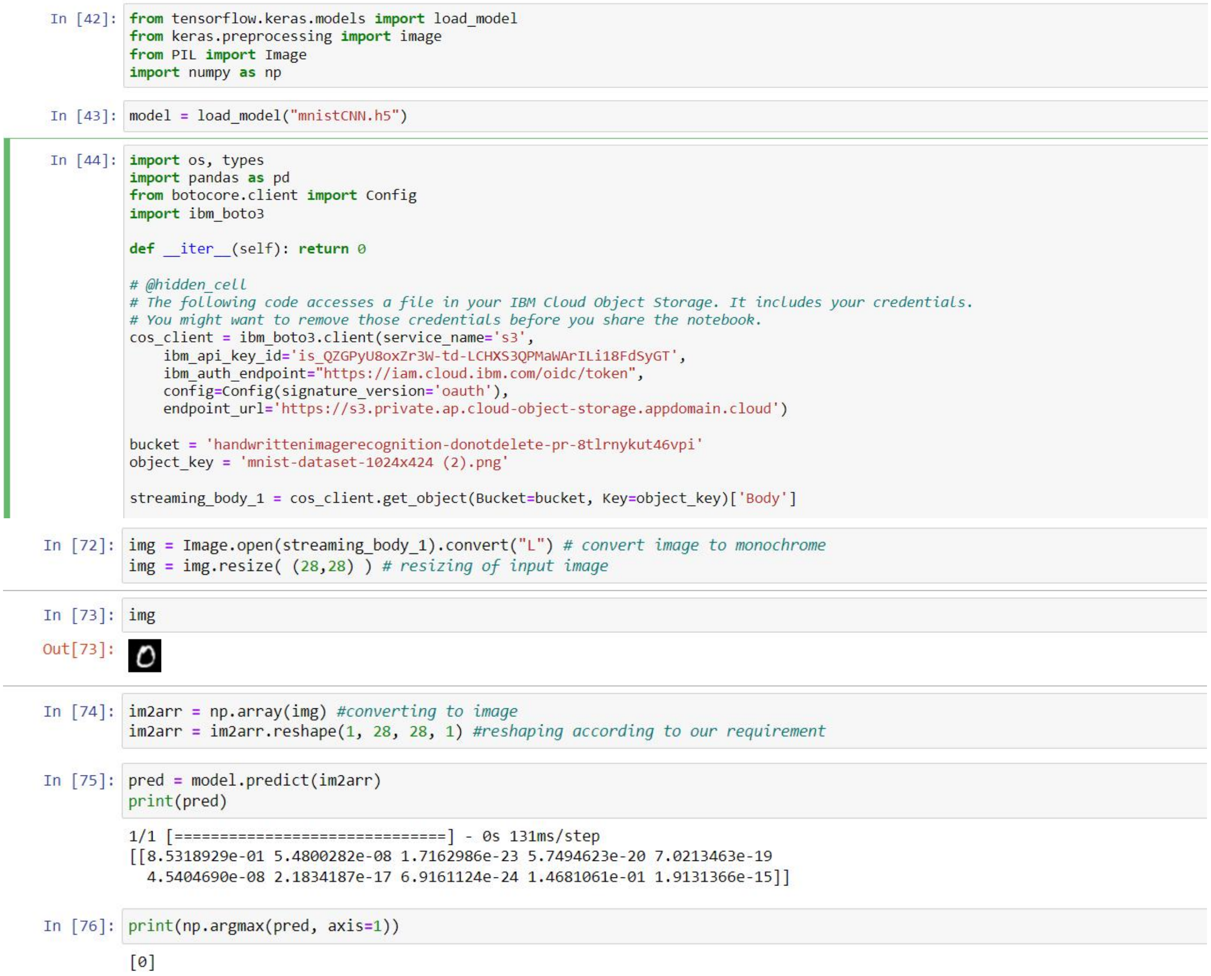
```
In [14]: prediction = model.predict(X_test[:4])
         print(prediction)

         1/1 [==============================] - 0s 72ms/step
         [[3.8433906e-16 6.5686089e-21 3.8334762e-13 8.1888611e-09 3.9723816e-19
           1.8113695e-20 5.7925079e-22 1.0000000e+00 2.5419873e-13 4.9337090e-10]
          [4.4655248e-09 1.7746237e-13 1.0000000e+00 2.2536929e-12 9.4864845e-12
           3.8605491e-18 1.2167060e-09 6.4662982e-15 4.4071021e-11 5.9230380e-17]
          [2.2888838e-05 9.5401281e-01 3.6367394e-03 2.8729941e-10 3.8494419e-02
           7.9682616e-07 2.2353850e-08 1.1339487e-07 3.8283330e-03 3.8186681e-06]
          [1.0000000e+00 1.4508436e-14 1.0795142e-08 2.8123864e-14 5.8369025e-11
           1.0042213e-13 9.1741335e-09 4.3876316e-09 1.7514876e-10 5.0460141e-08]]
```



Save the model

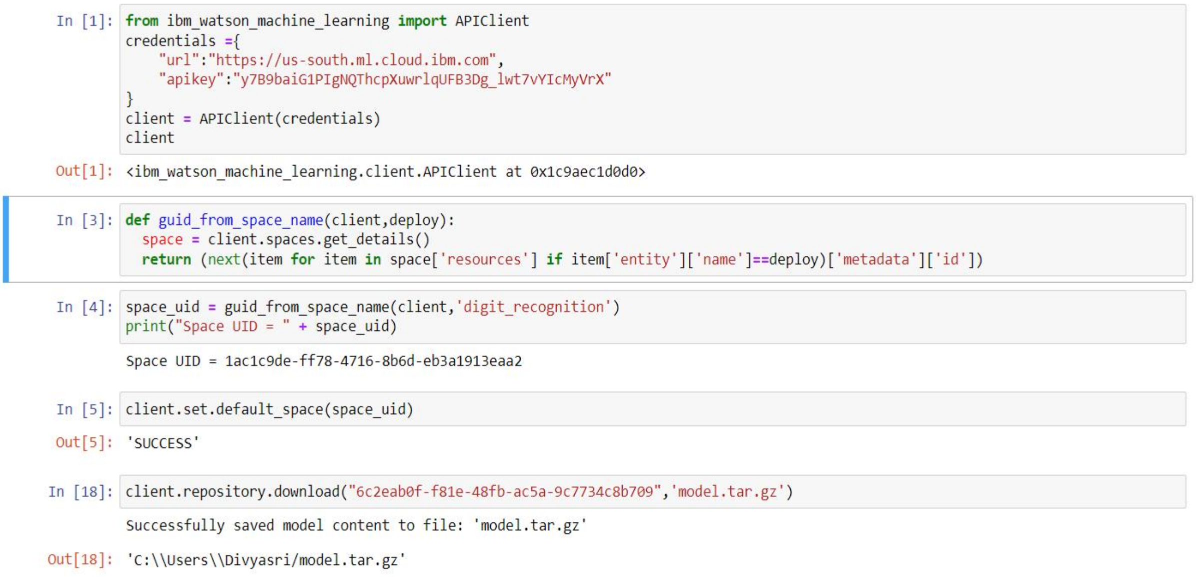
```
In [21]: model.save('mnistCNN.h5')
```

CNN_PREDICTION:



```
In [42]: from tensorflow.keras.models import load_model
         from keras.preprocessing import image
         from PIL import Image
         import numpy as np

In [43]: model = load_model("mnistCNN.h5")

In [44]: 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='is_QZGPyU8oxZr3W-td-LCHXS3QPMaWArILi18FdSyGT',
             ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
             config=Config(signature_version='oauth'),
             endpoint_url='https://s3.private.ap.cloud-object-storage.appdomain.cloud')

         bucket = 'handwrittenimagerecognition-donotdelete-pr-8tlrnykut46vpi'
         object_key = 'mnist-dataset-1024x424 (2).png'

         streaming_body_1 = cos_client.get_object(Bucket=bucket, Key=object_key)['Body']

In [72]: img = Image.open(streaming_body_1).convert("L") # convert image to monochrome
         img = img.resize( (28,28) ) # resizing of input image

In [73]: img

Out[73]:

In [74]: im2arr = np.array(img) #converting to image
         im2arr = im2arr.reshape(1, 28, 28, 1) #reshaping according to our requirement

In [75]: pred = model.predict(im2arr)
         print(pred)

         1/1 [==============================] - 0s 131ms/step
         [[8.5318929e-01 5.4800282e-08 1.7162986e-23 5.7494623e-20 7.0213463e-19
           4.5404690e-08 2.1834187e-17 6.9161124e-24 1.4681061e-01 1.9131366e-15]]

In [76]: print(np.argmax(pred, axis=1))

         [0]
```

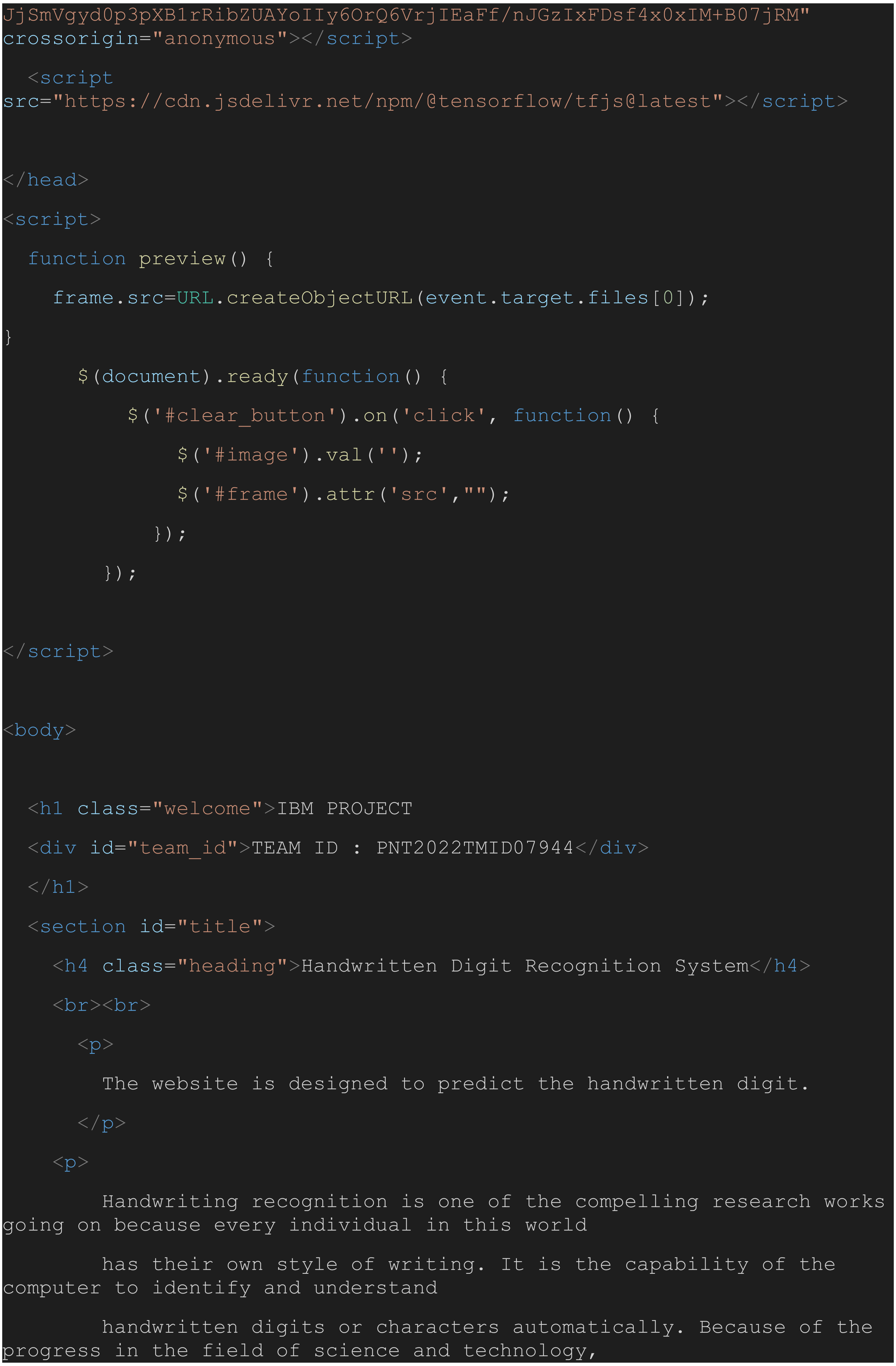
TRAIN THE MODEL ON IBM:



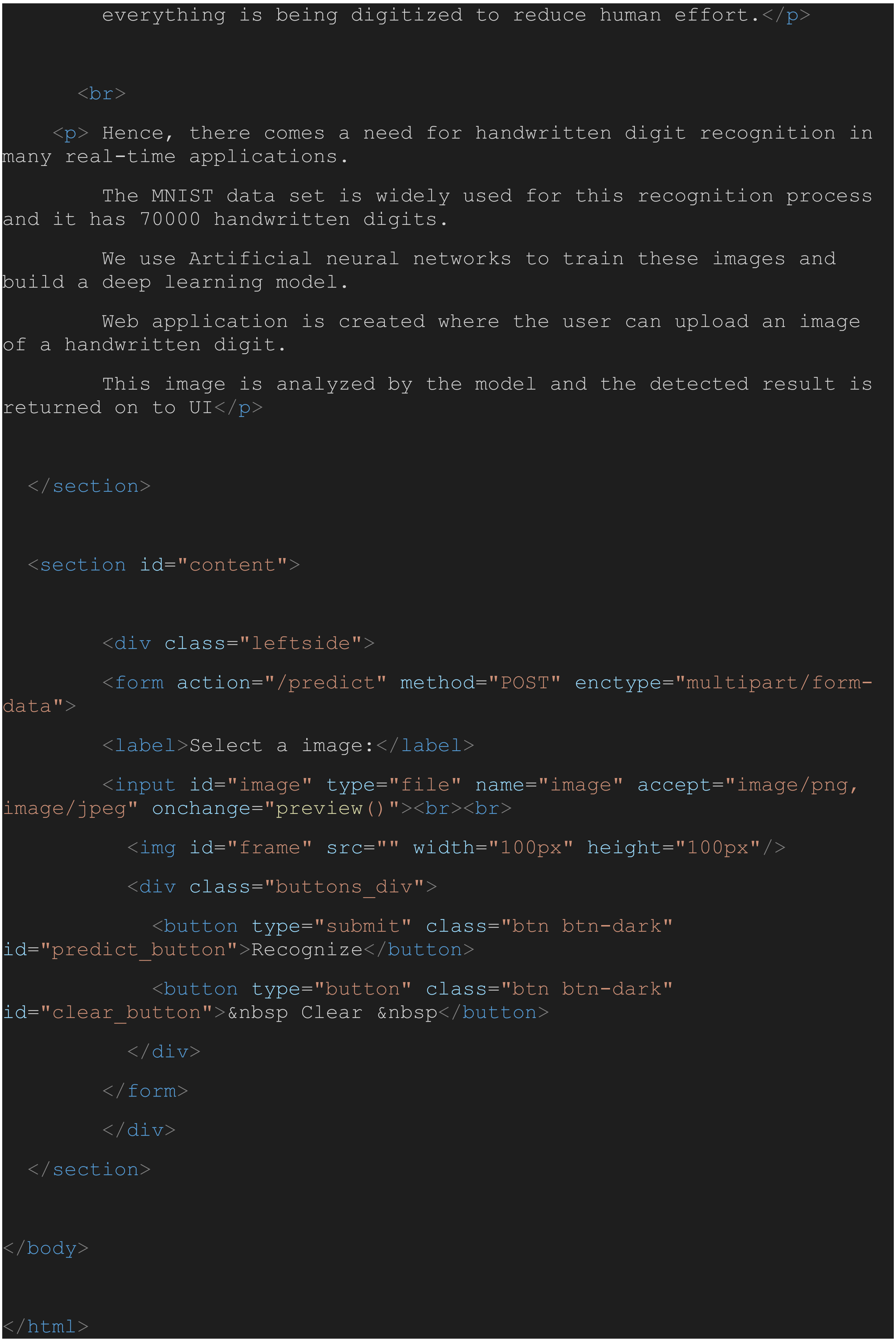
```
In [1]: from ibm_watson_machine_learning import APIClient
        credentials ={
            "url":"https://us-south.ml.cloud.ibm.com",
            "apikey":"y7B9baiG1PIgNQThcpXuwrlqUFB3Dg_lwt7vYIcMyVrX"
        }
        client = APIClient(credentials)
        client

Out[1]: <ibm_watson_machine_learning.client.APIClient at 0x1c9aec1d0d0>

In [3]: 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 [4]: space_uid = guid_from_space_name(client,'digit_recognition')
        print("Space UID = " + space_uid)

        Space UID = 1ac1c9de-ff78-4716-8b6d-eb3a1913eaa2

In [5]: client.set.default_space(space_uid)

Out[5]: 'SUCCESS'

In [18]: client.repository.download("6c2eab0f-f81e-48fb-ac5a-9c7734c8b709",'model.tar.gz')

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

Out[18]: 'C:\\Users\\Divyasri/model.tar.gz'
```

HOME PAGE(HTML) – index.html

```
<html>

<head>

  <title>Digit Recognition WebApp</title>

   <meta name="viewport" content="width=device-width">

  <!-- GoogleFont -->

  <link
href="https://fonts.googleapis.com/css2?family=Prompt:wght@600&display=swap" rel="stylesheet">

  <link
href="https://fonts.googleapis.com/css2?family=Varela+Round&display=swap" rel="stylesheet">

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

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

  <!-- bootstrap -->

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

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

  <!-- fontawesome -->

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



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

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

  <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js" integrity="sha384-
```

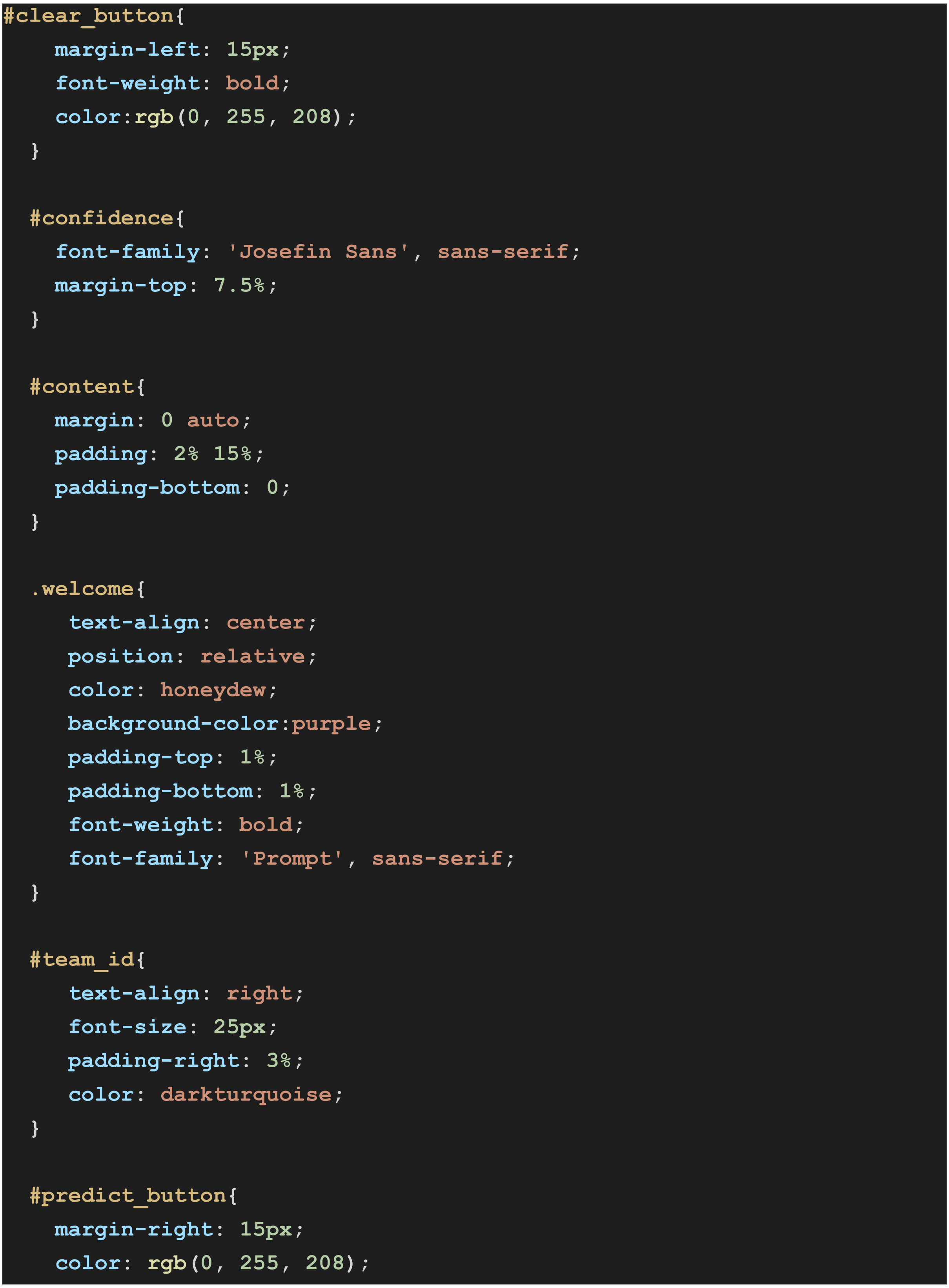


```
JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM"
crossorigin="anonymous"></script>

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


</head>

<script>

  function preview() {

    frame.src=URL.createObjectURL(event.target.files[0]);

}

      $(document).ready(function() {

          $('#clear_button').on('click', function() {

              $('#image').val('');

              $('#frame').attr('src',"");

            });

        });


</script>


<body>


  <h1 class="welcome">IBM PROJECT

  <div id="team_id">TEAM ID : PNT2022TMID07944</div>

  </h1>

  <section id="title">

    <h4 class="heading">Handwritten Digit Recognition System</h4>

    <br><br>

      <p>

        The website is designed to predict the handwritten digit.

      </p>

    <p>

        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 digitized to reduce human effort.</p>


      <br>

    <p> Hence, there comes a need for handwritten digit recognition in
many real-time applications.

        The 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 analyzed by the model and the detected result is
returned on to UI</p>


  </section>


  <section id="content">


        <div class="leftside">

        <form action="/predict" method="POST" enctype="multipart/form-
data">

        <label>Select a image:</label>

        <input id="image" type="file" name="image" accept="image/png,
image/jpeg" onchange="preview()"><br><br>

          <img id="frame" src="" width="100px" height="100px"/>

          <div class="buttons_div">

            <button type="submit" class="btn btn-dark"
id="predict_button">Recognize</button>

            <button type="button" class="btn btn-dark"
id="clear_button">&nbsp Clear &nbsp</button>

          </div>

        </form>

        </div>

  </section>


</body>


</html>
```

HOME PAGE(CSS) – style.css



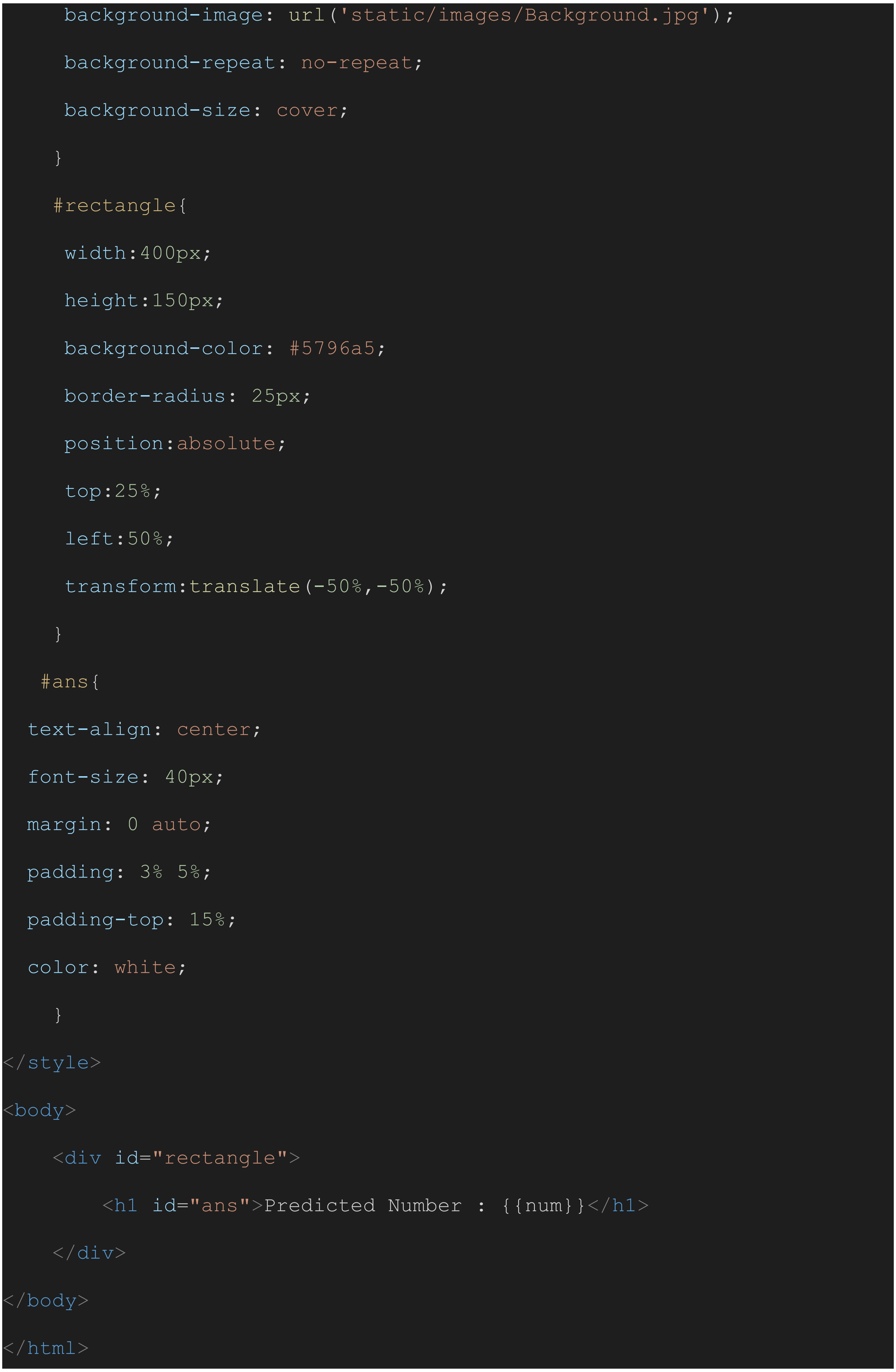
```
#clear_button{
    margin-left: 15px;
    font-weight: bold;
    color:rgb(0, 255, 208);
  }

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

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

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

  #team_id{
     text-align: right;
     font-size: 25px;
     padding-right: 3%;
     color: darkturquoise;
  }

  #predict_button{
    margin-right: 15px;
    color: rgb(0, 255, 208);
```

```css
  font-weight: bold;
}

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

#result{
  font-size: 5rem;
}

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

.btn {
    font-size: 15px;
    padding: 10px;
    -webkit-appearance: none;
    background: #eee;
    border: 1px solid #888;
    margin-top: 20px;
    margin-bottom: 20px;
}

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

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

.leftside{
  text-align: center;
  margin: 0 auto;
```

```
    margin-top: 2%;
    /* padding-left: 10%; */
  }

  #frame{
    margin-right: 10%;
  }

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

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

  @media (min-width: 720px) {
    .leftside{
      padding-left: 10%;
    }
  }
```

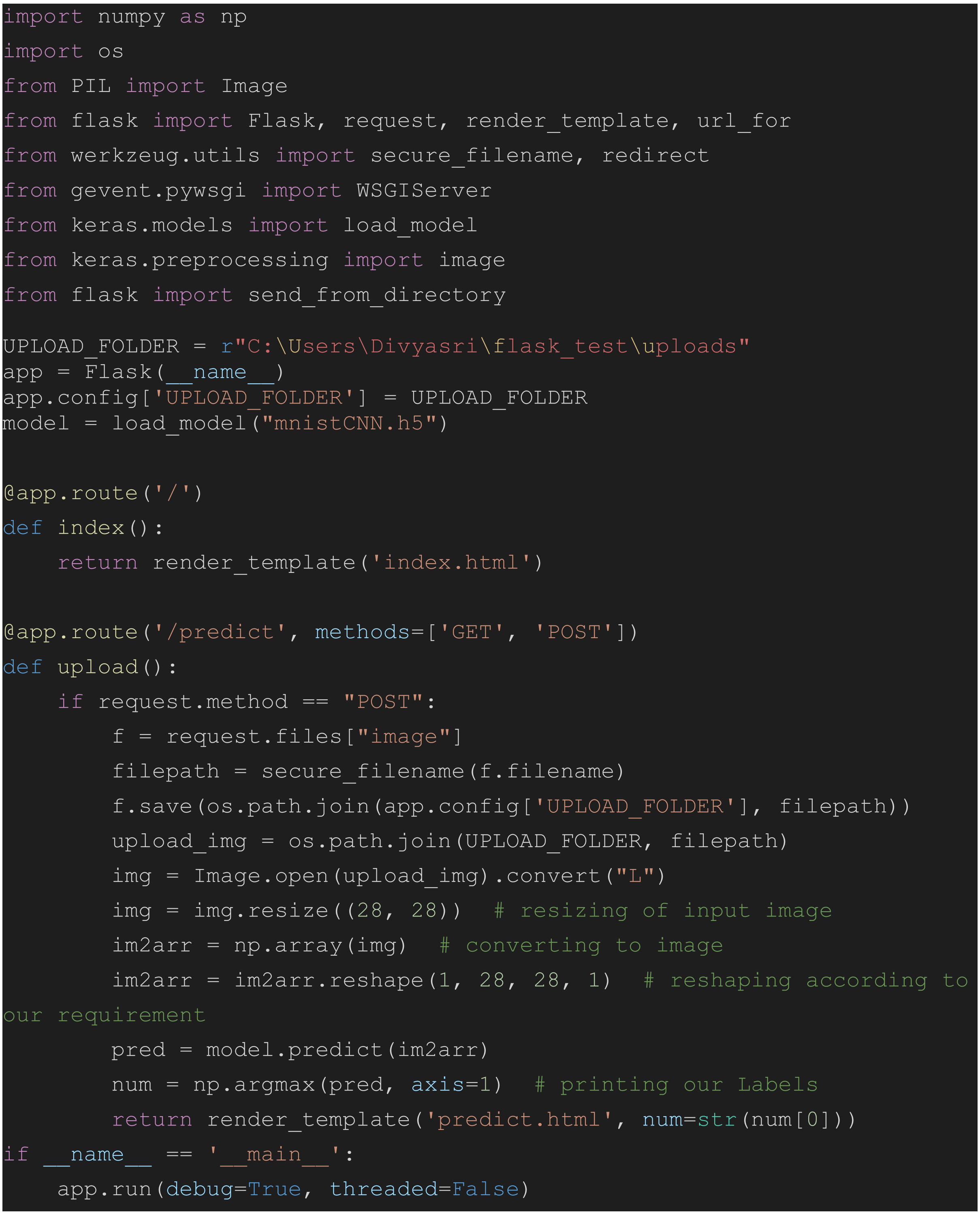
PREDICT PAGE (HTML) – predict.html

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>Prediction</title>

</head>

<style>

    body{
```



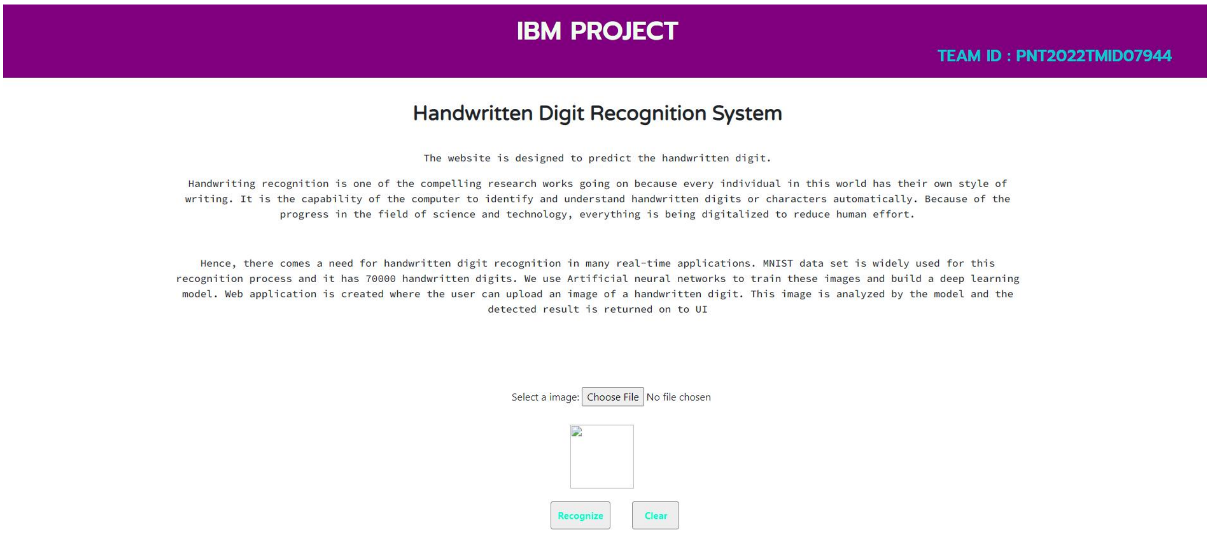
```
     background-image: url('static/images/Background.jpg');

     background-repeat: no-repeat;

     background-size: cover;

    }

    #rectangle{

     width:400px;

     height:150px;

     background-color: #5796a5;

     border-radius: 25px;

     position:absolute;

     top:25%;

     left:50%;

     transform:translate(-50%,-50%);

    }

   #ans{

  text-align: center;

  font-size: 40px;

  margin: 0 auto;

  padding: 3% 5%;

  padding-top: 15%;

  color: white;

    }

</style>

<body>

    <div id="rectangle">

        <h1 id="ans">Predicted Number : {{num}}</h1>

    </div>

</body>

</html>
```

FLASK APP - app.py



```
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 = r"C:\Users\Divyasri\flask_test\uploads"
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))  # 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)
```

SCREENSHOTS:



IBM PROJECT

TEAM ID : PNT2022TMID07944

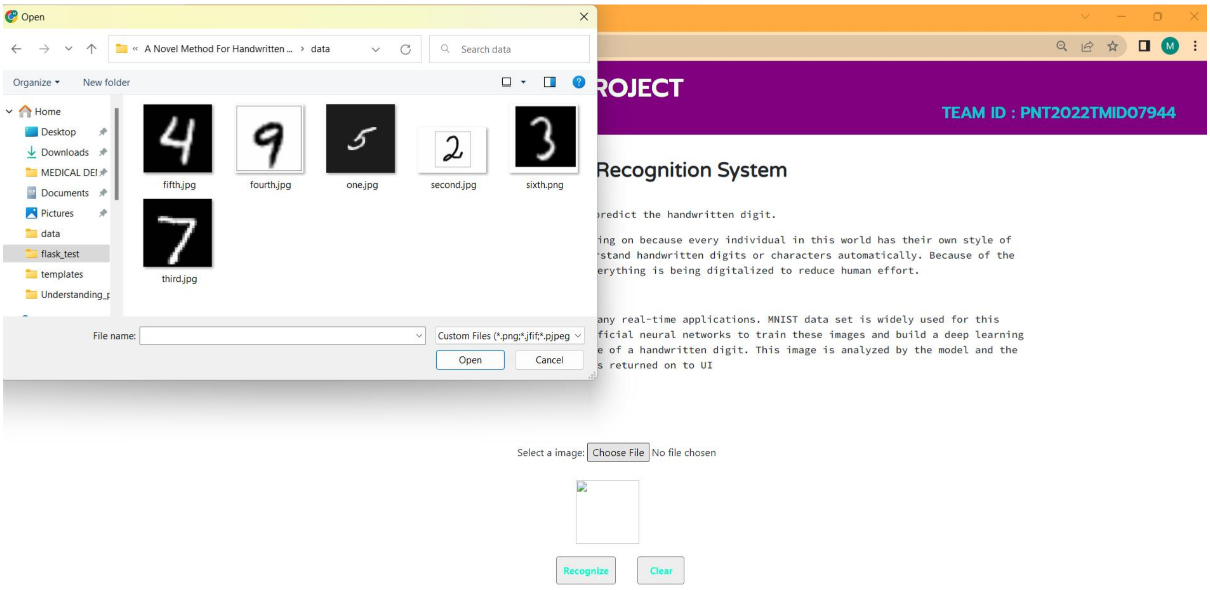
Handwritten Digit Recognition System

The website is designed to predict the handwritten digit.

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 analyzed by the model and the detected result is returned on to UI

Select a image: Choose File No file chosen



ROJECT

TEAM ID : PNT2022TMID07944

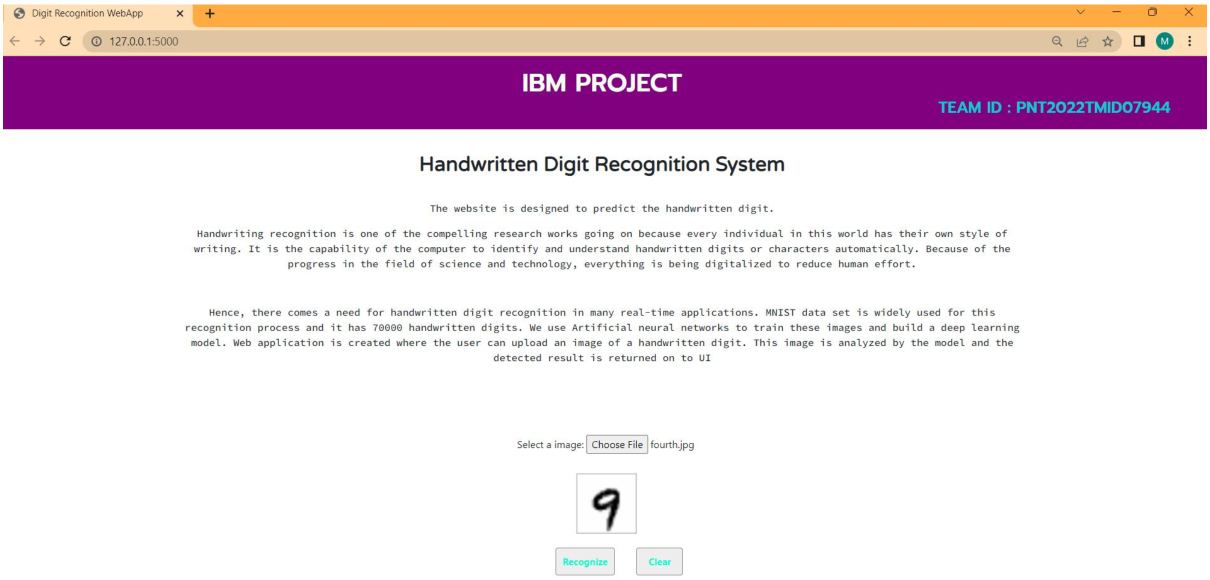
Recognition System

predict the handwritten digit.

ing on because every individual in this world has their own style of
stand handwritten digits or characters automatically. Because of the
erything is being digitalized to reduce human effort.

any real-time applications. MNIST data set is widely used for this
ficial neural networks to train these images and build a deep learning
e of a handwritten digit. This image is analyzed by the model and the
s returned on to UI

Select a image: Choose File No file chosen



IBM PROJECT

TEAM ID : PNT2022TMID07944

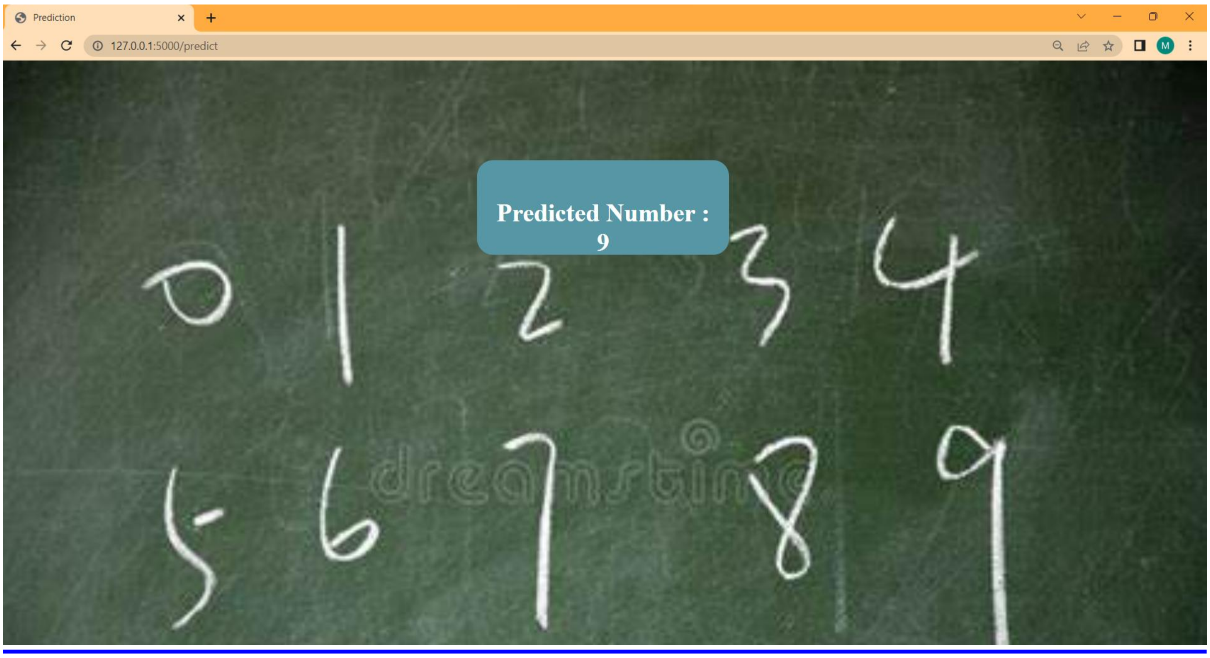
Handwritten Digit Recognition System

The website is designed to predict the handwritten digit.

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 analyzed by the model and the detected result is returned on to UI

Select a image: Choose File fourth.jpg



Predicted Number : 9

GITHUB LINK:

https://github.com/IBM-EPBL/IBM-Project-8316-1664169729

PROJECT DEMO LINK:

https://drive.google.com/file/d/1xOTNue0h7v-w-WE45vlgoLPIXll_0Ycl/view?usp=sharing