PROJECT REPORT

A NOVEL METHOD FOR

HANDWRITTEN DIGIT RECOGNITION

submitted by

PNT2022TMID26315

Abhishek S	-	211719104001
Abinesh R	-	211719104002
Aneerudh V	-	211719104016
Ashwin Adhithya K	-	211719104022

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

1.1 PROJECT OVERVIEW

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

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

1.2 PURPOSE

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

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. 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

Title:Handwritten English Character and Digit Recognition(2021)

Author:Al-Mahmud; Asnuva Tanvin; Sazia Rahman

one of the most sought-after technologies is a handwritten character recognition system. It has the potential to solve a wide range of issues and bring about radical change in our lives. We used Convolutional Neural Networks (CNNs) to recognize handwritten English capital letters and digits in this research. We improved a previously developed CNN architecture by adjusting hyperparameters and minimizing the model's overfitting. The MNIST digit dataset is used to evaluate the experiments, which are then compared to different methods. On the MNIST dataset, 99.47 percent test accuracy was attained,which is superior to other approaches. The research was then expanded upon by the addition of a new dataset for recognizing English capital letters. 98.94 percent accuracy was achieved on this extended dataset.

Title**: Image Classification using Deep Learning: An Experimental Study on Handwritten Digit Recognition**

Author: Mukesh Kumar Rohil; Raju Singh

This paper presents an experimental study of the use of Deep Learning using Convolution Neural Networks (CNNs) for Image Classification. Specially, the problem being addressed here is of recognition of handwritten digits. The objective is to report variations in testing errors and accuracies with varying kernel size and varying number of feature maps. We performed handwritten digit classification using neural network and deep learning for a subset from the MNIST dataset, which contains 60,000 training images and 10,000 test images in all. It is observed that the accuracy and loss are stabilizing with minor change in the kernel size and the number of feature maps.

Title:**Real Time Handwritten Digits Recognition Using Convolutional Neural Network.**
Author:Kaveti Upender; Venkata Siva Kumar Pasupuleti.

Reading handwritten information like examination answer sheets is still a difficult task for many of us, because each one of us is having a different interpretation style. As the world is moving towards digitization, converting the handwritten information to a readable digital format reduces the difficulty. This approach will be beneficial for the readers as it gives a better understanding of the information. With the help of machine learning and deep learning algorithms, the handwritten patterns can be recognized and classify them accordingly to a digital format with human level accuracy. This research paper deals with predicting the real time handwritten digits only. To classify the handwritten digits MNIST data set is used for training the model. OpenCV python library is used for detecting the patterns in the real time handwritten digits. These detected patterns are predicted to human level accuracy with the help of a Convolutional Neural Network model.

Title: An Efficient And Improved Scheme For Handwritten Digit Recognition Based On Convolutional Neural Network (2019)

Author: Ali, Saqib and Shaukat, Zeeshan and Azeem, Muhammad and Sakhawat, Zareen and Mahmood, Tariq   and others

This study uses rectified linear units (ReLU) activation and a convolutional neural network (CNN) that incorporates the Deeplearning4j (DL4J) architecture to recognize handwritten digits. The proposed CNN framework has all the necessary parameters for a high level of MNIST digit classification accuracy. The system's training takes into account the time factor as well. The system is also tested by altering the number of CNN layers for additional accuracy verification. It is important to note that the CNN architecture consists of two convolutional layers, the first with 32 filters and a 5x5 window size and the second with 64 filters and a 7x7 window size. In comparison to earlier proposed systems, the experimental findings show that the proposed CNN architecture for the MNIST dataset demonstrates great performance in terms of time and accuracy. As a result, handwritten numbers are detected with a recognition rate of 99.89% and high precision (99.21%) in a short amount of time.

Title: Improved Handwritten Digit Recognition Using Quantum K-Nearest Neighbor Algorithm (2019)

Author: Wang, Yuxiang and Wang, Ruijin and Li, Dongfen and Adu-Gyamfi, Daniel and Tian, Kaibin and Zhu, Yixin

The KNN classical machine learning technique is used in this research to enable quantum parallel computing and superposition. They used the KNN algorithm with quantum acceleration to enhance handwritten digit recognition. When dealing with more complicated and sizable handwritten digital data sets, their suggested method considerably lowered the computational time complexity of the traditional KNN algorithm. The paper offered a theoretical investigation of how quantum concepts can be applied to machine learning. Finally, they established a fundamental operational concept and procedure for machine learning with quantum acceleration.

Title:Handwritten Digit Recognition Using Machine And Deep Learning Algorithms (2021)

Author: Pashine, Samay and Dixit, Ritik and Kushwah, Rishi*ka*

In this study, they developed three deep and machine learning-based models for handwritten digit recognition using MNIST datasets. To determine which model was the most accurate, they compared them based on their individual properties.Support vector machines are among the simplest classifiers, making them faster than other algorithms and providing the highest training accuracy rate in this situation. However, due to their simplicity, SVMs cannot categorize complicated and ambiguous images as accurately as MLP and CNN algorithms can. In their research, they discovered that CNN produced the most precise outcomes for handwritten digit recognition. This led them to the conclusion that CNN is the most effective

solution for all types of prediction issues, including those using picture data. Next, by comparing the execution times of the algorithms, they determined that increasing the number of epochs without changing the configuration of the algorithm is pointless due to the limitation of a certain model, and they discovered that beyond a certain number of epochs, the model begins over-fitting the dataset and provides biased predictions

2.3 PROBLEM STATEMENT DEFINITION

For years, the traffic department has been combating traffic law violators. These offenders endanger not only their own lives, but also the lives of other individuals. Punishing these offenders is critical to ensuring that others do not become like them. Identification of these offenders is next to impossible because it is impossible for the average individual to write down the license plate of a reckless driver. Therefore, the goal of this project is to help the traffic department identify these offenders and reduce traffic violations as a result.
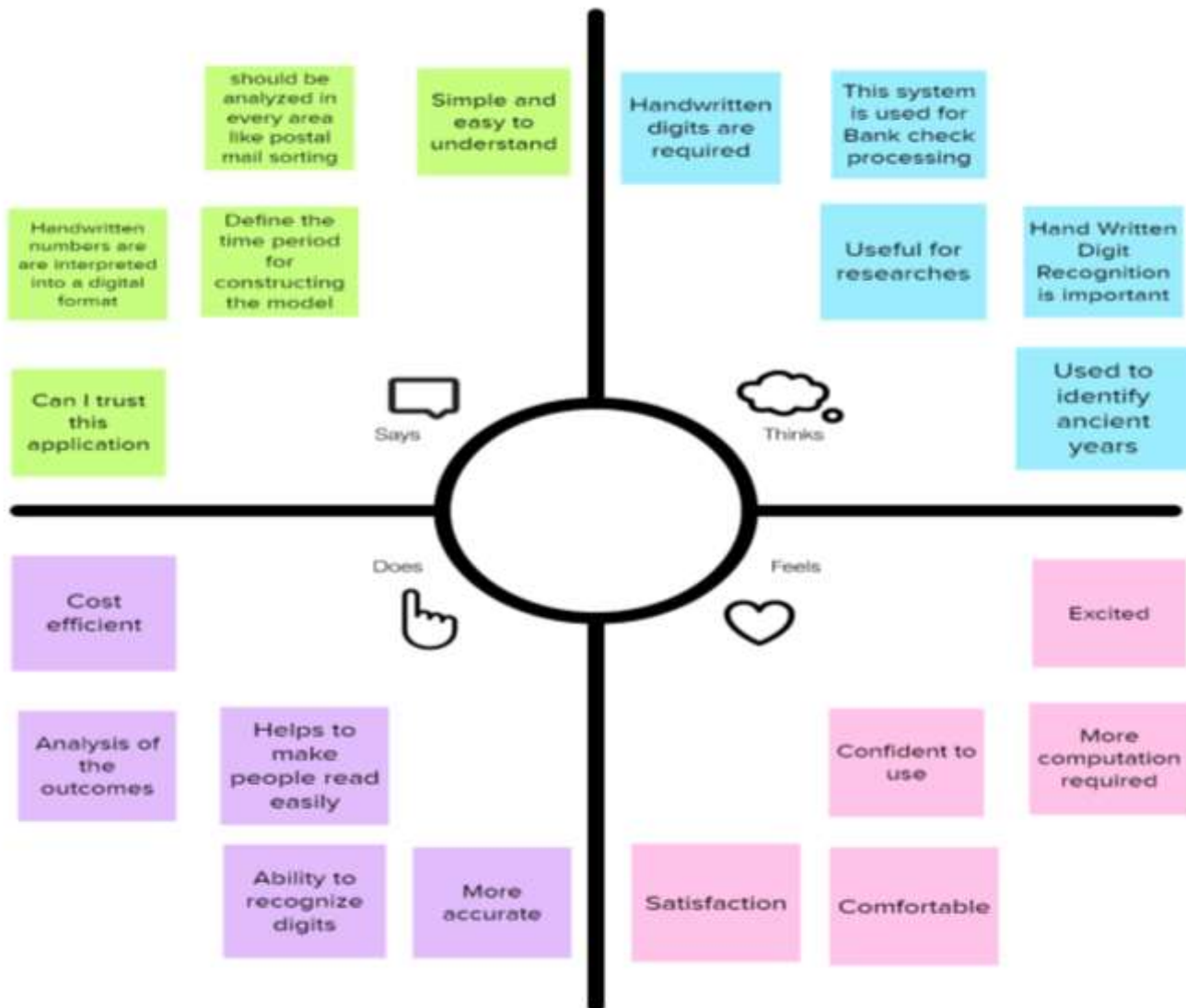
I am	Describe customer with 3-4 key characteristics - who are they?	Describe the customer and their attributes here
I'm trying to	List their outcome or "job" the core about - what are they trying to achieve?	List the thing they are trying to achieve here
but	Describe what problems or barriers stand in the way - what bothers them most?	Describe the problems or barriers that get in the way here
because	Enter the "root cause" of why the problem or barrier exists - what needs to be solved?	Describe the reason the problems or barriers exist
which makes me feel	Describe the emotions from the customer's point of view - how does it impact them emotionally?	Describe the emotions the result from experiencing the problems or barriers

Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	Bank Staff	Process the cheque	It takes long time	Handwriting was not clear and with different style	Irritated and tensed
PS-2	Maths Staff	Trying to evaluate the paper	It consumes more time	Question numbers are not clear and numbers are overlapped	Confused and Anxious

CHAPTER 3
IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

should be analyzed in every area like postal mail sorting

Simple and easy to understand

Handwritten digits are required

This system is used for Bank check processing

Handwritten numbers are are interpreted into a digital format

Define the time period for constructing the model

Useful for researches

Hand Written Digit Recognition is important

Can I trust this application

Says

Thinks

Used to identify ancient years

Does

Feels

Cost efficient

Excited

Analysis of the outcomes

Helps to make people read easily

Confident to use

More computation required

Ability to recognize digits

More accurate

Satisfaction

Comfortable

3.2 IDEATION & BRAINSTORMING

1

Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

⏱ 5 minutes

PROBLEM

It is the hard task for the machine because handwritten digits are not perfect and can be made with many different flavours(styles). The handwritten digit recognition is the solution to this problem which uses the image of a digit and recognizes the digit present in the image. The digits ranges from 0 to 9.

Key rules of brainstorming
To run an smooth and productive session

Stay in topic.	Encourage wild ideas.
Defer Judgment.	Listen to others.
Go for volume.	If possible, be visual.

2

Brainstorm

⏱

Person 1

Person 2

$$4 \cdot 2 \cdot 3$$
$$4 \cdot 9 \cdot 0$$
$$5 \cdot 1 \cdot 1$$
$$9 \cdot 0 \cdot 3$$
$$6 \cdot 7 \cdot 4$$

Person 3
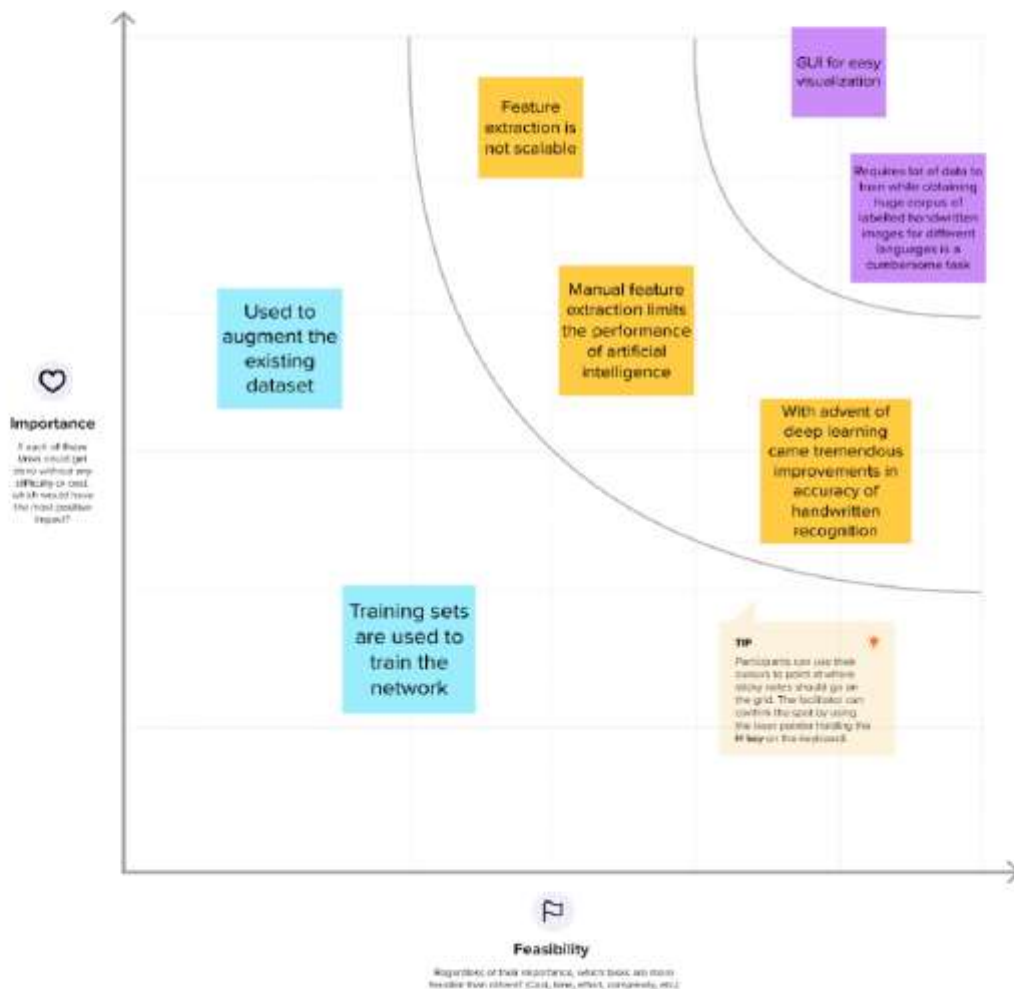
Person 4

3

Group ideas

Step-3: Idea Prioritization

④

Prioritize

Your team should all be on the same page about what's important
moving forward. Place your ideas on this grid to determine which
ideas are important and which are feasible.

🕐 20 minutes

GUI for easy visualization

Feature extraction is not scalable

Requires lot of data to train while obtaining huge corpus of labelled handwritten images for different languages is a cumbersome task

Used to augment the existing dataset

Manual feature extraction limits the performance of artificial intelligence

With advent of deep learning came tremendous improvements in accuracy of handwritten recognition

Importance

Is each of these ideas should get done without any difficulty or cost, which would have the most positive impact?

Training sets are used to train the network

TIP
Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the H key on the keyboard.

Feasibility

Regardless of their importance, which ideas are more feasible than others? (Cost, time, effort, complexity, etc.)

3.3 PROPOSED SOLUTION

S. No.	Parameter	Description
1	Problem Statement (Problem to be solved)	It is the hard task for the machine because handwritten digits are not perfect and can be made with many different flavours. To extract the digits from the image and recognize the digits.
2	Idea / Solution description	To provide an efficient and feasible system for handwritten digits which uses image of the digit and recognize the digit present in the image.
3	Novelty / Uniqueness	· To create the user interface where the user can register, login and upload image for recognition of handwritten digits
4	Social Impact / Customer Satisfaction	• To process the bank cheque • To help the researchers • For paper evaluation
5	Business Model (Revenue Model)	By using this solution • Manpower can be reduced. • Help the teachers to evaluate the papers easily without manual correction • · Helps the bank to verify the account number from cheques without manual checking
6	Scalability of the Solution	It can be used and implemented in any device.

3.4 PROBLEM SOLUTION FIT

1. CUSTOMER SEGMENT(S) **CS**	6. CUSTOMER LIMITATIONS **CL**	5. AVAILABLE SOLUTIONS **AS**
People who want to read handwritten numbers from the paper (bank staff for processing bank cheque, Teachers for evaluating papers)	Blurred and damaged image will not give accurate solution	Traditional handwritten digit recognition systems have relied heavily on handcrafted features and extensive prior knowledge.

2. PROBLEMS / PAINS **PR**	9. PROBLEM ROOT / CAUSE **RC**	7. BEHAVIOR **BU**
People may face difficulties while reading the handwritten images. Handwritten digits are not always of the same style, size and position, not easily understandable. As they differ from person to person. The problem is to classify the digits.	People may face difficulties while reading the handwritten images. Handwritten digits are not always of the same style, size and position, not easily understandable. As they differ from person to person. The problem is to classify the digits.	Client must use this system with clear image and good handwriting to get better results.

3. TRIGGERS TO ACT **TR**	10. YOUR SOLUTION **SL**	8. CHANNELS of BEHAVIOR **CH**
When the person wants a digitalized format of the handwritten digits	It uses VGG-16 (Visual Geometry Group) to recognize the digits. It comes under Convolutional Neural Network. Neural Network is used to train and identify written digits. After training and testing the accuracy will be increase	**ONLINE** Extract channels from Behavior block
4 EMOTIONS BEFORE / AFTER **EM** Confused, irritated, tension, exhausted > convinced, curious, satisfied		**OFFLINE** Extract channels from different handwritten images

CHAPTER 4
REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

| FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	GUI	Allows the user to insert the handwritten image and get the digitized form of the digits. Created for easy virtualization.
FR-2	Uploading image	User can upload the handwritten image
FR-3	Writing in Canvas	User can directly draw/write in the canvas
FR-4	Evaluation	· The MNIST dataset should be trained using CNN to create a trained the model · The trained model has to be tested by using a test data · Predict the output for the input data and display it in a GUI

4.2 NON FUNCTIONAL REQUIREMENTS

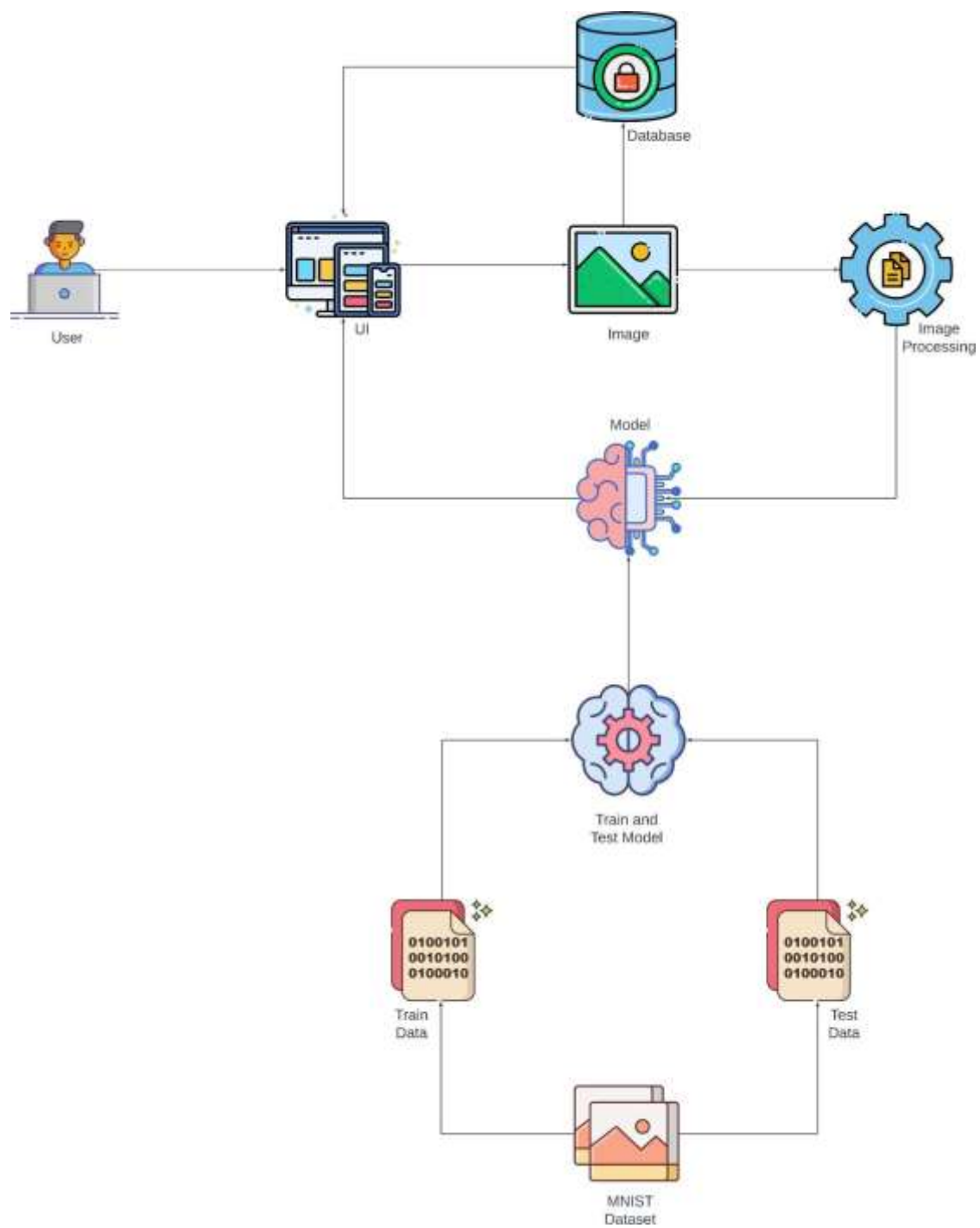| FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The recognition of handwritten characters is one of the major issues with pattern recognition applications. Filling out forms, processing bank checks, and sorting mail are examples of applications using digit recognition.
NFR-2	**Security**	• The segmentation capabilities of the generative models are powered by recognition. • The method makes use of a relatively.
NFR-3	**Reliability**	• The neural network uses the data to automatically determine rules for reading handwritten numerals. By increasing the number of training instances, the network may also learn more about handwriting and hence improve its accuracy. • To recognise handwritten numbers, a wide range of approaches and algorithms can be employed, including Deep Learning/CNN, SVM, Gaussian
NFR-4	**Performance**	Optical character recognition (OCR) technology gives more accuracy rates for typed text in high-quality pictures.
NFR-5	**Availability**	Available for the user who wants to convert handwritten image to digital format
NFR-6	**Scalability**	The task of handwritten digit recognition using a classifier is of great importance and use in a variety of applications, including online handwriting recognition on computer tablets, the processing of bank check amounts, numeric entries in forms filled out by hand and more.

CHAPTER 5
PROJECT DESIGN

5.1DATA FLOW DIAGRAM
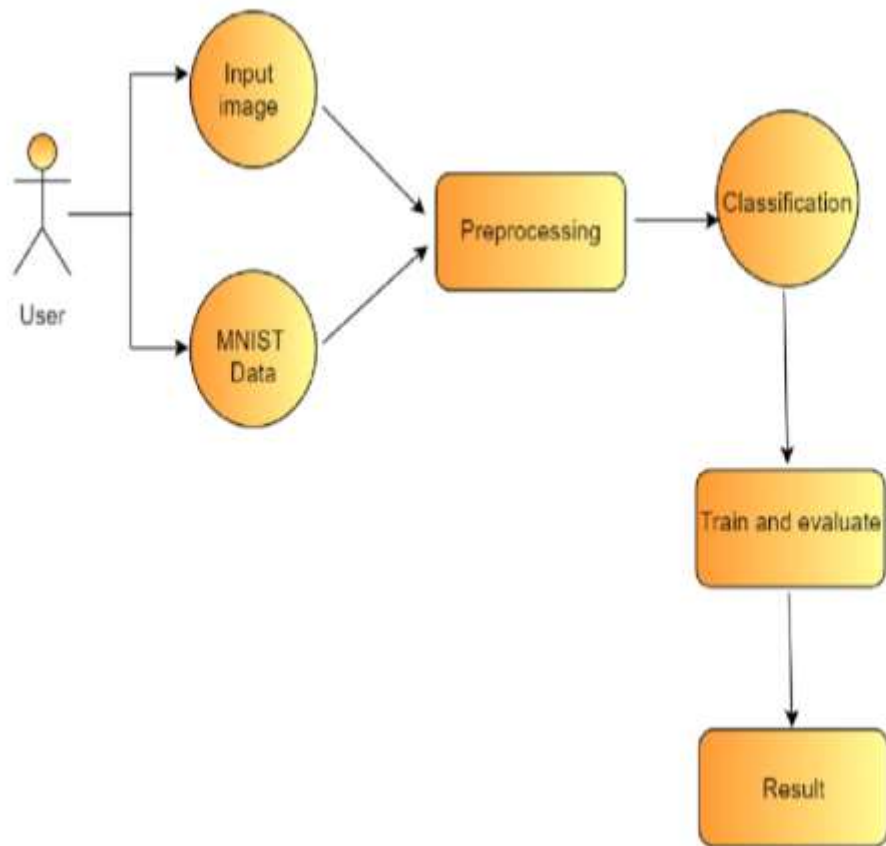
5.2 SOLUTION & TECHNICAL ARCHITECTURE

USER **IBM CLOUD** **WEB INTERFACE**

Input data

Watson studio

Web UI

Deep learning
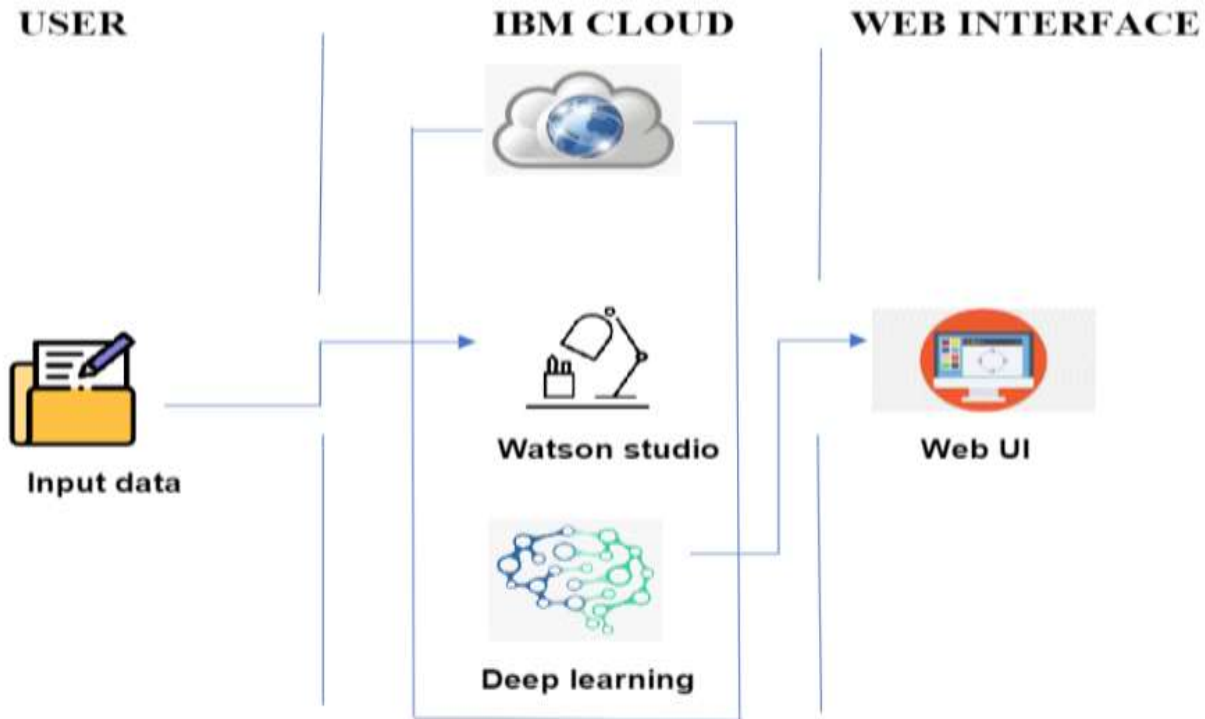
Guidelines:

- The HTML and CSS are used for the user interface for the user to use the application
- The user can see the information on how the image is being recognized.
- Once the button is launched, the user can see the screen to upload the image.
- After uploading the image, the predicted result will be displayed.

Database

User

UI

Image

Image
Processing

Model

Train and
Test Model

0100101
0010100
0100010

Train
Data

0100101
0010100
0100010

Test
Data

MNIST
Dataset

5.3 USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Web user (customer)	Access web page	USN-1	As a user, anyone can access the web page to upload the handwritten image	I can access my web page through online at any time	High	Sprint-1
	Usage of handwritten data	USN-2	As per the style of the handwriting, it is easy to predict the input	Prediction can be done in an easy way	High	Sprint-2
	Accuracy of the handwriting	USN-3	By using the prediction model, the user can check whether the digit is recognized correctly	Prediction of handwritten digit will be accurate	High	Sprint-3
	View the result	USN-4	As a user, he/she can view the digitalized form of the input	Final result will be displayed	High	Sprint-3
Customer Care Executive	Upload clear image/ draw clearly	USN-5	As a user, he/she need to upload clear and neat image to increase accuracy	Result will be accurate	High	Sprint-3

CHAPTER 6
PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNING AND ESTIMATION

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for application	10	High	Abhishek S
Sprint-1	Log in	USN-2	As a user, I can log in into the application	10	High	Abinesh R
Sprint-2	Model creation	USN-3	As a user, I can use the model for predicting the handwritten digits	10	High	Aneerudh V
Sprint-2	Prediction	USN-4	As a user, I can predict the digitalized output	10	High	Ashwin Adhithya K
Sprint-3	Upload Image of the handwritten document	USN-5	As a user, I can able to upload images of the handwritten digit documents	8	Medium	Abhishek S
Sprint-3	Draw the digits on canvas	USN-6	As a user, I can able to draw the digit in canvas	12	High	Abinesh R
Sprint-4	Digit Recognition	USN-7	As a user, I can able to get the digital output of the digit from uploaded handwritten images	9	Medium	Aneerudh V
Sprint-4	Digit Recognition	USN-8	As a user, I can able to get the digital output of the digit from canvas	11	High	Ashwin Adhithya K

6.2 SPRINT DELIVERY SCHEDULE

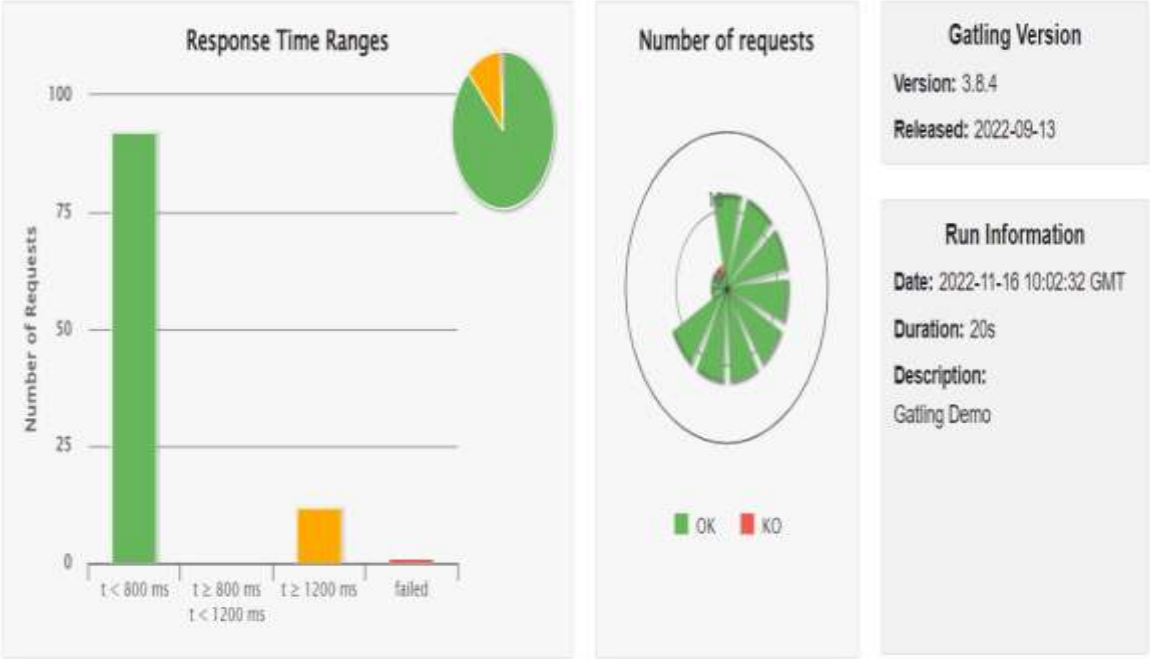| Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022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

CHAPTER 7
CODING & SOLUTIONING

```python
1  import os
2  import random
3  import string
4  from pathlib import Path
5  import numpy as np
6  from tensorflow.keras.models import load_model
7  from PIL import Image, ImageOps
8  import cv2
9
```

```python
11  def recognize(image: bytes) -> int:
12      """
13      Predicts the digit in the image.
14
15      Args:
16          image (bytes): The image data.
17
18      Returns:
19          Tuple: The best prediction, other predictions and file name
20      """
21
22      model=load_model(Path("./model/digit.h5"))
23      image = cv2.imread(image)
24      grey = cv2.cvtcolor(image.copy(), cv2.COLOR_BGR2GRAY)
25      ret, thresh = cv2.threshold(grey.copy(), 75, 255, cv2.THRESH_BINARY_INV)
26      contours, _ = cv2.findContours(thresh.copy(), cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
27      preprocessed_digits = []
28      for c in contours:
29          x,y,w,h = cv2.boundingRect(c)
30          cv2.rectangle(image, (x,y), (x+w, y+h), color=(0, 255, 0), thickness=2)
31          digit = thresh[y:y+h, x:x+w]
32          resized_digit = cv2.resize(digit, (18,18))
33          padded_digit = np.pad(resized_digit, ((5,5),(5,5)), "constant", constant_values=0)
34          preprocessed_digits.append(padded_digit)
35      for digit in preprocessed_digits:
36          prediction = model.predict(digit.reshape(1, 28, 28, 1))
37          best= np.argmax(prediction)
38
39
40
41      return best, "1.jpg"
```

CHAPTER 8
TESTING

8.1 TEST CASES

Response Time Ranges	Number of requests	Gatling Version

Gatling Version

Version: 3.8.4

Released: 2022-09-13

Run Information

Date: 2022-11-16 10:02:32 GMT

Duration: 20s

Description:

Gatling Demo

```java
import java.time.Duration;
import java.util.*;

import io.gatling.javaapi.core.*;
import io.gatling.javaapi.http.*;
import io.gatling.javaapi.jdbc.*;

import static io.gatling.javaapi.core.CoreDsl.*;
import static io.gatling.javaapi.http.HttpDsl.*;
import static io.gatling.javaapi.jdbc.JdbcDsl.*;

public class RecordedSimulation extends Simulation {

    {
        HttpProtocolBuilder httpProtocol = http
            .baseUrl("http://127.0.0.1:5000")
            .inferHtmlResources()
            .acceptHeader("text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9")
            .acceptEncodingHeader("gzip, deflate")
            .acceptLanguageHeader("en-US,en;q=0.9")
            .upgradeInsecureRequestsHeader("1")
            .userAgentHeader("Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.0.0 Safari/537.36");

        Map<CharSequence, String> headers_0 = new HashMap<>();
        headers_0.put("Sec-Fetch-Dest", "document");
        headers_0.put("Sec-Fetch-Mode", "navigate");
        headers_0.put("Sec-Fetch-Site", "same-origin");
        headers_0.put("Sec-Fetch-User", "?1");
        headers_0.put("sec-ch-ua", "\"Google Chrome\";v=\"107\", \"Chromium\";v=\"107\", \"Not=A?Brand\";v=\"24\"");
        headers_0.put("sec-ch-ua-mobile", "?0");
        headers_0.put("sec-ch-ua-platform", "Windows");

        Map<CharSequence, String> headers_2 = new HashMap<>();
        headers_2.put("Cache-Control", "max-age=0");
        headers_2.put("Sec-Fetch-Dest", "document");
        headers_2.put("Sec-Fetch-Mode", "navigate");
        headers_2.put("Sec-Fetch-Site", "same-origin");
```

8.2 USER ACCEPTANCE TESTING

Test Case ID	Feature Type	Component	Test Scenario	Steps To Execute	Test Data	Expected Result	Actual Result	Status
HomePage_TC_OO1	Functional	Home Page	Verify user is able to see the Login/Signup popup when user clicked on My account button	1.Enter URL and click go 2.Click on My Account icon 3.Verify login/Signup popup displayed or not	-	Login/Signup popup should display	Working as expected	Pass
RegisterPage_TC_OO1	UI	Registration Page	Verify the UI elements in Register/Signup page	1.Enter URL and click go 2.Click on My Account dropdown button 3.Verify Register/Signup popup with below UI elements: a.name text box b. email text box c. phone number text box d. password text box e. gender text box f. Already have an account? Click login	-	Application should show below UI elements: a.name text box b. email text box c. phone number text box d. password text box e. gender text box f. Already have an account? Click login	Working as expected	Pass
RegisterPage_TC_OO2	Functional	Registration Page	Verify the users entering the unique email	1.Enter URL and click go 2.Click on My Account dropdown button 3.Verify Register/Signup page accepts only unique email	-	Application should allow only unique email address	Working as expected	Pass
RegisterPage_TC_OO2	Functional	Registration Page	Verify that the user can able to register with valid credentials	1.Enter URL and click go 2.Click on My Account dropdown button 3.Click Register/Signup popup A. Enter name b. Enter email c. Enter phone number d. Enter password e. Enter gender f. Click Register button	-	User should navigate to sign in page	Working as expected	Pass

LoginPage_T C_OO1	UI	Login page	Verify the UI elements in Login/Sign in page	1.Enter URL and click go 2.Click on My Account dropdown button 3.Verify login/Signup popup with below UI elements: a. email text box b. password text box c. Login button d. New customer? Create account link	-	Application should show below UI elements: a.email text box b.password text box c.Login button with orange colour d.New customer? Create account link	Working as expected	Pass
LoginPage_T C_OO2	Functional	Login page	Verify user is able to log into application with Valid credentials	1.Enter URL and click go 2.Click on My Account dropdown button and click on sign in/login pop up 3.Enter Valid email in Email text box 4.Enter valid password in password text box 5.Click on login button	Email: chalam@gmail.com password: Testing123	User should navigate to Description page	Working as expected	Pass
LoginPage_T C_OO3	Functional	Login page	Verify user is not able to log into application with Invalid credentials	1.Enter URL and click go 2.Click on My Account dropdown button and click on sign in/login pop up 3.Enter Invalid email in Email text box 4.Enter valid password	Username: chalam@gmail password: Testing123	Application should show 'Email not available' validation message.	Working as expected	Pass

				in password text box 5.Click on login button				
LoginPage_T C_OO4	Functional	Login page	Verify user is not able to log into application with Invalid credentials	1.Enter URL and click go2.Click on My Account dropdown button and click on sign in/login pop up3.Enter Valid username/email in Email text box4.Enter Invalid password in password text box5.Click on login button	Username: chalam@gmail.compassword: Testing1236786867 86876876	Application should show 'Incorrect password' validation message.	Working as expected	Pass
DescriptionP age_TC_OO2	UI	Description page	Verify user is able to see the description and predict button	1.Enter URL and click go 2.Click on My Account dropdown button 3.Enter Valid email in Email text box 4.Enter valid password in password text box 5.Click on login button		Application should navigate to Description page and user can able to view the description and predict button	Working as expected	Pass

DescriptionPage_TC_OO2	Functional	Description page	Verify user is able to click the predict button	1.Enter URL and click go 2.Click on My Account dropdown button 3.Enter Valid email in Email text box 4.Enter valid password in password text box 5.Click on login button 6.Click on predict button		User Should navigate to Prediction page	Working as expected	Pass
DescriptionPage_TC_OO3	Functional	Description page	Verify user is able to click the logout button			User should navigate to Home page after clicking on Logout button	Working as expected	Pass
PredictionPage_TC_OO1	UI	Prediction Page	Verify user is able to Choose and submit the file and predict the output	1.Enter URL and click go 2.Click on My Account dropdown button 3.Enter Valid email in Email text box 4.Enter valid password in password text box 5.Click on login button 6.Click on predict button 7.Click on Choose button to select the file		User can able to upload the file and predict the output	Working as expected	Pass
PredictionPage_TC_OO2	Functional	Prediction Page	Verify user is able to choose only the accepted file format		A.pdf	User can able to upload only .png, .jpg, .jpeg files		Fail

8.2.1 DEFECT ANALYSIS

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	20	4	2	10	36
Duplicate	1	0	0	1	2
External	2	4	1	2	9
Fixed	3	6	4	10	23
Not Reproduced	1	2	1	1	5
Skipped	1	1	0	1	3
Won't Fix	0	5	3	7	15
Totals	28	22	11	32	99

8.2.2TESTCASE ANALYSIS
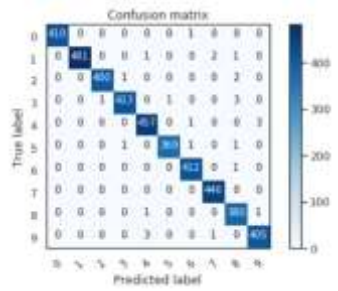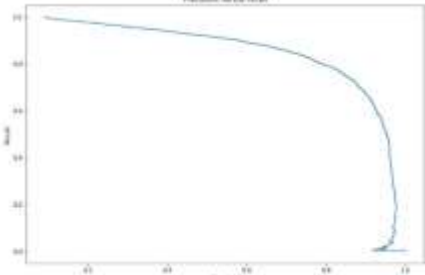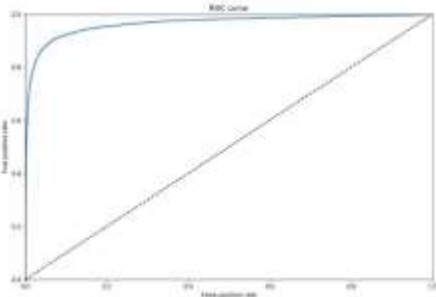
Section	Total Cases	Not Tested	Fail	Pass
Print Engine	10	0	0	10
Client Application	20	0	0	20
Security	2	0	0	2
Exception Reporting	9	0	0	9
Final Report Output	7	0	0	7
Version Control	2	0	0	2

CHAPTER 9
RESULTS

9.1 PERFORMANCE METRICS

S.No.	Parameter	Values	Screenshot
1.	Model Summary	Model: "sequential" Layer (type) Output Shape Param # conv2d (Conv2D) (None, 26, 26, 64) 640 conv2d_1 (Conv2D) (None, 24, 24, 32) 18464 flatten (Flatten) (None, 18432) 0 dense (Dense) (None, 10) 184330 Total params: 203,434 Trainable params: 203,434 Non-trainable params: 0	
2.	Accuracy	Training Accuracy -0.9979166388511658 Validation Accuracy -0.98089998960495	
3. `	Metrics	**Classification Model:** precision,recall,f1-score,support	

4.	Metrics	Confusion Matrix	
5.	Metrics	Precision-Recall or PR curve	
6.	Metrics	ROC (Receiver Operating Characteristics) curve	

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

DISADVANTAGES

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

CHAPTER 11
CONCLUSION

This project demonstrated a web application that uses machine learning to recognise handwritten numbers. Flask, HTML, CSS, JavaScript, and a few other technologies were used to create this project. The model predicts the handwritten digit using a CNN network. During testing, the model achieved a 98% 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. There is so much room for improvement, which can be implemented in subsequent versions.

CHAPTER 12
FUTURE SCOPE

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

- Add support to detect digits from manual writing in canvas
- 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

MODEL CREATION:

```
from keras.datasets import mnist

import matplotlib.pyplot as plt

from keras.utils import np_utils

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Conv2D,Dense,Flatten

from tensorflow.keras.optimizers import Adam

(X_train,y_train),(

X_test,y_test) =mnist.load_data()

print(X_train.shape)

print(X_test.shape)

print(y_test.shape)

print(y_train.shape)

print("The label value is ",y_test[10]) #Value in y_test

plt.imshow(X_test[10])

print("The label value is ",y_test[65]) #Value in y_test

plt.imshow(X_test[65])

X_train.shape

X_test.shape

 X_train1 = X_train.reshape(60000, 28, 28, 1).astype('float32')

X_test1 = X_test.reshape(10000, 28, 28, 1).astype('float32')

number_of_classes= 10

y_train1 = np_utils.to_categorical(y_train,number_of_classes)

y_test1 = np_utils.to_categorical(y_test,number_of_classes)

print("After encoding the value",y_test[10] ,"become", y_test1[10])

print("After encoding the value",y_test[100] ,"become", y_test1[100])

 print("After encoding the value",y_test[65] ,"become", y_test1[65])
```

```python
model = Sequential()

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

model.add(Conv2D(32, (3, 3), activation="relu"))

model.add(Flatten())

model.add(Dense(number_of_classes, activation="softmax"))

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

model.fit(X_train1, y_train1, batch_size=32, epochs=5, validation_data=(X_test1,y_test1))

metrics = model.evaluate(X_test1, y_test1, verbose=0)

print("Metrics (Test Loss & Test Accuracy): ")

print(metrics)

prediction = model.predict(X_test1[:4])

print(prediction)

import numpy as np

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

print(y_test1[:4])

model.save("model.h5")

from tensorflow.keras.models import load_model

model=load_model("model.h5")

model.summary()
```

FLASK APP:

```python
from flask import Flask, render_template, request,redirect,session, url_for

from flask_mail import Mail, Message

from itsdangerous import URLSafeTimedSerializer, SignatureExpired

import mysql.connector

import os

from flask_mysqldb import MySQL

from recognize import recognize

import requests

from io import BytesIO
```

```python
from werkzeug.utils import secure_filename

app = Flask(__name__)

app.secret_key=os.urandom(24)

app.config['MYSQL_HOST'] = 'localhost'

app.config['MYSQL_USER'] = 'root'

app.config['MYSQL_PASSWORD'] = ''

app.config['MYSQL_DB'] = 'digit_recognition'

mysql = MySQL(app)

@app.route('/')

def index():

    return render_template('index.html')

@app.route('/login')

def login():

    return render_template('login.html')

    @app.route('/register/')

    def about():

        return render_template('form.html')

@app.route('/home')

def home():

    if 'email' in session:

        return render_template('home.html')

    else:

        return redirect('/')

    @app.route('/login_validation',methods=['POST'])

    def login_validation():

        if request.method == "POST":

            email=request.form.get('email')

            password=request.form.get('password')

            error = None
```

```python
    if mysql:
        print("Connection Successful!")
        cursor = mysql.connection.cursor()
        cursor.execute("""SELECT * FROM `users` where `Email` LIKE '{}' """.format(email))
        users = cursor.fetchall()
        cursor.close()
        cursor1 = mysql.connection.cursor()
        cursor1.execute("""SELECT * FROM `users` where `Email` LIKE '{}' and `Password` LIKE
'{}'""".format(email, password))
        users1 = cursor1.fetchall()
        cursor1.close()
    else:
        print("Connection Failed!")
    if len(users)>0:
        if len(users1)>0:
            session['email'] = users[0][1]
            return redirect('/home')
        else:
            error = "Wrong password"
    else:
        error = "Email not available"
    return render_template('login.html',error=error)
@app.route('/add_user',methods=['POST'])
def add_user():
    username=request.form.get('username')
    email = request.form.get('email')
    password = request.form.get('password')
    phone = request.form.get('phone')
    gender = request.form.get('gender')
    if mysql:
```

```python
        print("Connection Successful!")

        cursor = mysql.connection.cursor()

        cursor.execute(

            """INSERT   INTO   `users`   (`FullName`,`Email`,`Password`,`PhoneNo`,`Gender`)   VALUES
('{}','{}','{}','{}','{}')""".format(username,email, password,phone,gender))

        mysql.connection.commit()

        cursor.close()

    else:

        print("Connection Failed!")

    return redirect('/login')

@app.route('/logout')

def logout():

    return redirect('/')

@app.route('/predictpage',methods=['POST'])

def predictpage():

    return render_template('prediction.html')

@app.route('/submit',methods=['POST'])

def submit():

    if request.method == 'POST':

        # Upload file flask

        uploaded_img = request.files['image']

        # Upload file to database (defined uploaded folder in static path)

        uploaded_img.save('./static/data/1.jpg')

        # Storing uploaded file path in flask session

        session['uploaded_img_file_path'] = "./static/data/1.jpg"

        return render_template('prediction.html')

@app.route('/prediction',methods=('POST', "GET"))

def predict():

    # Retrieving uploaded file path from session

    img_file_path = session.get('uploaded_img_file_path', None)
```

```python
        best, img1 = recognize(img_file_path)

        return render_template("prediction.html", best=best, img_name=img1)

if __name__=="__main__":

    app.run(debug=True)
```

RECOGNIZER(PYTHON):

```python
import os

import random

import string

from pathlib import Path

import numpy as np

from tensorflow.keras.models import load_model

from PIL import Image, ImageOps

import cv2

def recognize(image: bytes) -> int:

    """

    Predicts the digit in the image


    Args:

        image (bytes): The image data.

    Returns:

        tuple: The best prediction, other predictions and file name

    """

    model=load_model(Path("./model/digit.h5"))

    image = cv2.imread(image)

    grey = cv2.cvtColor(image.copy(), cv2.COLOR_BGR2GRAY)

    ret, thresh = cv2.threshold(grey.copy(), 75, 255, cv2.THRESH_BINARY_INV)

    contours, _ = cv2.findContours(thresh.copy(), cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)

    preprocessed_digits = []

    for c in contours:

        x,y,w,h = cv2.boundingRect(c)

        cv2.rectangle(image, (x,y), (x+w, y+h), color=(0, 255, 0), thickness=2)

        digit = thresh[y:y+h, x:x+w]

        resized_digit = cv2.resize(digit, (18,18))
```

```
        padded_digit = np.pad(resized_digit, ((5,5),(5,5)), "constant", constant_values=0)
        preprocessed_digits.append(padded_digit)
    for digit in preprocessed_digits:
      prediction = model.predict(digit.reshape(1, 28, 28, 1))
      best= np.argmax(prediction)
    return best, "1.jp
```

FORM PAGE(HTML):

```html
<!doctype html>
<html lang="en">
 <head>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
<link rel="stylesheet" type="text/css" href="{{ url_for('static',filename='css/style.css')}}">
  <!-- Bootstrap CSS -->
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/css/bootstrap.min.css"
integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">
<title>Handwritten Digit Recognition</title>
  <img src="{{ url_for('static', filename='images/registerimg2.jpg') }}" alt="" width="600" height="650"
style="margin-left: 100px;margin-right:0%;">
   <style>
*{
 margin: 0;
 padding: 0;
 box-sizing: border-box;
 font-family: 'Poppins',sans-serif;
}
body{
 height: 100vh;
 display: flex;
 justify-content: center;
 align-items: center;
 padding: 10px;
```

```css
    background: linear-gradient(135deg, #71b7e6, #9b59b6);
}
.container{
 max-width: 700px;
 margin-left: 0%;
 width: 100%;
 background-color: #fff;
 padding: 25px 30px;
 border-radius: 5px;
 box-shadow: 0 5px 10px rgba(0,0,0,0.15);
}
.container .title{
 font-size: 25px;
 font-weight: 500;
 position: relative;
}
.container .title::before{
 content: "";
 position: absolute;
 left: 0;
 bottom: 0;
 height: 3px;
 width: 30px;
 border-radius: 5px;
 background: linear-gradient(135deg, #71b7e6, #9b59b6);
}
.content form .user-details{
 display:inline;
 flex-wrap: wrap;
 justify-content: space-between;
 margin: 20px 0 12px 0;
}
 form .input-box span.details{
 display: block;
 font-weight: 500;
```

```css
    margin-bottom: 5px;
}
.user-details .input-box input{
 height: 45px;
 width: 100%;
 outline: none;
 font-size: 16px;
 border-radius: 5px;
 padding-left: 15px;
 border: 1px solid #ccc;
 border-bottom-width: 2px;
 transition: all 0.3s ease;
}
.user-details .input-box input:focus,
.user-details .input-box input:valid{
 border-color: #9b59b6;
}
 form .gender-details .gender-title{
 font-size: 20px;
 font-weight: 500;
 }
 form .category{
  display: flex;
  width: 80%;
  margin: 14px 0 ;
  justify-content: space-between;
 }
 form .category label{
  display: flex;
  align-items: center;
  cursor: pointer;
 }
 form input[type="radio"]{
  display: none;
 }
```

```css
form .button{
  height: 45px;
  margin: 35px 0
}
form .button input{
  height: 100%;
  width: 100%;
  border-radius: 5px;
  border: none;
  color: #fff;
  font-size: 18px;
  font-weight: 500;
  letter-spacing: 1px;
  cursor: pointer;
  transition: all 0.3s ease;
  background: linear-gradient(135deg, #71b7e6, #9b59b6);
}
form .button input:hover{
/* transform: scale(0.99); */
background: linear-gradient(-135deg, #71b7e6, #9b59b6);
}
@media(max-width: 584px){
.container{
max-width: 100%;
}
form .user-details .input-box{
  margin-bottom: 15px;
  width: 100%;
}
form .category{
  width: 100%;
}
.content form .user-details{
  max-height: 300px;
  overflow-y: scroll;
```

```
        }
      .user-details::-webkit-scrollbar{
        width: 5px;
      }
      }
      @media(max-width: 459px){
      .container .content .category{
        flex-direction: column;
      }
    }
    </style>
    </head>
    <body class="bg-nav">
      <div class="container">
        <div class="title">Registration</div><br>
        <div class="content">
          <form method="post" action="/add_user">
            <div class="user-details">
              <div class="input-box">
                <label>Username</label><br>
                <input type = "text" class="form-control" name="username" placeholder="Enter your name" required>
              </div>
              <div class="input-box">
                <label>Email</label><br>
                <input type = "email" class="form-control" name="email" placeholder="Enter your email" required>
              </div>
              <div class="input-box">
                <label>Password</label><br>
                <input type="password" class="form-control" name="password" placeholder="Enter your password"
required>
              </div>
              <div class="input-box">
                <label>Phone Number</label><br>
                <input type = "number" class="form-control" name="phone" placeholder="Enter your number" required>
              </div>
```

```html
    <div class="input-box">
     <label>Gender</label><br>
     <input type = "text" class="form-control" placeholder="Enter Male/Female/Others" name="gender"
required>
    </div>
    <div class="button">
     <input type="submit" class="btn btn-primary btn-block btn-lg" value="Register">
    </div>
   </div>
    <p style="text-align:center;">Already have an Account?   <a href="/login" class="reg">Login</a></h4>
   </form>
  </div>
 </div>
</body>
```

INDEX PAGE(HTML):

```html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>User Account Dropdown Menu Using Html CSS & Vanilla Javascript</title>
  <link rel="stylesheet" href="style.css">
  <link     href="https://fonts.googleapis.com/css2?family=Poppins:wght@200;300;400;600&display=swap"
rel="stylesheet">
  <link
href="https://fonts.googleapis.com/css?family=Material+Icons|Material+Icons+Outlined|Material+Icons+Two+
Tone|Material+Icons+Round|Material+Icons+Sharp" rel="stylesheet">
  <style>
    *{
      font-family: "poppins", sans-serif;
      margin: 0;
      padding: 0;
    }
```

```css
.icons-size{
  color: white;
  font-size: 14px;
}
.text{
  position: fixed;
  right: 950px;
  top:22px
}
.action{
  position: fixed;
  right: 1450px;
  top:20px
}
.action .profile{
  border-radius: 50%;
  cursor: pointer;
  height: 60px;
  overflow: hidden;
  position: relative;
  width: 60px;
}
.action .profile img{
  width: 100%;
  top:0;
  position: absolute;
  object-fit: cover;
  left: 0;
  height: 100%;
}
.action .menu{
  background-color:#FFF;
  box-sizing:0 5px 25px rgba(0,0,0,0.1);
  border-radius: 15px;
  padding: 10px 20px;
```

```css
    position: absolute;
    left: -10px;
    width: 200px;
    transition: 0.5s;
    top: 120px;
    visibility: hidden;
    opacity: 0;
}
.action .menu.active{
    opacity: 1;
    top: 80px;
    visibility: visible;
}
.action .menu::before{
    background-color:#fff;
    content: '';
    height: 20px;
    position: absolute;
    right: 190px;
    transform:rotate(45deg);
    top:-5px;
    width: 20px;
}
.action .menu h3{
    color: #555;
    font-size: 16px;
    font-weight: 600;
    line-height: 1.3em;
    padding: 20px 0px;
    text-align: left;
    width: 100%;
}
.action .menu h3 div{
    color: #818181;
    font-size: 14px;
```

```css
    font-weight: 400;
}
.action .menu ul li{
    align-items: center;
    border-top:1px solid rgba(0,0,0,0.05);
    display: flex;
    justify-content: left;
    list-style: none;
    padding: 10px 0px;
}
.action .menu ul li img{
    max-width: 20px;
    margin-right: 10px;
    opacity: 0.5;
    transition:0.5s
}
.action .menu ul li a{
    display: inline-block;
    color: #555;
    font-size: 14px;
    font-weight: 600;
    padding-left: 15px;
    text-decoration: none;
    text-transform: uppercase;
    transition: 0.5s;
}
.action .menu ul li:hover img{
    opacity: 1;
}
.action .menu ul li:hover a{
    color:#ff00ff;
}
.msg{
    position: fixed;
    right: 950px;
```

```
       left:0px;
       top:250px
   }
       .image {
       background-image: url("{{ url_for('static', filename='images/index2.jpg') }}");
       background-color: #cccccc;
       height: 753px;
       width: 1536px;
       background-position: center;
       background-repeat: no-repeat;
       background-size: cover;
       }
   </style>
</head>
<body>
    <div class="image">
       <div class="action">
          <div class="profile" onclick="menuToggle();">
             <img src="{{ url_for('static', filename='images/user.jpg') }}" alt="">
              </div>
    <div class="menu">
           <h3>
              User Account
           </h3>
           <ul>
             <li>
                <span class="material-icons icons-size">person</span>
                <a href="/register/">Sign-up</a>
             </li>
             <li>
                <span class="material-icons icons-size">mode</span>
                <a href="/login">Sign-in</a>
             </li>
           </ul>
         </div>
```

```html
        </div>
      <div class="text">
        <h1 style="color: white;">Handwritten digit Recognisor</h1>
      </div>
      <div class="msg">
        <h1 style="color: white;font-size: 50px;">
        <pre>
          Numbers
          Rule
          The
          Universe
        </pre>
        </h1>
      </div>
    </div>
  <script>
    function menuToggle(){
      const toggleMenu = document.querySelector('.menu');
      toggleMenu.classList.toggle('active')
    }
  </script>
</body>
```

LOGIN PAGE(HTML):

```html
<!doctype html>
<html lang="en">
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <link rel="stylesheet" type="text/css" href="{{ url_for('static',filename='css/style.css')}}">
    <!-- Bootstrap CSS -->
    <link    rel="stylesheet"    href="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/css/bootstrap.min.css"
integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
```

```html
    crossorigin="anonymous">
    <link        href="https://fonts.googleapis.com/css2?family=Poppins:wght@200;300;400;600&display=swap"
rel="stylesheet">
    <link
href="https://fonts.googleapis.com/css?family=Material+Icons|Material+Icons+Outlined|Material+Icons+Two+T
one|Material+Icons+Round|Material+Icons+Sharp" rel="stylesheet">
    <title>Handwritten Digit Recognition</title>
    <style>
     .image {
     background-image: url("{{ url_for('static', filename='images/loginimg2.jpg') }}");
     background-color: #cccccc;
     height: 753px;
     background-position: center;
     background-repeat: no-repeat;
     background-size: cover;
     position: relative;
     }
    </style>
  </head>
  <body class="bg-nav">
   <div class="image">
   <div id="content" >
      <div class="container">
       <br><br>
       <h1 style="text-align:center;">Sign In</h1>
       <div class="row">
        <div class="col-md-6" style="margin-left: 300px;">
           <div class="card">
             <div class="card-body" style="border:2px solid black">
               <form class="form" method="post" action="/login_validation">
                 <label>Email</label><br>
                 <input type = "email" class="form-control" name="email"><br>
                 <label>Password</label><br>
                 <input type="password" class="form-control" name="password"><br><br>
                 {% if error %}
```

```html
            <p style="text-align: center;"><strong>Error</strong>: {{error}}</p>
            {% endif %}
            <input type="submit" class="btn btn-primary btn-block btn-lg" value="Login">
          </form>
          <br>
          <p style="margin-top:10px;text-align:center;">Not a member? <a href="/register/">Create
Account</a></p>
        </div>
      </div>
    </div>
  </div>
  </div>
  </div>
<!-- Optional JavaScript -->
  <!-- jQuery first, then Popper.js, then Bootstrap JS -->
  <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://cdn.jsdelivr.net/npm/popper.js@1.14.7/dist/umd/popper.min.js" integrity="sha384-
UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1"
crossorigin="anonymous"></script>
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/js/bootstrap.min.js" integrity="sha384-
JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM"
crossorigin="anonymous"></script>
  </body>
</html>
```

HOME PAGE (HTML):

<!DOCTYPE html>

<html lang="en">

 <head>

  <!-- Required meta tags -->

  <meta charset="utf-8">

  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

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

  <!-- Bootstrap CSS -->

  <link   rel="stylesheet"   href="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/css/bootstrap.min.css"
  integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
  crossorigin="anonymous">

 <title>Handwritten Digit Recognition</title>

   </head>

  <body>

  <div class="bg-nav text-light d-flex flex-column flex-md-row align-items-center p-3 px-md-4 mb-3 bg-dark
  border-bottom shadow-sm" >

   <h5 class="my-0 mr-md-auto" style="color: black;font-weight: bolder;">Handwritten Digit Recognisor</h5>

   Log Out

  </div>

<main role="main">

<section class="album py-3 text-center">

    <form action="/predictpage" method="post">

     <div class="button">

      <input type="submit" class="btn btn-primary btn-block btn-lg" value="Proceed to recognise the handwritten
    digits">

      </div>

    </form>

   </section>

<div class="jumbotron py-8 bg-dark">

    <div class="container">

```html
    <div class="row">

        <div class="card mb-2 shadow-sm">

            <div class="card-body" style="border:5px solid black;">

            <h3 style="text-align: center;">Description</h3><br>

            <p class="card-text" style="font-size: 20px;">The handwritten digit recognition is the ability of
computers to recognize human handwritten digits.

            It is a hard task for the machine because handwritten digits are not perfect, vary from person-to-person
and can be made with many different flavors.

            The handwritten digit recognition is the solution to this problem which uses the image of a digit and
recognizes the digit present in the image.</p>

            </div>

        </div><br><br>

      </div>

      </div>

     </div>

    </main>

  <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://cdn.jsdelivr.net/npm/popper.js@1.14.7/dist/umd/popper.min.js"    integrity="sha384-
    UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1"
    crossorigin="anonymous"></script>

        <script    src="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/js/bootstrap.min.js"    integrity="sha384-
    JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM"
    crossorigin="anonymous"></script>

    </body>

    </html>
```

HOME PAGE (CSS):

```css
.bg-nav
{
    background: #e704c9;  /* fallback for old browsers */
    background: -webkit-linear-gradient(to right, #E5E5BE, #db0ac2);  /* Chrome 10-25, Safari 5.1-6 */
    background: linear-gradient(to right, #E5E5BE, #f104b6); /* W3C, IE 10+/ Edge, Firefox 16+, Chrome 26+,
Opera 12+, Safari 7+ */
}
.row{
    margin-top:80px;
```

PREDICT PAGE (HTML):

```html
<!DOCTYPE html>

<html lang="en">

 <head>

    <!-- Required meta tags -->

    <meta charset="utf-8">

    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

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

    <!-- Bootstrap CSS -->

    <link     rel="stylesheet"     href="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/css/bootstrap.min.css"
integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">

<title>Handwritten Digit Recognition</title>

  </head>

  <body>
```

```html
 <div class="bg-nav text-light d-flex flex-column flex-md-row align-items-center p-3 px-md-4 mb-3 bg-dark border-bottom shadow-sm" >

 <h5 class="my-0 mr-md-auto" style="color: black;font-weight: bolder;">Handwritten Digit Recognisor</h5>

 <a class="btn btn-outline-primary" href="/logout" style="color: whitesmoke;">Log Out</a>

</div>

<main role="main">

 <section class="album py-3 text-center">

  <form action="/predictpage" method="post">

   <div class="button">

    <input type="submit" class="btn btn-primary btn-block btn-lg" value="Proceed to recognise the handwritten digits">

   </div>

  </form>

 </section>

<div class="jumbotron py-8 bg-dark">

  <div class="container">

 <div class="row">

     <div class="card mb-2 shadow-sm">

      <div class="card-body" style="border:5px solid black;">

      <h3 style="text-align: center;">Description</h3><br>

      <p class="card-text" style="font-size: 20px;">The handwritten digit recognition is the ability of computers to recognize human handwritten digits.

        It is a hard task for the machine because handwritten digits are not perfect, vary from person-to-person and can be made with many different flavors.
```

The handwritten digit recognition is the solution to this problem which uses the image of a digit and recognizes the digit present in the image.</p>

                            </div>

                        </div>

                    </div>

                </div>

            </div>

        </main>

    <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://cdn.jsdelivr.net/npm/popper.js@1.14.7/dist/umd/popper.min.js" integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1" crossorigin="anonymous"></script>

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/js/bootstrap.min.js" integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM" crossorigin="anonymous"></script>

  </body>

  </html>

GITHUB

https://github.com/IBM-EPBL/IBM-Project-22740-1659857284