PROJECT REPORT

A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

Submitted By

Team ID - PNT2022TMID44131

Vadivelan A	723919104010
Ajithkumar N	723919104301
Fathima Anjoom PV	723919104303
Krishnaveni G	723919104004

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 recognize handwritten digits form various sources such as images, documents, and so on. This project aims to let users take advantage of machine learning to reduce manual task 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 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 numbers like 1 and 7, 5 and 6, 3 and 8, 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

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

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.

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

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.

Improved Handwritten Digit Recognition Using Convolutional Neural Networks (CNN) (2020)

Ahlawat, Savita and Choudhary, Amit and Nayyar, Anand and Singh, Saurabh and Yoon, Byungun

This paper's primary goal was to enhance handwritten digit recognition ability. To avoid difficult pre-processing, expensive feature extraction, and a complex ensemble (classifier combination) method of a standard recognition system, they examined different convolutional neural network variations. Their current work makes suggestions on the function of several hyper-parameters through thorough evaluation utilizing an MNIST dataset. They also confirmed that optimizing hyper-parameters is crucial for enhancing CNN architecture performance. With the Adam optimizer for the MNIST database, they were able to surpass many previously published results with a recognition rate of 99.89%. Through the trials, it is made abundantly evident how the performance of handwritten digit recognition is affected by the number of convolutional layers in CNN architecture. According to the paper, evolutionary algorithms can be explored for optimizing convolutional filter kernel sizes, CNN learning parameters, and the quantity of layers and learning rates.

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

Pashine, Samay and Dixit, Ritik and Kushwah, Rishika

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

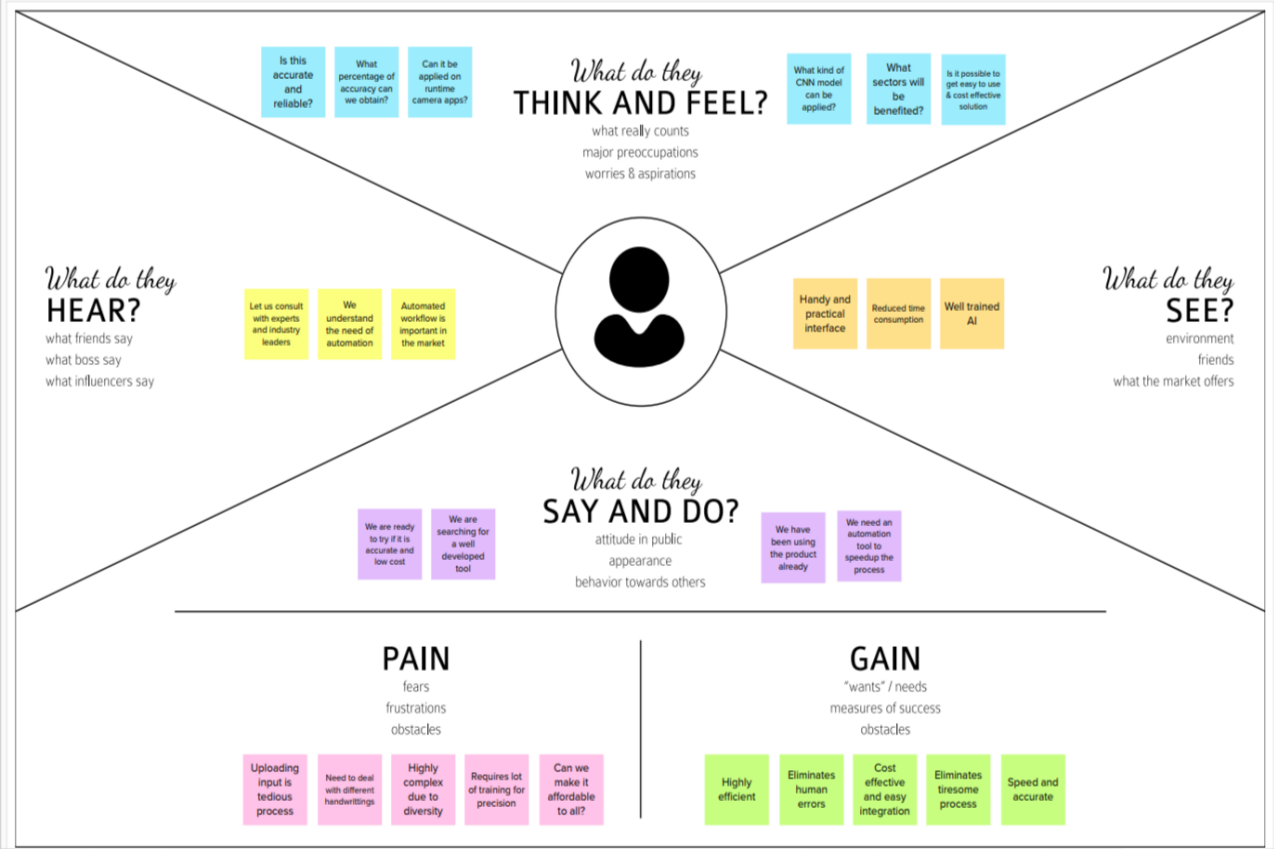
recognition. This led them to the conclusion that CNN is the most effective solution for all types of prediction issues, including those using picture data. Next, by comparing the execution times of the algorithms, they determined that increasing the number of epochs without changing the configuration 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.

CHAPTER 3
IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming

1

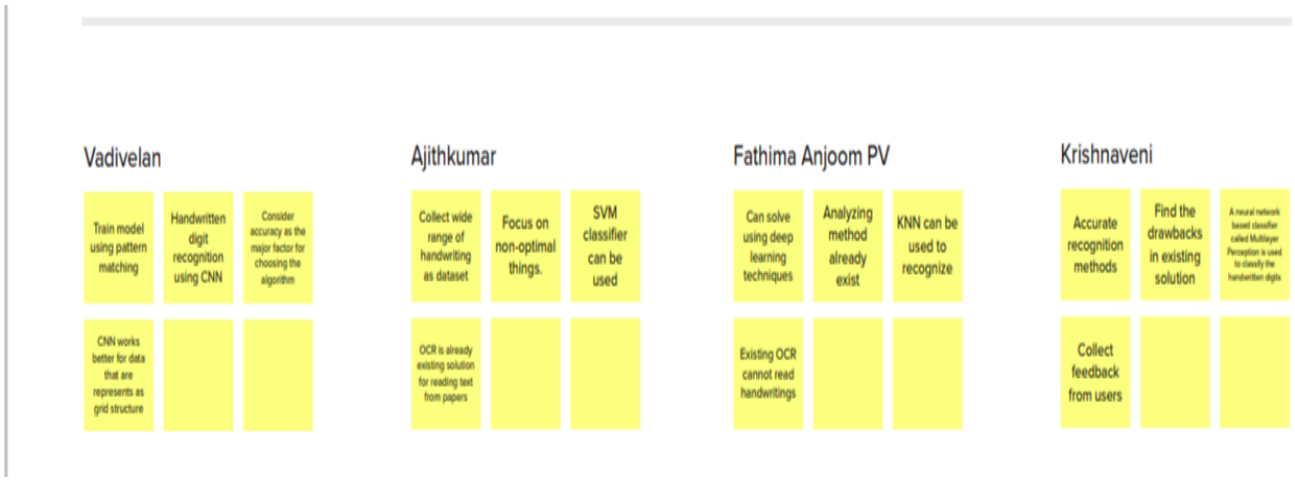
Define your problem statement

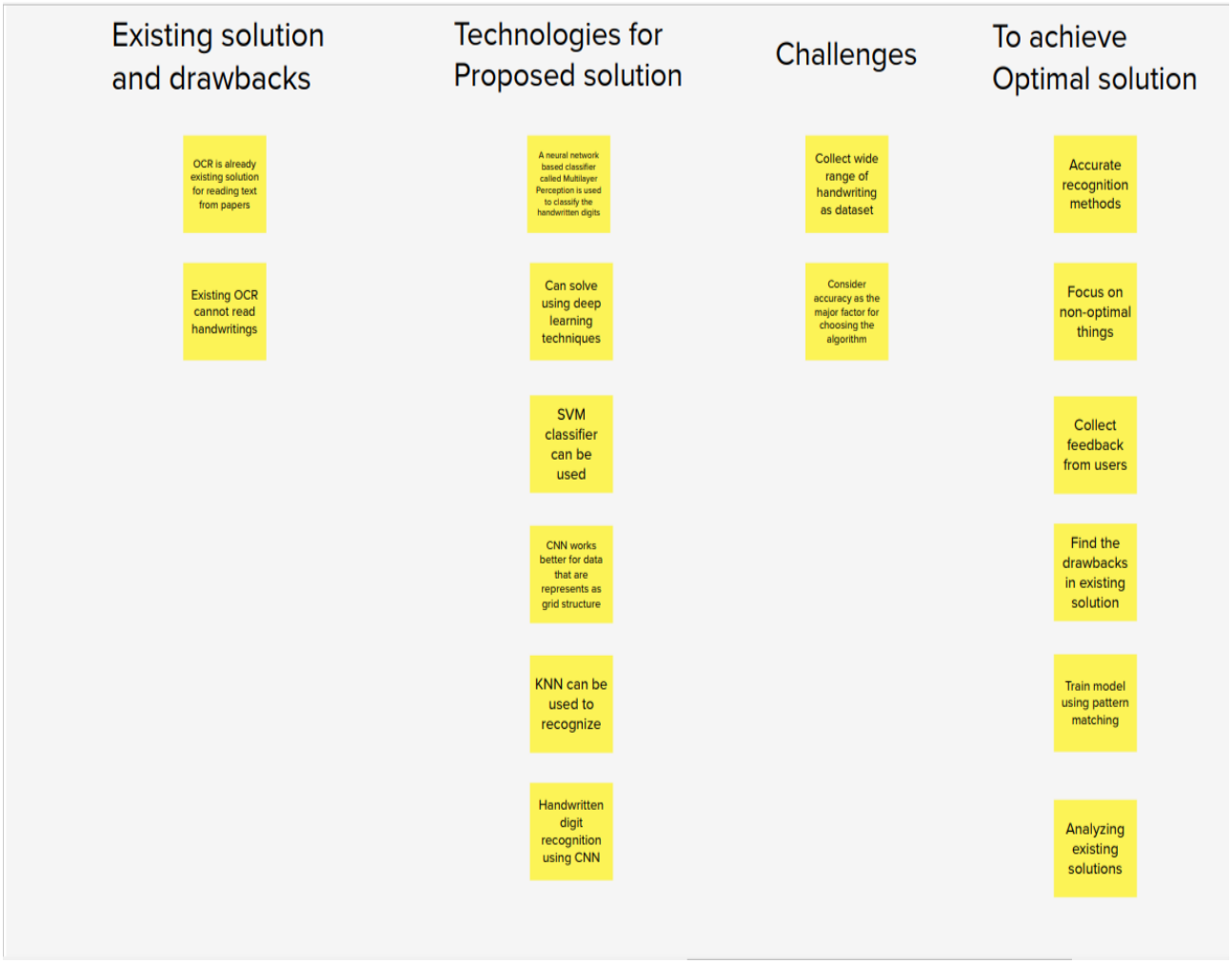
What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

5 minutes

PROBLEM

Recognize complex handwritten digits with time and cost effective.


Vadivelan
Train model using pattern matching
Handwritten digit recognition using CNN
Consider accuracy as the major factor for choosing the algorithm
CNN works better for data that are represents as grid structure
Ajithkumar
Collect wide range of handwriting as dataset
Focus on non-optimal things.
SVM classifier can be used
OCR is already existing solution for reading text from papers
Fathima Anjoom PV
Can solve using deep learning techniques
Analyzing method already exist
KNN can be used to recognize
Existing OCR cannot read handwritings
Krishnaveni
Accurate recognition methods
Find the drawbacks in existing solution
A neural network based classifier called Multilayer Perception is used to classify the handwritten digits
Collect feedback from users

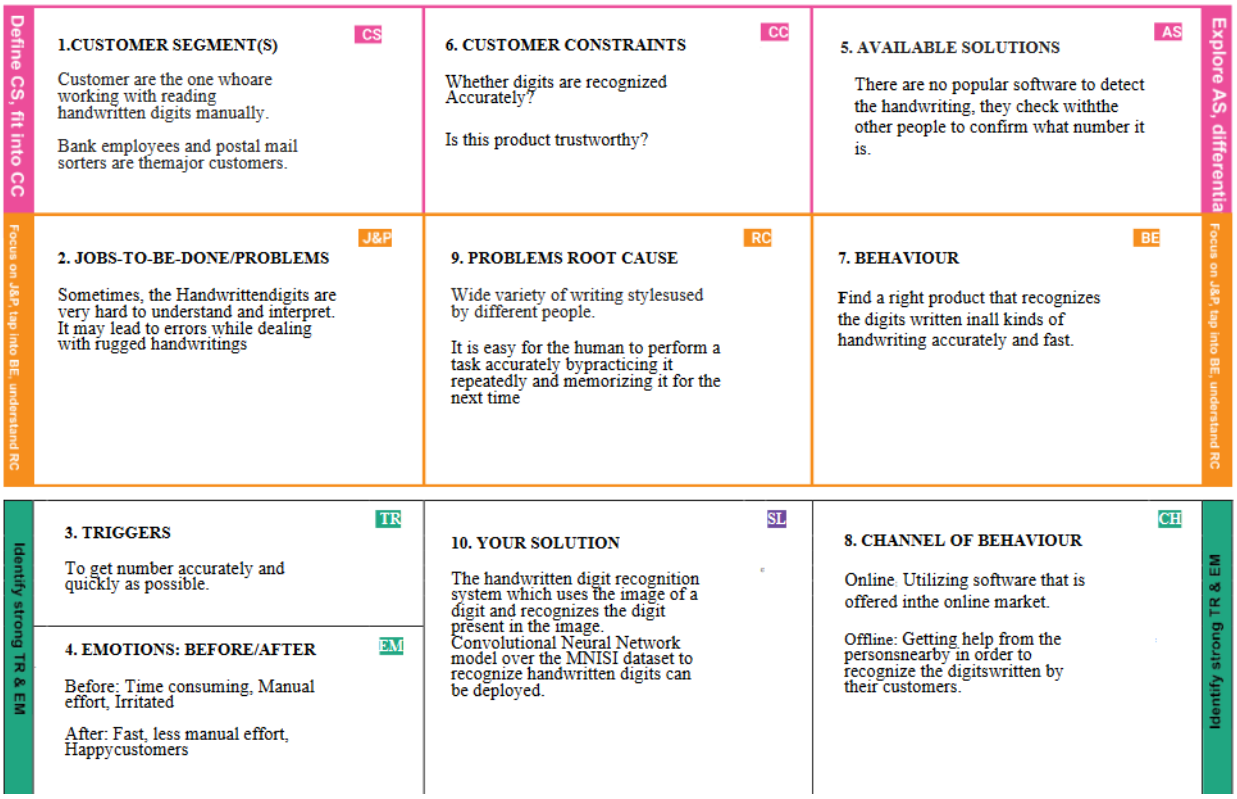

Existing solution and drawbacks
OCR is already existing solution for reading text from papers
Existing OCR cannot read handwritings
Technologies for Proposed solution
A neural network based classifier called Multilayer Perception is used to classify the handwritten digits
Can solve using deep learning techniques
SVM classifier can be used
CNN works better for data that are represents as grid structure
KNN can be used to recognize
Handwritten digit recognition using CNN
Challenges
Collect wide range of handwriting as dataset
Consider accuracy as the major factor for choosing the algorithm
To achieve Optimal solution
Accurate recognition methods
Focus on non-optimal things
Collect feedback from users
Find the drawbacks in existing solution
Train model using pattern matching
Analyzing existing solutions


Collect feedback from users
Find the drawbacks in existing solution
KNN can be used to recognize
SVM classifier can be used
Analyzing existing solutions
Focus on non-optimal things
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.

3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	To identify handwritten digits.
2.	Idea / Solution description	Create a application like “Digit4u” which gets photo of the handwritten digits as input, recognize digits using IBM datasets and displays the recognized digit as output.
3.	Novelty / Uniqueness	Handwritten digits of various handwriting can be recognized.
4.	Social Impact / Customer Satisfaction	Elimination of recognition of digits in sectors such as banks, postal code and transports
5.	Business Model (Revenue Model)	1. Google ads – Ads can be displayed in the application 2. Subscription – Subscription can be provided to access specific features.
6.	Scalability of the Solution	Furtherly this application can be modified to recognize all sort of text of all languages and can be created as a web extension.

3.4 Problem Solution Fit



CHAPTER 4
REQUIREMENT ANALYSIS

4.1 Functional Requirement

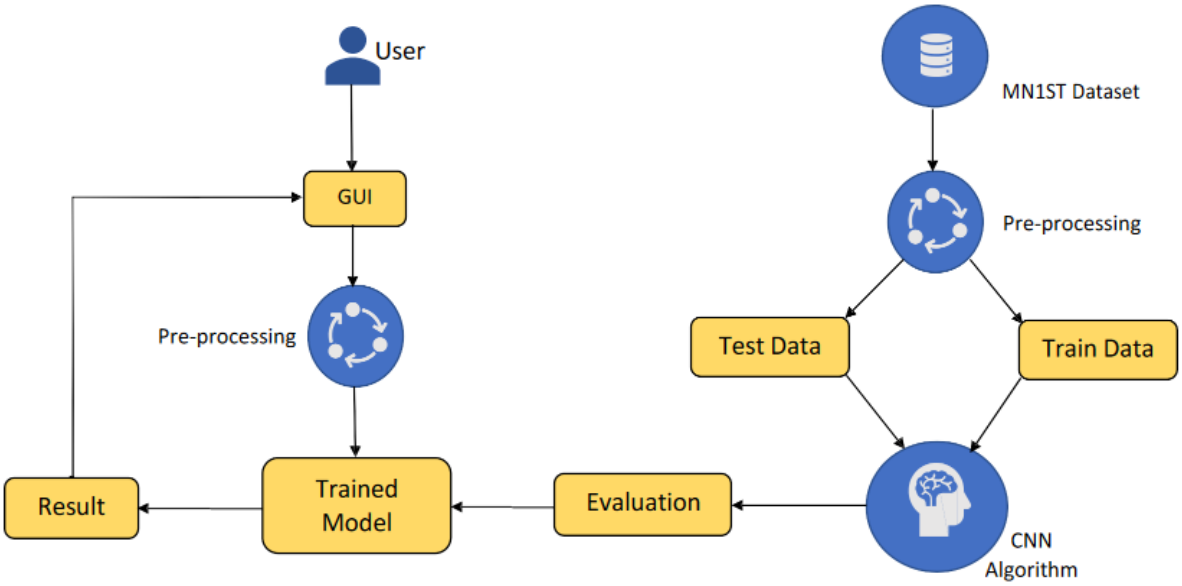
FR No.	Functional Requirement (Epic)	Sub Requirement (Story | Sub-Task)
FR-1	User Input	GUI allows the user to input image by browsing the device storage
FR-2	Model	The MNIST dataset should be trained using CNN to create a trained model
FR-3	Prediction	The trained model has to be tested by using the test data provided by MNIST and the accuracy of the model should be above 90%
FR-4	Evaluation	Ensure that the output produced by the model is correct

4.2 Non-Functional Requirement

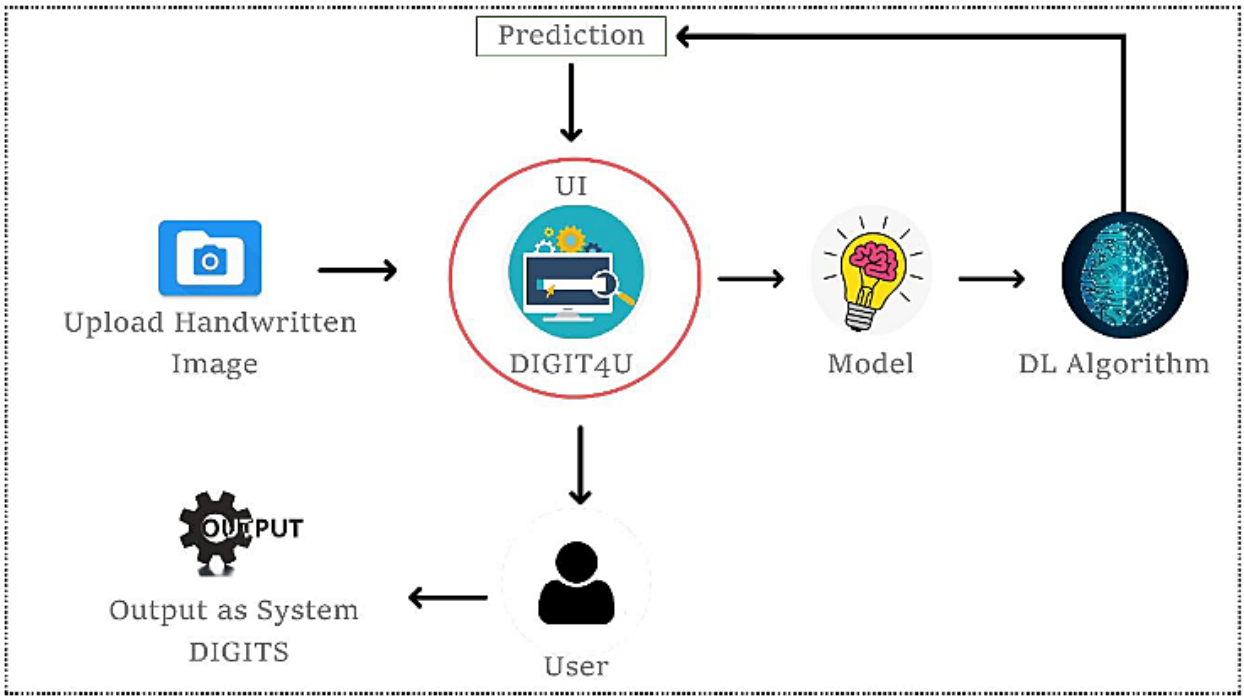
FR No.	Non-Funcational Requirement	Description
NFR-1	Usability	Can predict digits with accuracy. The model can be used in bank check processing, data entry etc
NFR-2	Security	It ensures security as the uploaded image is not stored in any database
NFR-3	Reliability	Can process confidential information without data leakage as the data is never stored in any database.
NFR-4	Performance	Improvement in fast prediction. We use CNN algorithm for accurate prediction
NFR-5	Availability	Available for web and mobile browsers
NFR-6	Scalability	Helps many individuals with low time consumption and high accuracy

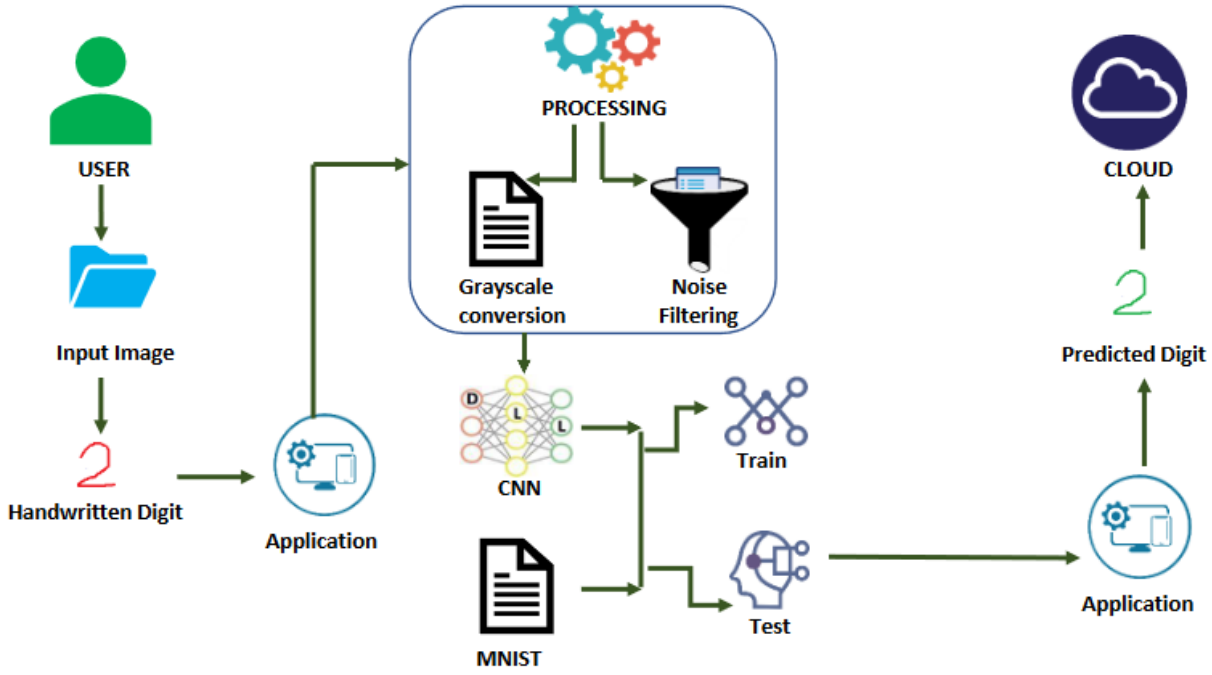
CHAPTER 5
PROJECT DESIGN

5.1 Data Flow Diagrams



5.2 Solution & Technical Architecture





5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User StoryI Task	Acceptance criteria	Priority	Release
Customer (Webuser)	Home	USN-1	In the Home Page, I can view the guidelines of how to use the website	I can view the guidelines	low	Sprint-1
	Dashboard	USN-2	As a user,I can see Home Page & Prediction Page	I can access the dashboard	Low	Sprint-2
	Choose Input	USN-3	In Prediction Page, I can upload an image of handwritten digit for prediction	I can upload my input by browsing the device storage	Medium	Sprint-3
		USN-4	As a user, I can get an accuracy rate with the prediction	I can get different forms of output	High	Sprint-4

	Recognize	USN-5	As a user,I can see that the GUI processing the input using trained model	I can perform handwrittendigit prediction	High	Sprint-1
	Prediction	USN-6	As a user, I can get accuracy rate by pressing the predict button	I can get the accuracy ofthe output	Medium	Sprint-1
Customer (Mobile user)	Home	USN-7	As a user, I can access application in mobilephone	I can access the dashboard with mobile	Medium	Sprint-1
	Recognize	USN-8	I can upload input and retrieve output with accuracy by using the mobile	I can upload input image and get output with a mobile device	High	Sprint-2

CHAPTER 6
PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & 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	Fathima Anjoom PV
Sprint-1	Data Preprocessing	USN-2	As a user, I can load the dataset, handling the missing data, scaling and split data into train and test.	10	Medium	Ajithkumar N, Krishnaveni G
Sprint-2	Model Building	USN-3	As a user, I will get an application with ML model which provides high accuracy of recognized handwritten digit.	5	High	Vadivelan A, Ajithkumar N, Fathima Anjoom PV, Krishnaveni G
Sprint-2	Add CNN layers	USN-4	Creating the model and adding the input, hidden, and output layers to it.	5	High	Vadivelan A, Ajithkumar N, Fathima Anjoom PV, Krishnaveni G
Sprint-2	Compiling the model	USN-5	With both the training data defined and model defined, it's time to configure the learning process.	2	Medium	Vadivelan A, Krishnaveni G
Sprint-2	Train & test the model	USN-6	As a user, let us train our model with our image dataset.	6	Medium	Ajithkumar N, Fathima Anjoom PV

Sprint-2	Save the model	USN-7	As a user, the model is saved & integratedwith an android application or web application in order to predict something.	2	Low	Vadivelan A
Sprint-3	Building UI Application	USN-8	As a user, I will upload the handwritten digit image to the application by clicking a upload button.	10	High	Vadivelan A, Ajithkumar N
Sprint-3		USN-9	As a user, I can know the details of thefundamental usage of the application.	5	Low	Krishnaveni G
Sprint-3		USN-10	As a user, I can see the predicted / recognized digits in the application.	5	Medium	Fathima Anjoom PV, Krishnaveni G
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	Vadivelan A, Ajithkumar N, Fathima Anjoom PV, Krishnaveni G
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	Vadivelan A, Ajithkumar N, Fathima Anjoom PV

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

7.1 Feature 1

The application have the most important feature that gets image of handwritten digit from user, recognize the digit and displays the predicted digit as output on UI.

Code:

```
img = Image.open(request.files['file'].stream).convert('L')
img = img.resize((28, 28))
im2arr = np.array(img)
im2arr = im2arr.reshape(1, 28, 28, 1)
prediction = model.predict(im2arr)
y_pred = np.argmax(prediction)
```

7.2 Feature 2

The application have feature that display the probability percentage of correctness of predicted digit.

Code:

```
prediction_percentage = str(round(max(prediction[0])*100, 2))+"%"
```

CHAPTER 8
TESTING

8.1 Testcases

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Commnets	TC for Automation(Y/N)	BUG ID	Executed By
Home Page_TC_OO1	Functional	Home Page	Verify user is able to see the Home is displayed or not	Internet Connection	1.Enter URL and click go	http://127.0.0.0:8000	Home Page should display	Working as expected	Pass	Steps are clear to follow	No	No Bug	Ajith kumar N
Home Page_TC_OO2	UI	Home Page	Verify the UI elements in Home page Shown	Web application need browser to run	1.Enter URL and click go 2.Verify Home page shown with below UI elements: Home button Recognize button	http://127.0.0.0:8000	Application should show below UI elements: A.Recognize button B.Home button	Working as expected	Pass	Steps are clear to follow	No	No Bug	Fathima Anjoom PV
Home Page_TC_OO3	Functional	Home page	Verify user is able to go to the recognize page	Open to browser	1.Enter URL(http://127.0.0.0:8000/) and click go 2.Click on Recognize button	http://127.0.0.0:8000	User should redirected to Recognize from homepage	Working as expected	Pass	Steps are clear to follow	No	No Bug	Krishnaveni G
Recognize Page_TC_OO4	UI	Recognize page	Verify the UI elements in Recognize page Shown	Web application need browser to run	1.Enter URL(http://127.0.0.0:8000/) and click go 2.Click on Recognize button 3.Page go to Recognize page	http://127.0.0.0:8000	Recognize page should show below UI elements: A.Recognize button with Blue colour B.Choose button with Green colour	Working as expected	Pass	Steps are clear to follow	No	No Bug	Vadivelan A
Recognize Page_TC_OO5	Functional	Choose Button	Verify user is able to click choose button and choose image	Internet Connection	1.Enter URL(http://192.168.161.52:8000) and click go 2.Click on Recognize button 3.Enter Recognize page then click choose button 4.My Computer console open then select user data 5.Click on select button	http://127.0.0.0:8000	Input image should chosen. After chosen status text show the chosen file name	Working as expected	Pass	Steps are clear to follow	No	No Bug	Ajith kumar N
Recognize Page_TC_OO6	Functional	Recognize page	Verify user is able to go to the recognize page	Internet Connection	1.Enter URL(http://127.0.0.0:8000/) and click go 2.Click on Recognize button	http://127.0.0.0:8000	Recognize page should display	Working as expected	Pass	Steps are clear to follow	No	No Bug	Fathima Anjoom PV
Recognize Page_TC_OO7	Functional	Recognize Button	Verify user is able to click choose button and choose image	Internet Connection	1.Enter URL(http://192.168.161.52:8000) and click go 2.Click on Recognize button 3.Enter Recognize page then Enter Recognize page then click choose button 4.My Computer console open then select user data 5.Click on select button 6.click recognize button	http://127.0.0.0:8000	If user not choose any file display "No File Selected" else redirect to result page	Working as expected	Pass	Steps are clear to follow	No	No Bug	Krishnaveni G
Result Page_TC_OO8	UI	Result page	Verify the UI elements in Result page Shown	Web application need browser to run	1.Enter URL(http://192.168.161.52:8000) and click go 2.Click on Recognize button 3.Enter Recognize page then Enter Recognize page then click choose button 4.My Computer console open then select user data 5.Click on select button 6.click recognize button	http://127.0.0.0:8000	Recognize page should show below UI elements: A.Recognize button with Blue colour B.Choose button with Green colour, Recognized digit as output, Probability percentage and user chosen image	Working as expected	Pass	Steps are clear to follow	No	No Bug	Vadivelan A
Result Page_TC_OO9	Functional	Choose Button	Verify user is able to click choose button and choose image	Internet Connection	1.Click choose button 2.My computer console open then select user data 3.Click on select button	http://127.0.0.0:8000	Input image should chosen. After chosen status text show the chosen file name	Working as expected	Pass	Steps are clear to follow	No	No Bug	Ajith kumar N
Result Page_TC_O10	Functional	Recognize Button	Verify user is able to click choose button and choose image	Internet Connection	1.click recognize button	http://127.0.0.0:8000	If user not choose any file display "No File Selected" else redirect to result page	Working as expected	Pass	Steps are clear to follow	No	No Bug	Vadivelan A

8.2 User Acceptance Testing

1. Purpose of Document

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

2. Defect Analysis

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

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	9	6	2	0	17
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	6	2	4	20	32
Not Reproduced	0	0	1	0	1
Skipped	0	0	0	1	1
Won't Fix	0	0	2	1	3
Totals	18	11	12	23	64

3. Test Case Analysis

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

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	4	0	0	4
Client Application	8	0	0	8
Security	2	0	0	2
Outsource Shipping	1	0	0	1
Exception Reporting	7	0	1	6
Final Report Output	4	0	0	4
Version Control	2	0	0	2

CHAPTER 9
RESULTS

9.1 Performance Matrics

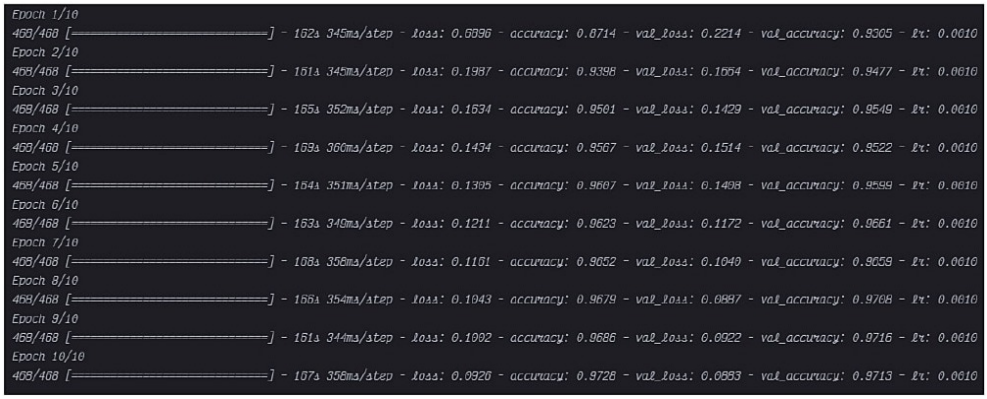
Model Performance Testing:

S.No.	Parameter	Values	Screenshot
1.	Model Summary	Total params: 203,434 Trainable params: 203,434 Non-trainable params: 0	See image 1(Below Attached)
2.	Accuracy	Training Accuracy - 0.9728 Validation Accuracy – 0.9713	See Image 2(Below Attached)
3.	Confidence Score (Only Yolo Projects)	Class Detected - 10 Confidence Score - 7	

Image 1:

```
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
_________________________________________________________________
```

Image 2:


Epoch 1/10
468/468 [==========] - 162s 345ms/step - loss: 0.0896 - accuracy: 0.8714 - val_loss: 0.2214 - val_accuracy: 0.9305 - lr: 0.0010
Epoch 2/10
468/468 [==========] - 161s 345ms/step - loss: 0.1987 - accuracy: 0.9398 - val_loss: 0.1654 - val_accuracy: 0.9477 - lr: 0.0010
Epoch 3/10
468/468 [==========] - 165s 352ms/step - loss: 0.1634 - accuracy: 0.9501 - val_loss: 0.1429 - val_accuracy: 0.9549 - lr: 0.0010
Epoch 4/10
468/468 [==========] - 169s 360ms/step - loss: 0.1434 - accuracy: 0.9567 - val_loss: 0.1514 - val_accuracy: 0.9522 - lr: 0.0010
Epoch 5/10
468/468 [==========] - 164s 351ms/step - loss: 0.1305 - accuracy: 0.9607 - val_loss: 0.1498 - val_accuracy: 0.9599 - lr: 0.0010
Epoch 6/10
468/468 [==========] - 163s 349ms/step - loss: 0.1211 - accuracy: 0.9623 - val_loss: 0.1172 - val_accuracy: 0.9661 - lr: 0.0010
Epoch 7/10
468/468 [==========] - 168s 358ms/step - loss: 0.1151 - accuracy: 0.9652 - val_loss: 0.1049 - val_accuracy: 0.9659 - lr: 0.0010
Epoch 8/10
468/468 [==========] - 166s 354ms/step - loss: 0.1043 - accuracy: 0.9679 - val_loss: 0.0887 - val_accuracy: 0.9708 - lr: 0.0010
Epoch 9/10
468/468 [==========] - 161s 344ms/step - loss: 0.1092 - accuracy: 0.9686 - val_loss: 0.0922 - val_accuracy: 0.9716 - lr: 0.0010
Epoch 10/10
468/468 [==========] - 167s 356ms/step - loss: 0.0926 - accuracy: 0.9728 - val_loss: 0.0883 - val_accuracy: 0.9713 - lr: 0.0010

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 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 97.13% recognition rate. The proposed solution 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 from digits multiple images and save the results
- Add support to detect multiple digits
- Improve model to detect digits from complex images

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

CHAPTER 13
APPENDIX

Source code

Model Creation

```
from keras.utils import np_utils
from tensorflow.keras.datasets import mnist #MNIST dataset

(X_train, Y_train), (X_test, Y_test) = mnist.load_data()

X_train = X_train.reshape(60000,28,28,1).astype('float32')
X_test = X_test.reshape(10000,28,28,1).astype('float32')

number_of_classes = 10
Y_train = np_utils.to_categorical(Y_train, number_of_classes)
Y_test = np_utils.to_categorical(Y_test,number_of_classes)

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Convolution2D,MaxPooling2D,Flatten,Dense
model = Sequential()
model.add(Convolution2D(64, (3,3), input_shape=(28,28,1),activation='relu'))
model.add(Convolution2D(32,(3,3),activation='relu'))
model.add(Flatten())
model.add(Dense(number_of_classes, activation='softmax'))

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

model.fit(X_train, Y_train, batch_size=32, epochs=5,
validation_data=(X_test,Y_test))

metrics = model.evaluate(X_test, Y_test, verbose=0)

prediction=model.predict(X_test[:4])
import numpy as np

model.save("models/mnistCNN.h5")

from tensorflow.keras.models import load_model
model = load_model(r'/content/models/mnistCNN.h5')
from PIL import Image
for index in range(4):
  img = Image.open('/content/models/sample3.png').convert("L")
  img = img.resize((28,28))
  im2arr = np.array(img)
  im2arr = im2arr.reshape(1,28,28,1)
```

```
  y_pred = model.predict(im2arr)
```

app.py

```
from flask import Flask, render_template, request
from PIL import Image
import numpy as np
from tensorflow import keras
from keras.models import load_model
import tensorflow as tf
import os
from io import BytesIO
import pybase64
model = load_model('models/mnistCNN.h5')
app = Flask(__name__)
@app.route('/')
def upload_file():
    return render_template('main.html')
@app.route('/about')
def upload_file1():
    return render_template('main.html')
@app.route('/upload')
def upload_file2():
    return render_template('index6.html')
@app.route('/predict', methods=['POST'])
def upload_image_file():
    if request.method == 'POST':
        file = request.files['file']
        img = Image.open(request.files['file'].stream).convert('L')
        img = img.resize((28, 28))
        im2arr = np.array(img)
        im2arr = im2arr.reshape(1, 28, 28, 1)
        prediction = model.predict(im2arr)
        y_pred = np.argmax(prediction)
        #prediction_percentage = str(max(list(map(lambda x: round(x*100, 2),
prediction[0]))))+"%"
        prediction_percentage = str(round(max(prediction[0])*100, 2))+"%"
        #y_pred = pd.Series(prediction,name="Label")
        filename = file.filename
        path = os.path.join("static/images", filename)
        img = Image.open(file.stream)
        file.save(path)
        #encoded_string = pybase64.b64encode(open(path, "rb").read()).decode('UTF-8')
        if filename.endswith('jpg') or filename.endswith('jpeg'):
            with BytesIO() as buf:
                img.save(buf, 'jpeg')
                image_bytes = buf.getvalue()
            encoded_string = pybase64.b64encode(image_bytes).decode()
            encoded_string = ""+encoded_string
        if filename.endswith('png'):
```

```
            with BytesIO() as buf:
                img.save(buf, 'png')
                image_bytes = buf.getvalue()
            encoded_string = pybase64.b64encode(image_bytes).decode()
            encoded_string = ""+encoded_string
        os.remove(path)
        if (y_pred == 0 or y_pred == 1 or y_pred == 2 or y_pred == 3 or y_pred == 4 or
y_pred == 5 or y_pred == 6 or y_pred == 7 or y_pred == 8 or y_pred == 9):
            return render_template("result.html", digit=y_pred,
user_image=encoded_string, percentage=prediction_percentage, showcase=str(y_pred))
        else:
            return render_template("result.html", digit="No digit found.",
user_image=encoded_string, percentage=prediction_percentage)
    else:
        return None
if __name__ == '__main__':
    app.run(host='0.0.0.0', port=8000, debug=True)
```

main.html

```
<html>
  <head>
    <title>Handwritten Digit Recognition System</title>
    <link rel="stylesheet" href="../static/style.css">
    <style>
      body{
        background-image:url("../static/images/onebg.jpg");
      }
    </style>
    <script>
      window.onload = function(){
        document.getElementById("loader").style.display="none";
        document.getElementById("content").style.visibility="visible";
      }
    </script>
  </head>
<body>
  <span id="loader">
    <img src="../static/loading.gif">
  </span>
  <span id="content">
  <div class="header">
    <p><a href="/">Home</a><a href="/upload">Recognize</a></p>
  </div>
  <h1>Handwritten Recognition System</h1>
  <p class="introduction">Handwritten Text Recognition is a Technology that is much needed
in this world as of today.
    The digit recognition system is used to recognize the digits from different sources like
email, bank cheque, papers, images, etc.
    Before proper implementation of this technology we have relied on writting texts with
our own hands which can result in errors.
    It's diffcult to store and access physical data with effiecieny.
```

The project presents recognizing the handwritten digits (0 to 9) from the famous MNIST dataset.

Here we will be using artificial neural networks convolution neural networks.

```
</p>
</span>
</body>
</html>
```

index6.html

```
<html>
<head>
   <title>Handwritten Digit Recognition System</title>
   <link rel="stylesheet" href="../static/style.css">
   <script src="../static/script.js"></script>
   <style>
      body{
         background-image:url("../static/images/twobg.jpg");
         background-repeat:no-repeat;
         background-position:bottom;
      }
   </style>
</head>
<body>
   <span id="loader">
      <img src="../static/loading.gif">
   </span>
   <span id="content">
   <div class="header">
      <p><a href="/">Home</a><a href="/upload">Recognize</a></p>
   </div>
   <h2><strong>Digit Recognition</strong></h2>
   <div class="container">
   <form class="form" action="/predict" method="POST" enctype="multipart/form-data">
   <input type="file" name="file" id="upload" style="display:none"  accept="image/x-
png,image/jpeg" required>
   <label for="upload" class="upload">Choose</label>
   <input type='submit' class="recognize" value="Recognize" id="submit">
   <p id="status"></p>
   </form>
   </div></span>
</body>
</html>
```

result.html

```
<html>
<head>
   <title>Handwritten Digit Recognition System</title>
   <link rel="stylesheet" href="../static/style.css">
   <script src="../static/script.js"></script>
   <style>
      body{
         background-image:url("../static/images/twobg.jpg");
```

```
        background-repeat:no-repeat;
        background-position:bottom;
      }
    </style>
</head>
<body>
    <span id="loader">
      <img src="../static/loading.gif">
    </span>
    <span id="content">
    <div class="header">
      <p><a href="/">Home</a><a href="/upload">Recognize</a></p>
    </div>
    <h2><strong>Digit Recognition</strong></h2>
    <div class="container">
    <span>
    <form class="form" action="/predict" method="POST" enctype="multipart/form-data">
    <input type="file" name="file" id="upload" style="display:none"  accept="image/x-
png,image/jpeg" required>
    <label for="upload" class="upload">Choose</label>
    <input type='submit' class="recognize" value="Recognize" id="submit">
    <p id="status"></p>
    </form>
    </span>
    <div><h3>Recognized digit: {{digit}}</h3>
      <h4>Probability: {{percentage}}</h4>
      <img src="{{ user_image }}" alt="User Image" class="user-image"></div>
    </div>
    </span>
</body>
</html>
```

style.css

```
#loader{
    display:flex;
    justify-content:center;
    align-items:center;
    height:100%;
}
#content{
    visibility:none;
}
.header{
    width:100%;
    height:auto;
    display:flex;
    justify-content:flex-end;
    background-color:rgba(255,255,255,.7);
    border-radius:10px;
    -webkit-border-radius:10px;
    -moz-border-radius:10px;
    -ms-border-radius:10px;
    -o-border-radius:10px;
}
```

```
.header a{
    color:black;
    font-size: 15pt;
    padding:10px;
    text-decoration:none;
}
.header a:hover{
    border-bottom: 2px black solid;
}
h1, h2, .introduction, .form{
    text-align:center;
}
h1, .introduction{
    color:white;
}
h2{
    color:red;
}
.introduction{
    font-size:15pt;
}
.upload{
    background-color:rgb(55,210,180);
    border-radius:5px;
    cursor:pointer;
    -webkit-border-radius:5px;
    -moz-border-radius:5px;
    -ms-border-radius:5px;
    -o-border-radius:5px;
}
.upload:hover{
    background-color:rgb(0, 136, 109);
}
.recognize{
    background-color:rgb(3,120,255);
    border:none;
    cursor:pointer;
}
.recognize:hover{
    background-color:rgb(0, 72, 155);
}
.upload, .recognize{
    padding:10px;
    color:white;
    border-radius:5px;
    -webkit-border-radius:5px;
    -moz-border-radius:5px;
    -ms-border-radius:5px;
    -o-border-radius:5px;
}
.container{
    display:flex;
    width:100%;
    justify-content:space-around;
```

```
}
.user-image{
    width:150px;
    height:150px;
}
#status{
    color:red;
    visibility:hidden;
}
```

script.js

```
window.onload = function () {
  document.getElementById("loader").style.display="none";
  document.getElementById("content").style.visibility="visible";
  input = document.getElementById("upload");
  submit = document.getElementById("submit");
  statustxt = document.getElementById("status");
  input.addEventListener("change", function () {
    if (input.value == "") statustxt.innerText = "No file selected";
    else statustxt.innerText = input.files[0].name;
    statustxt.style.visibility="visible";
  });
  submit.addEventListener('click',function(){
    if (input.value == "") statustxt.innerText = "No file selected";
    else statustxt.innerText = "Working on it...";
    statustxt.style.visibility="visible";
  });
};
```

Github & Project Demo Link

Github Link: https://github.com/IBM-EPBL/IBM-Project-52354-1660998080

Project Demo Link:
https://drive.google.com/file/d/1EL2spqi9RLnFiSx4I4D4MtcZQ6jh7Uk5/view?usp=share_link