A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

PROJECT REPORT

Submitted by

TEAM ID:PNT2022TMID18280

TEAM MEMBERS

TEAM LEAD : Snowlin Faustina

TEAM MEMBER 1 : Arthi G

TEAM MEMBER 2 : Shalini S

TEAM MEMBER 3 : Nivetha A

TABLE OF CONTENTS

1. INTRODUCTION 5

1.1 Project Overview 5

1.2 Purpose 5

2. LITERATURE SURVEY 6

2.1 Existing problem 6

2.2 References 6

2.3 Problem Statement Definition 7

3. IDEATION & PROPOSED SOLUTION 8

3.1 Empathy Map Canvas 8

3.2 Ideation and Brainstorming 9

3.3 Proposed Solution 10

3.4 Problem Solution fit 11

3.5 Proposed Model 12

4. REQUIREMENT ANALYSIS 14

4.1 Functional requirement 14

4.2 Non-Functional requirements 14

5. PROJECT DESIGN 16

5.1 Data Flow Diagrams 16

5.2 Solution & Technical Architecture 17

5.3 User Stories 18

6. PROJECT PLANNING & SCHEDULING 19

6.1 Sprint Planning And Estimation 19

6.2 Sprint Delivery Schedule 20

7. CODING & SOLUTIONING 21

7.1 Database Schema 25

8. TESTING 26

8.1 Test Cases 26

8.2 User Acceptance Testing 27

8.2.1 Defect Analysis 27

8.2.2 Test Case Analysis 28

9. RESULTS 29

9.1 Performance Metrics 29

10. ADVANTAGES & DISADVANTAGES 30

11. CONCLUSION 32

12. FUTURE SCOPE 33

13. APPENDIX **34**

Source code 34

Github Link 37

Demo Link 37

`

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

Handwritten Digit Recognition is the capacity of a computer to interpret the manually written digits from various sources like messages, bank cheques, papers, pictures, and so forth and in various situations for web based handwriting recognition on PC tablets, identifying number plates of vehicles, handling bank cheques, digits entered in any forms. We use artificial neural network to train these images and build a model. Web application is created where the user can upload an image of a handwritten digit. This image is analyzed by the model and the detected result is returned on to the UI.

1.2 PURPOSE

The objective of this project is to identify the handwritten digits given by the user. The numbers can be from 0-9.

`

CHAPTER 2

LITERATURE SURVEY

2.1 EXISTING PROBLEM

These days, an ever-increasing number of individuals use pictures to transmit data. It is additionally main stream to separate critical data from pictures. Image Recognition is an imperative research area for its generally used applications. In general, the field of pattern recognition, one of the difficult undertakings is the precise computerized recognition of human handwriting. Without a doubt, this is a very difficult issue because there is an extensive diversity in handwriting from an individual to another individual. Despite the fact that, this difference does not make any issues to people, yet, anyway it is increasingly hard to instruct computers to interpret general handwriting. For the image recognition issue, it is essential to make out how information is depicted onto images. The scope of this is to implement a Handwritten Digit Recognition framework and think about the diverse classifiers and different techniques by concentrating on how to accomplish close to human performance. For an undertaking of composing diverse digits (0-9) for various people the general issue confronted would be of digit order issue and the closeness between the digits like 1 and 7, 5 and 6, 3 and 8, 9 and 8 and so forth.

2.2 REFERENCES

[1] Raghunanth Dey et al proposed A digit recognition system based on a sliding window with edit distance for handwritten digit images is proposed. It had an accuracy of 96.83% when trained with MNIST and tested on MNIST. It had an accuracy of 72.7% when trained with MNIST and tested on ARDIS.

[2] Monji Kherallah et al proposed a novel way of handwriting modeling system based on Beta-elliptic approach and a hierarchical recognition system of digits based on an association

`

of SOM, FKNN, and MLPNN. The system achieved a recognition rate of 95.08% and a SA error of 0.065%.

[3] C. Zhang et al proposed a new type of handwritten digit recognition system based on convolutional neural network (CNN). Good performance in this experiment shows that our system can automatically recognize the handwritten digital content appearing in the target area.

[4] J.Memon et al presents the state of the art results and techniques on OCR and also provide research directions by highlighting research gaps.

[5] Gaurav Surve et al proposed Handwritten digit recognition has a wide range of applications, including identifying postal codes on envelopes, processing largescale financial statements, and processing bank form input. It was often thought that people who used computers for business had to change their input style to match what the computer expected, whether they were typing or filling out forms with letters.

2.3 PROBLEM STATEMENT DEFINITION

The total world is working with the various problems of the machine learning. The goal of the machine learning is to factorize and to manipulate the real life data and the real life part of the human interaction or complex ideas or the problems in the real life. The most curious of those is Handwritten Character Recognition because it is the building block of the human certified and the classification interaction between other humans.

So, the goal was to create an appropriate algorithm that can give the output of the handwritten character by taking just a picture of that character. If one asks about Image processing then this problem can’t be solved because there can be a lot of noises in that taken image which can’t be controlled by human. The main thing is when human write a handwritten character or for our case digit he has no single idea whether he has to draw it in the circulated pixels or just same as a standard image given .A machine can do that but not the human. So by matching only the pixels one can’t recognize that.

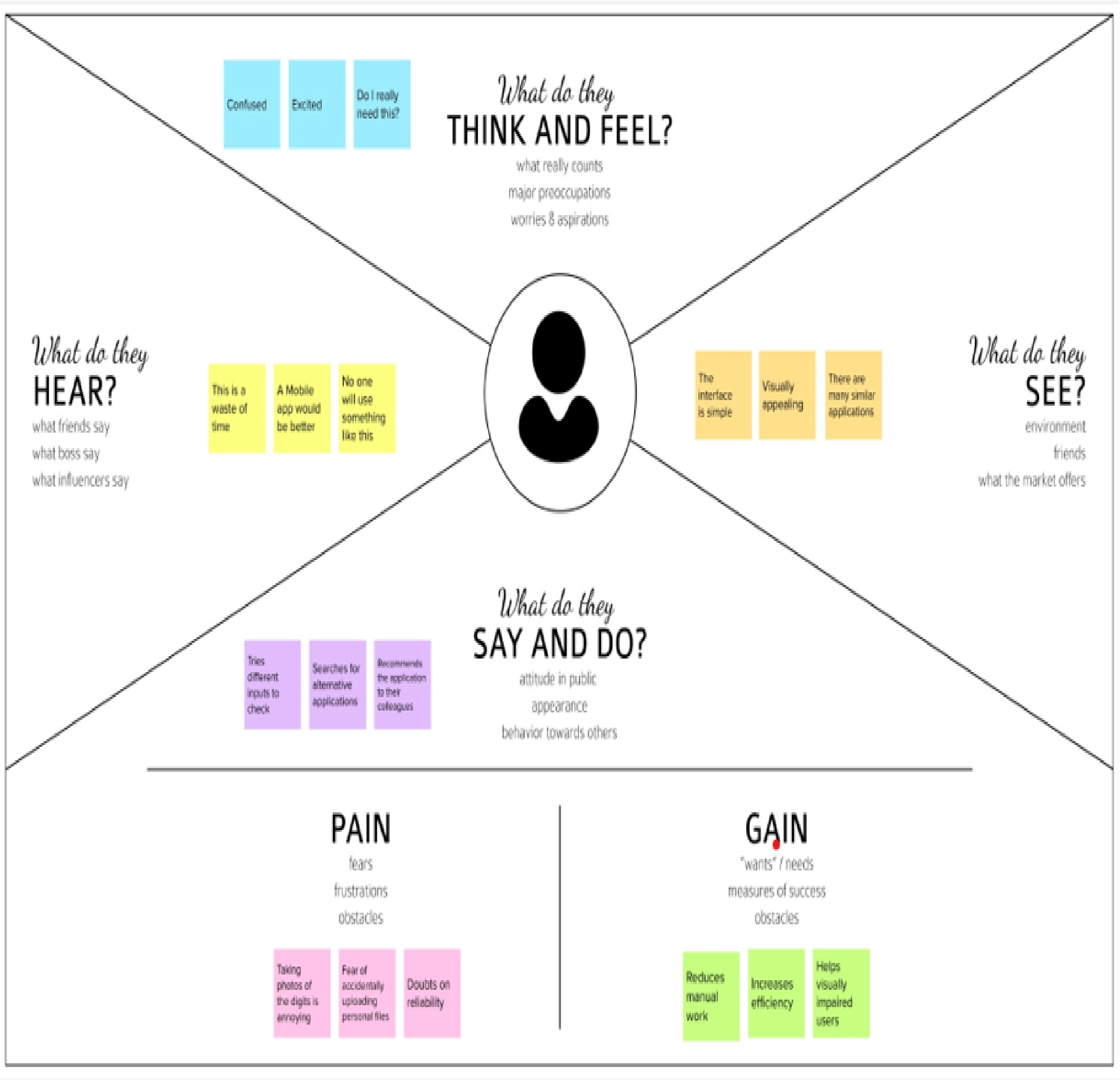
For this project one has to create a model by image processing and the machine learning. Both the techniques will be needed because these two techniques will enhance the technique of the machine learning and that can shape this project.

`

CHAPTER 3

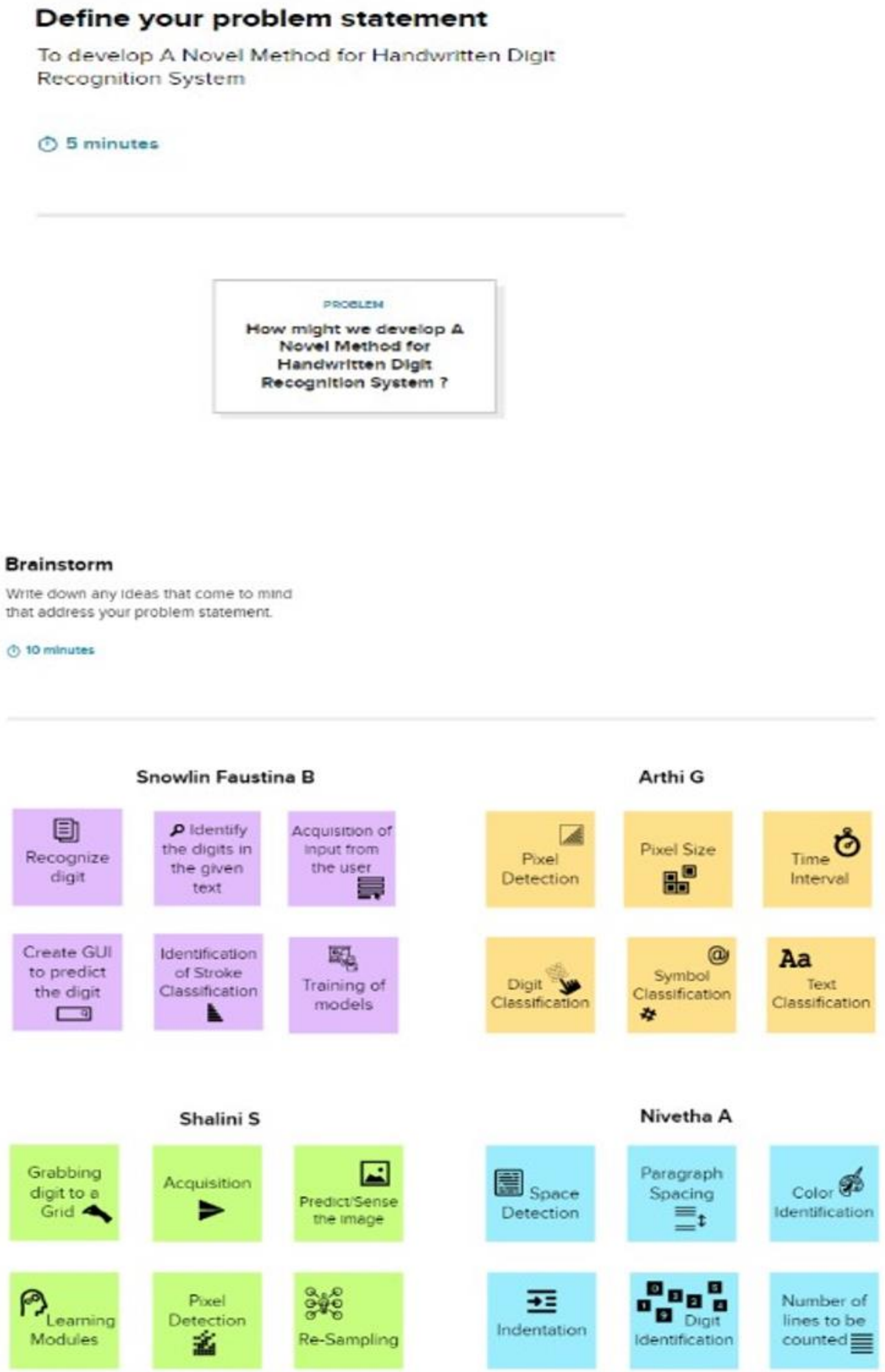
IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS



`

3.2 IDEATION AND BRAINSTORMING



Define your problem statement

To develop A Novel Method for Handwritten Digit Recognition System

5 minutes

PROBLEM

How might we develop A Novel Method for Handwritten Digit Recognition System ?

Brainstorm

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

10 minutes

Snowlin Faustina B

- Recognize digit
- Identify the digits in the given text
- Acquisition of input from the user
- Create GUI to predict the digit
- Identification of Stroke Classification
- Training of models

Arthi G

- Pixel Detection
- Pixel Size
- Time Interval
- Digit Classification
- Symbol Classification
- Aa Text Classification

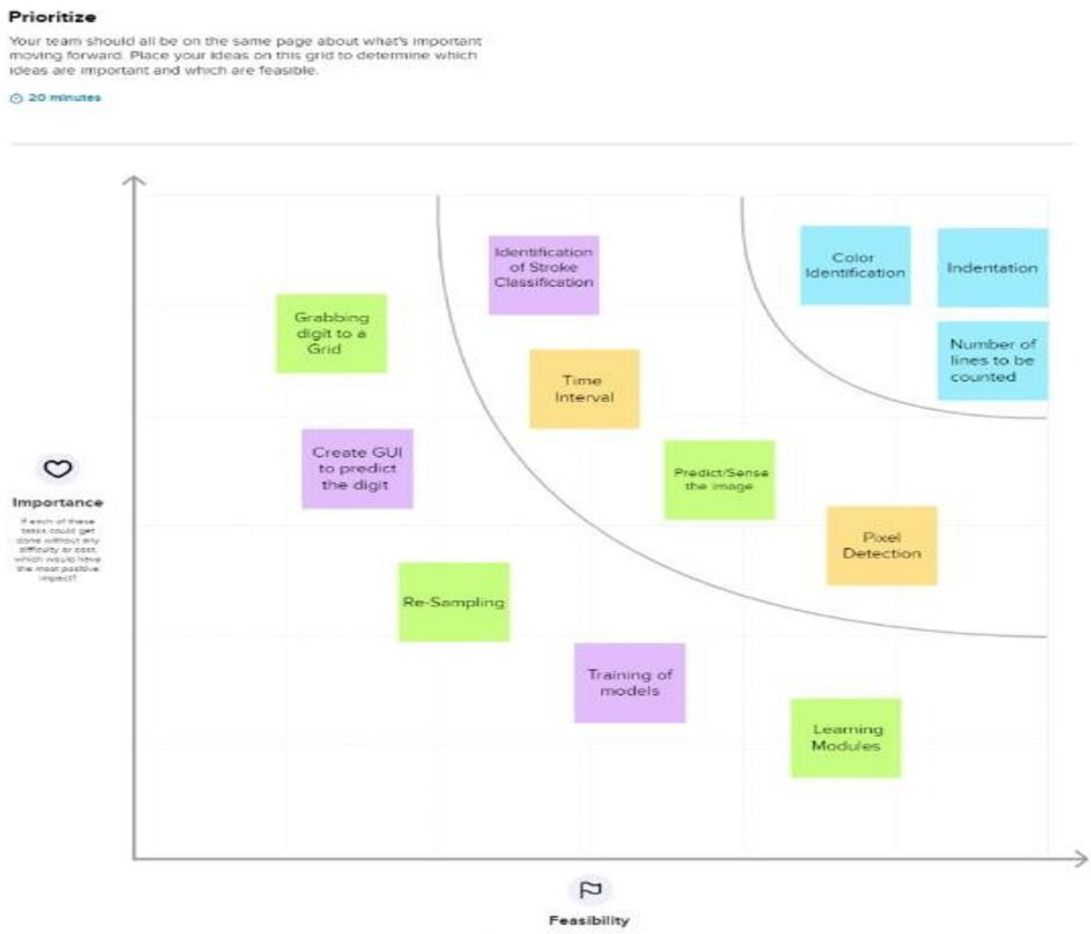
Shalini S

- Grabbing digit to a Grid
- Acquisition
- Predict/Sense the image
- Learning Modules
- Pixel Detection
- Re-Sampling

Nivetha A

- Space Detection
- Paragraph Spacing
- Color Identification
- Indentation
- Digit Identification
- Number of lines to be counted

`



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

Importance

If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?

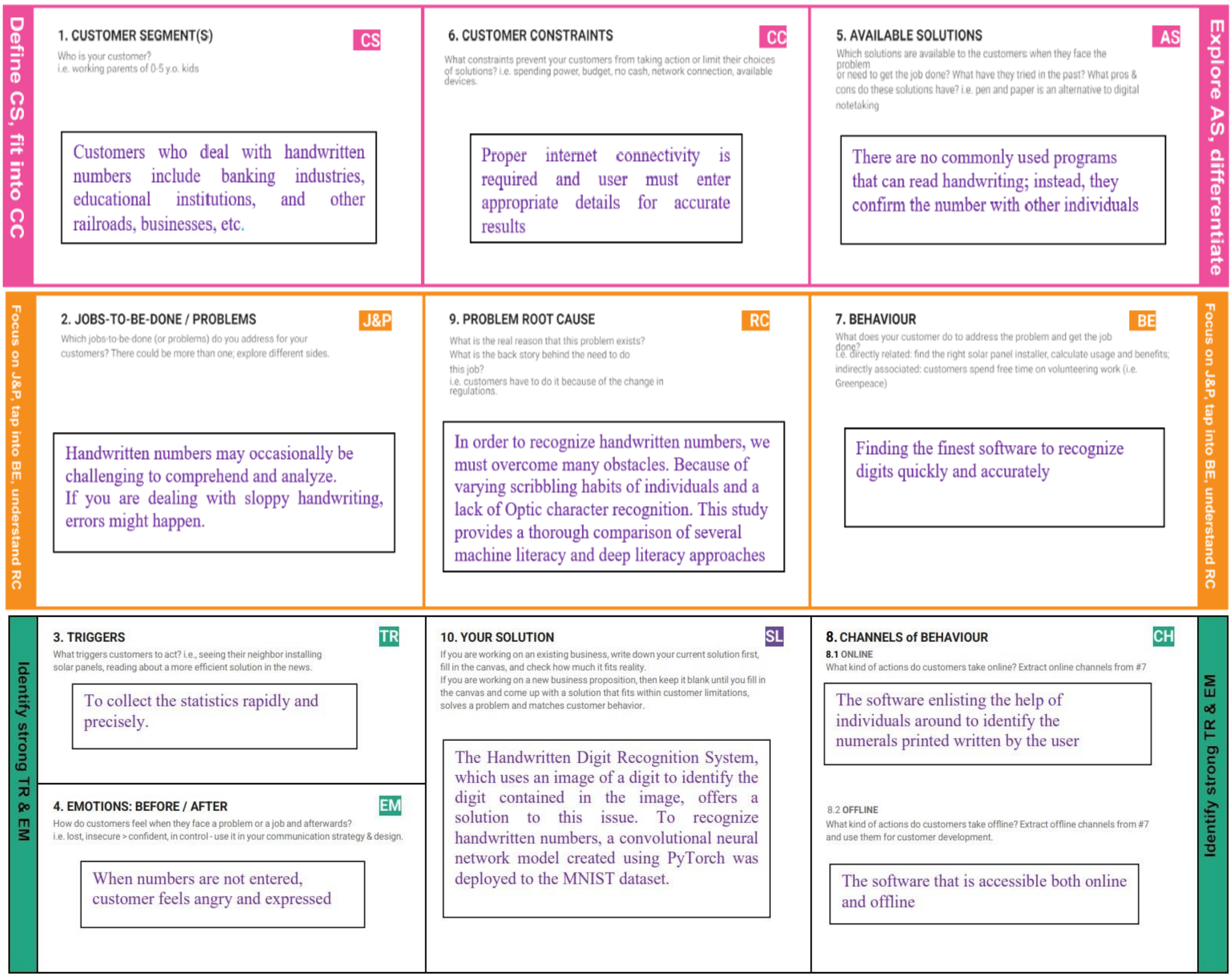
3.3 PROPOSED SOLUTION

S. No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	To propose a novel method to recognize hand written digits. Handwritten digit recognition is the ability of a computer to recognize the human handwritten digits from different sources like images, papers, touch screens, etc, and classify them into 10 predefined classes (0-9)
2.	Idea / Solution description	The capacity of a computer to differentiate human handwriting into 10 specified categories from various sources, such as photos, sheets, touch defences, etc (0-9). We encounter several difficulties in handwritten number identification because various people have different writing styles.
3.	Novelty / Uniqueness	Based on an examination of the thickness and form of the numerical picture, it can accurately and efficiently identify the digits.
4.	Social Impact / Customer Satisfaction	It can be used for the identification of car numbers, reading postal addresses, bank check amounts, and forms, addressing of letters and pattern recognition applications.

`

5.	Business Model (Revenue Model)	The goal of this is to provide efficientand trustworthy methods for online handwriting recognition on computer tablets, recognizing zip codes on mail for postal mail sorting, processing bank check amounts, numeric entries in forms filled up by hand.
6.	Scalability of the Solution	Due to its applicability in several machine learning and computer vision applications, handwritten digit recognition has become a crucial field and is enticing many people. The accuracy is scalable for the model.

3.4 PROBLEM SOLUTION FIT



Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Who is your customer?
i.e. working parents of 0-5 y.o. kids

Customers who deal with handwritten numbers include banking industries, educational institutions, and other railroads, businesses, etc.

6. CUSTOMER CONSTRAINTS CC

What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices.

Proper internet connectivity is required and user must enter appropriate details for accurate results

5. AVAILABLE SOLUTIONS AS

Which solutions are available to the customers when they face the problem
or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper is an alternative to digital notetaking

There are no commonly used programs that can read handwriting; instead, they confirm the number with other individuals

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS J&P

Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides.

Handwritten numbers may occasionally be challenging to comprehend and analyze.
If you are dealing with sloppy handwriting, errors might happen.

9. PROBLEM ROOT CAUSE RC

What is the real reason that this problem exists?
What is the back story behind the need to do this job?
i.e. customers have to do it because of the change in regulations.

In order to recognize handwritten numbers, we must overcome many obstacles. Because of varying scribbling habits of individuals and a lack of Optic character recognition. This study provides a thorough comparison of several machine literacy and deep literacy approaches

7. BEHAVIOUR BE

What does your customer do to address the problem and get the job done?
i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)

Finding the finest software to recognize digits quickly and accurately

Focus on J&P, tap into BE, understand RC

Identify strong TR & EM

3. TRIGGERS TR

What triggers customers to act? i.e., seeing their neighbor installing solar panels, reading about a more efficient solution in the news.

To collect the statistics rapidly and precisely.

4. EMOTIONS: BEFORE / AFTER EM

How do customers feel when they face a problem or a job and afterwards?
i.e. lost, insecure > confident, in control - use it in your communication strategy & design.

When numbers are not entered, customer feels angry and expressed

10. YOUR SOLUTION SL

If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality.
If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behavior.

The Handwritten Digit Recognition System, which uses an image of a digit to identify the digit contained in the image, offers a solution to this issue. To recognize handwritten numbers, a convolutional neural network model created using PyTorch was deployed to the MNIST dataset.

8. CHANNELS of BEHAVIOUR CH

8.1 ONLINE

What kind of actions do customers take online? Extract online channels from #7

The software enlisting the help of individuals around to identify the numerals printed written by the user

8.2 **OFFLINE**

What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.

The software that is accessible both online and offline

Identify strong TR & EM

`

3.5 PROPOSED MODEL

The proposed model contains four stages in order to classify and detect the digits.

- Pre-processing
- Segmentation
- Feature extraction
- Classification and Recognition

PRE-PROCESSING

The role of the pre-processing step is it performs various tasks on the input image. It basically upgrades the image by making it reasonable for segmentation. The fundamental motivation behind pre-processing is to take off a fascinating example from the background. For the most part, noise filtering, smoothing and standardization are to be done in this stage. The pre-processing additionally characterizes a smaller portrayal of the example. Binarization changes over a gray scale image into a binary image. The initial approach to the training set images that are to be processed in order to reduce the data, by thresholding them into a binary image.

SEGMENTATION

Once the pre-processing of the input images is completed, sub-images of individual digits are formed from the sequence of images. Pre-processed digit images are segmented into a sub-image of individual digits, which are assigned a number to each digit. Each individual digit is resized into pixels. In this step an edge detection technique is being used for segmentation of dataset images.

FEATURE EXTRACTION

After the completion of pre-processing stage and segmentation stage, the pre-processed images are represented in the form of a matrix which contains pixels of the images that are of very large size. In this way it will be valuable to represent the digits in the images which contain the necessary information.This activity is called feature extraction. In the feature extraction stage redundancy from the data is removed.

`

CLASSIFICATION AND RECOGNITION

In the classification and recognition step the extracted feature vectors are taken as an individual input to each of the following classifiers. In order to showcase the working system model extracted features are combined and defined using the classifiers.

`

CHAPTER 4
REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

FR No.	Functional Requirement	Sub Requirement (Story / Sub-Task)
1	Input image	Input images must be given by the user. The inputs are handwritten digits from 0 to 9. The model classifies them and convert them into digitalized form
2	Algorithm	Convolution Neural Network using keras is being used to implement handwritten digit recognition
3	Website	We are going to build a GUI in which you can draw the digit and recognize it straight away
4	Dataset	The MNIST dataset is an acronym that stands for the Modified National Institute of Standards and Technology dataset. It is a dataset of 60,000 small square 28×28 pixel grayscale images of handwritten single digits between 0 and 9
5	Cloud	The cloud allows employees to access files on any device. Cloud services are a good option for anyone looking to train and deploy memory-intensive, complex Machine Learning/Deep Learning models. Cloud services provides cost-effective solution

4.2 NON-FUNCTIONAL REQUIREMENTS

NFR No.	Non-Functional Requirement	Description
1	Usability	It is user-friendly and can be used by users of all skill levels. It is used in the identification of car numbers, reading of checks at banks and post offices, and the addressing of letters

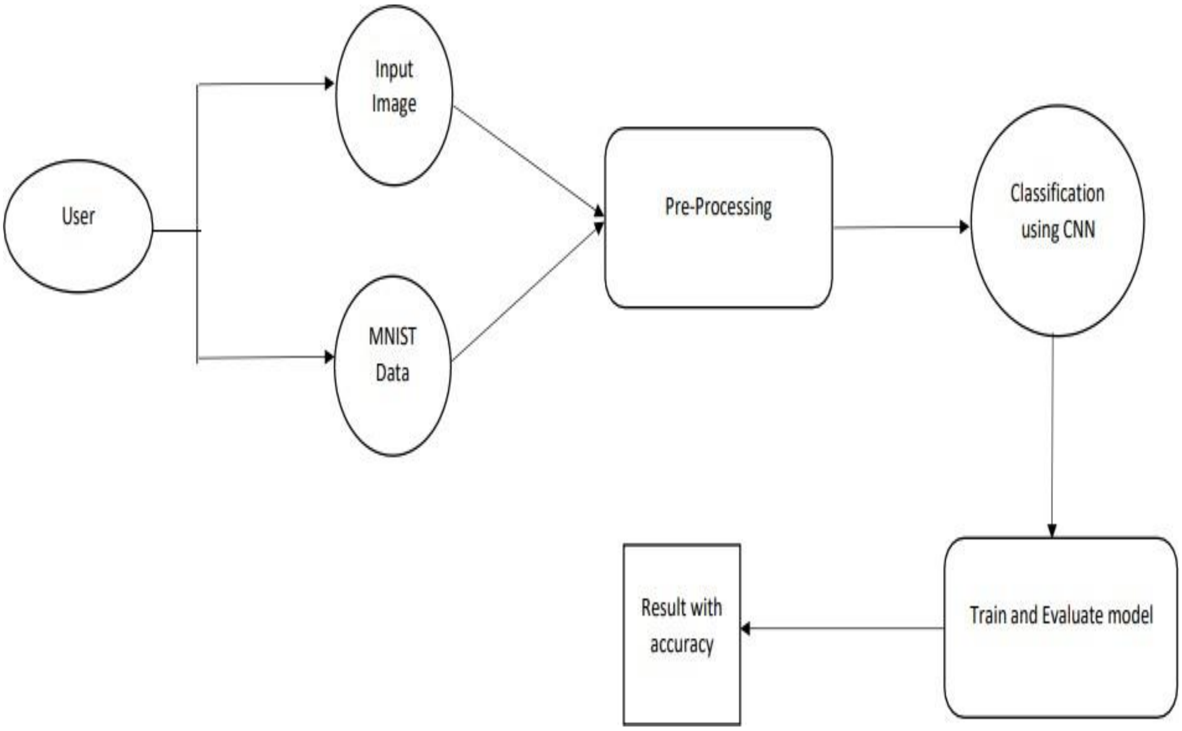
`

2	Security	Authentication can be used to verify the user
3	Accuracy	Diversity in human writing types, spacing differences, and irregularities of handwriting causes variation in accuracy
4	Performance	Performance metric is accuracy. Instantaneous recognition of the handwritten digits is also a metric
5	Availability	The website will be made public for everyone touse.
6	Scalability	In future, accuracy can be enhanced further and all the handwritten digits by the users can be stored and updated automatically in the dataset

`

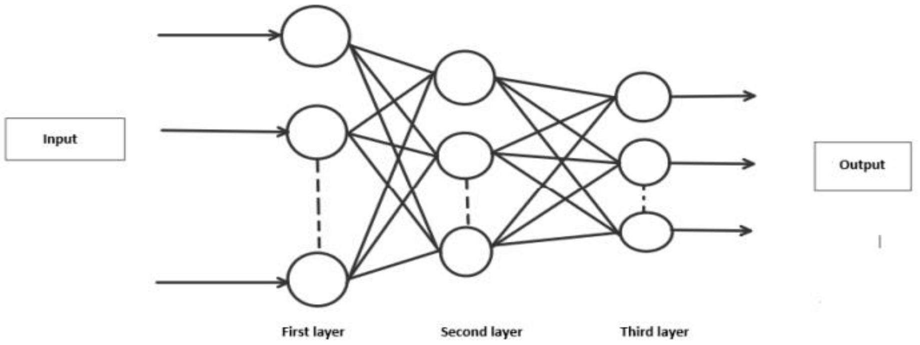
CHAPTER 5
PROJECT DESIGN

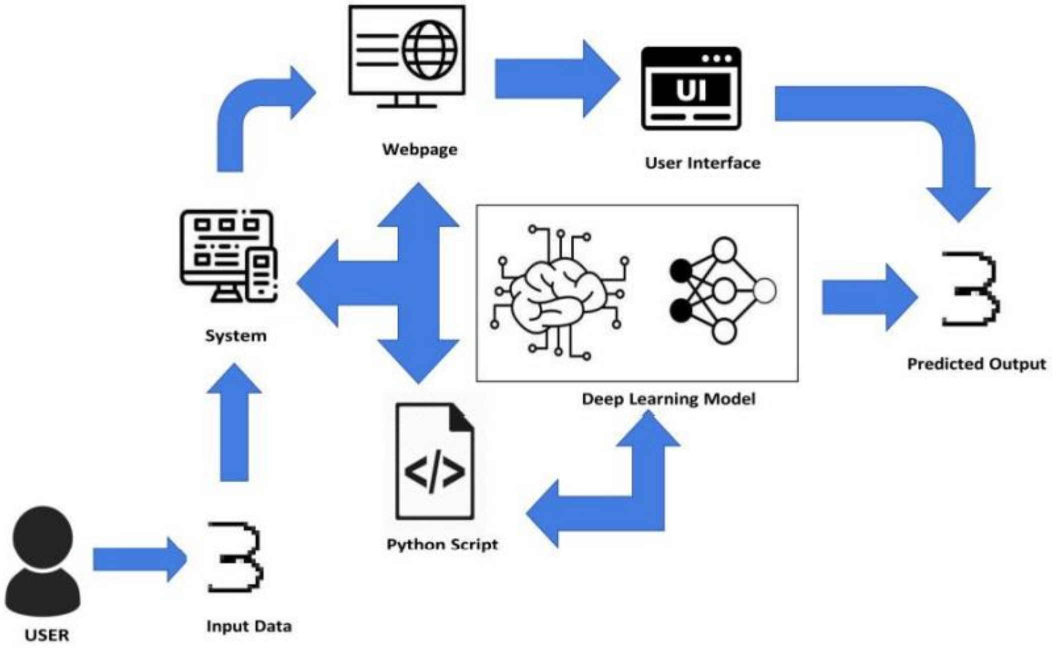
5.1 DATA FLOW DIAGRAMS



`

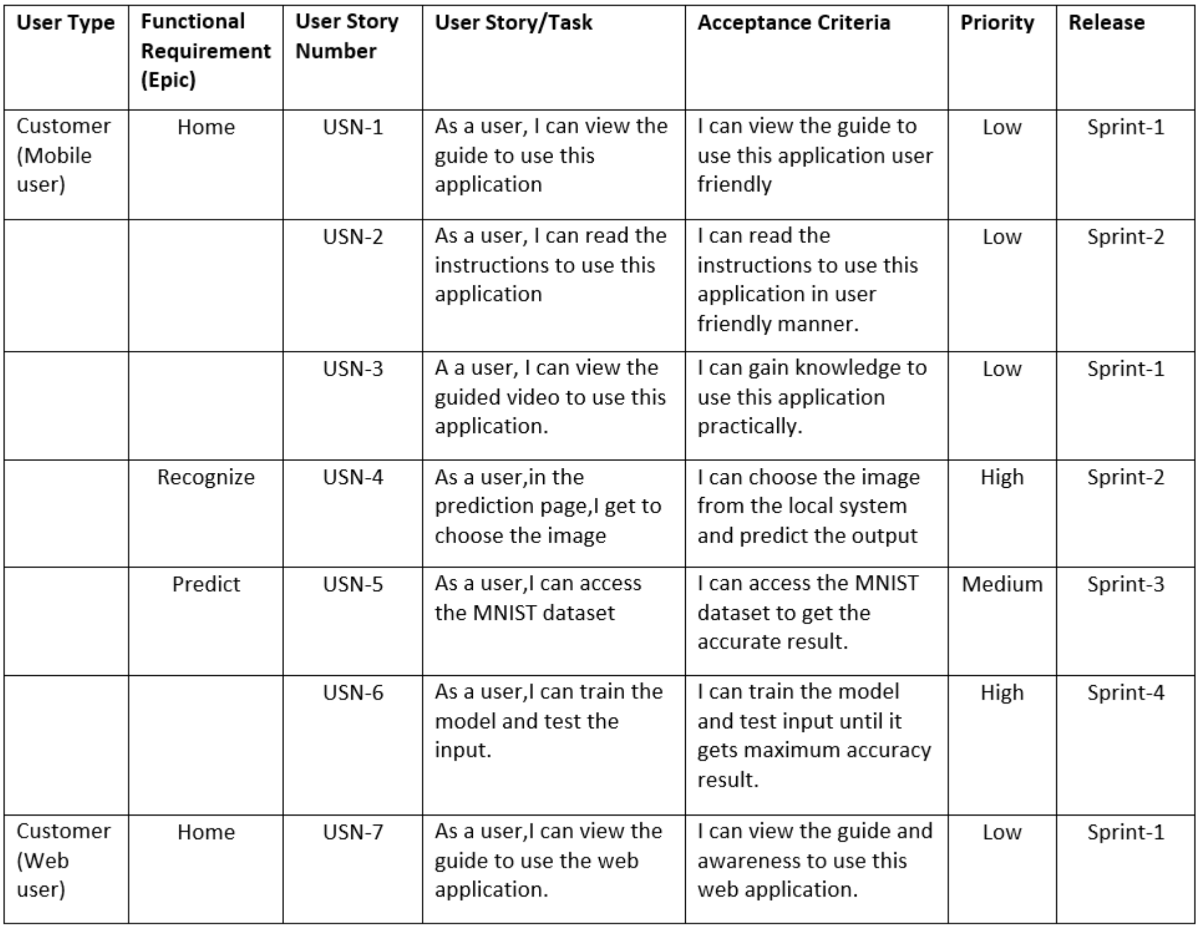
5.2 SOLUTION AND TECHNICAL ARCHITECTURE



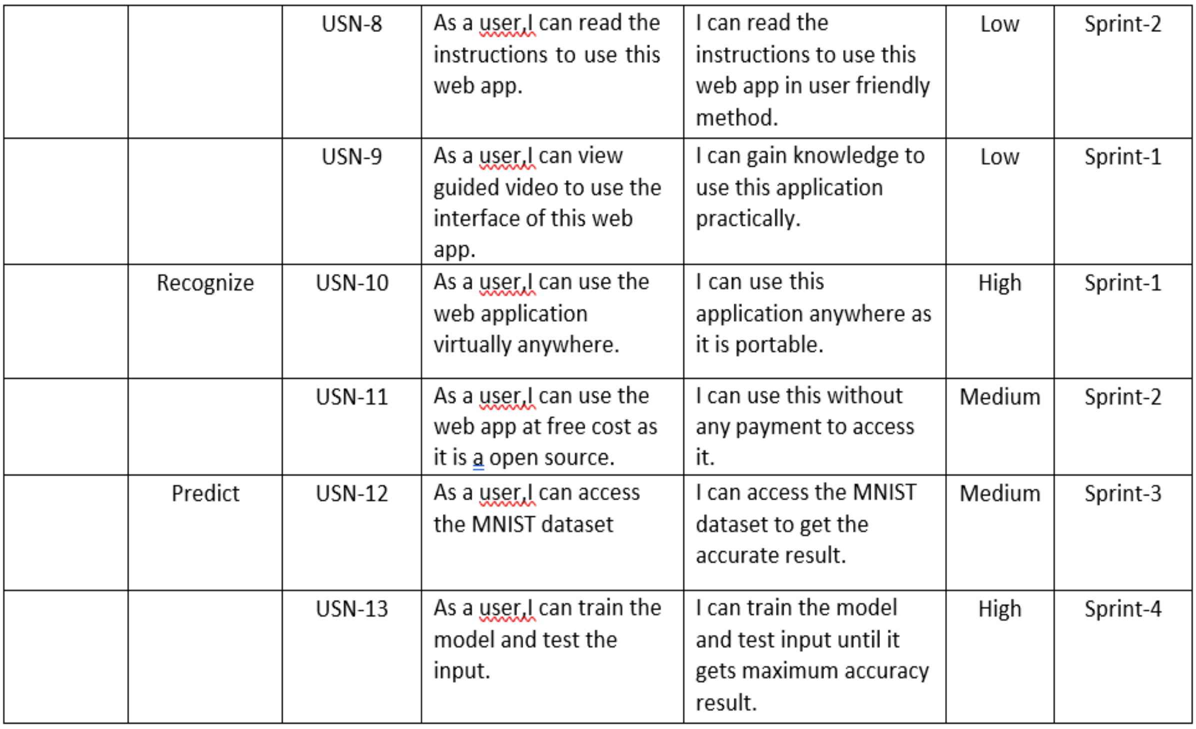


`

5.3 USER STORIES



User Type	Functional Requirement (Epic)	User Story Number	User Story/Task	Acceptance Criteria	Priority	Release
Customer (Mobile user)	Home	USN-1	As a user, I can view the guide to use this application	I can view the guide to use this application user friendly	Low	Sprint-1
		USN-2	As a user, I can read the instructions to use this application	I can read the instructions to use this application in user friendly manner.	Low	Sprint-2
		USN-3	A a user, I can view the guided video to use this application.	I can gain knowledge to use this application practically.	Low	Sprint-1
	Recognize	USN-4	As a user,in the prediction page,I get to choose the image	I can choose the image from the local system and predict the output	High	Sprint-2
	Predict	USN-5	As a user,I can access the MNIST dataset	I can access the MNIST dataset to get the accurate result.	Medium	Sprint-3
		USN-6	As a user,I can train the model and test the input.	I can train the model and test input until it gets maximum accuracy result.	High	Sprint-4
Customer (Web user)	Home	USN-7	As a user,I can view the guide to use the web application.	I can view the guide and awareness to use this web application.	Low	Sprint-1



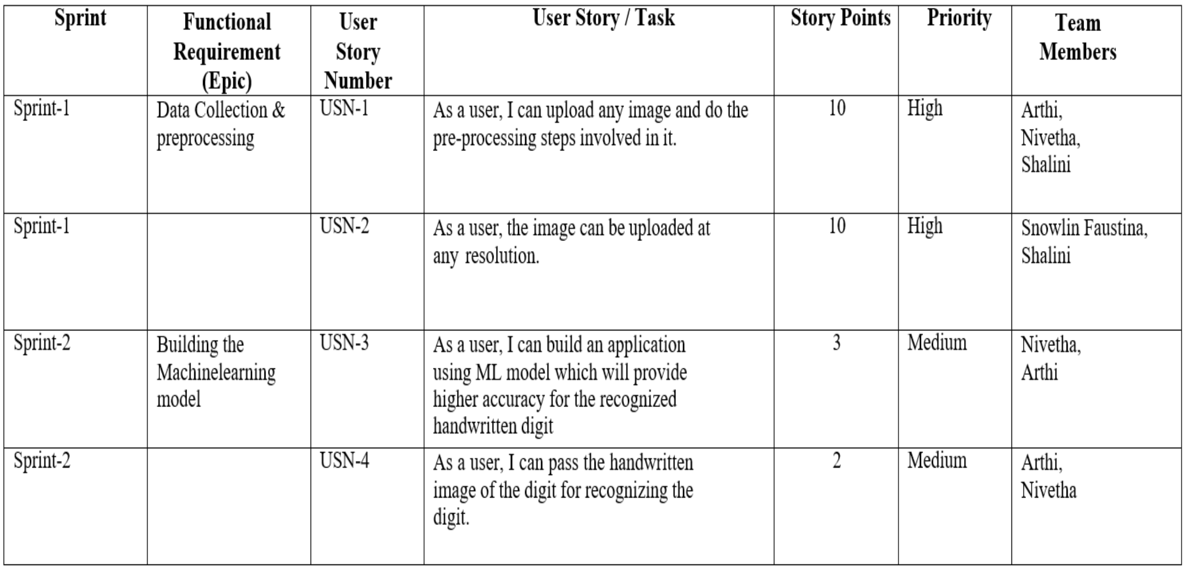
		USN-8	As a user,I can read the instructions to use this web app.	I can read the instructions to use this web app in user friendly method.	Low	Sprint-2
		USN-9	As a user,I can view guided video to use the interface of this web app.	I can gain knowledge to use this application practically.	Low	Sprint-1
	Recognize	USN-10	As a user,I can use the web application virtually anywhere.	I can use this application anywhere as it is portable.	High	Sprint-1
		USN-11	As a user,I can use the web app at free cost as it is a open source.	I can use this without any payment to access it.	Medium	Sprint-2
	Predict	USN-12	As a user,I can access the MNIST dataset	I can access the MNIST dataset to get the accurate result.	Medium	Sprint-3
		USN-13	As a user,I can train the model and test the input.	I can train the model and test input until it gets maximum accuracy result.	High	Sprint-4

`

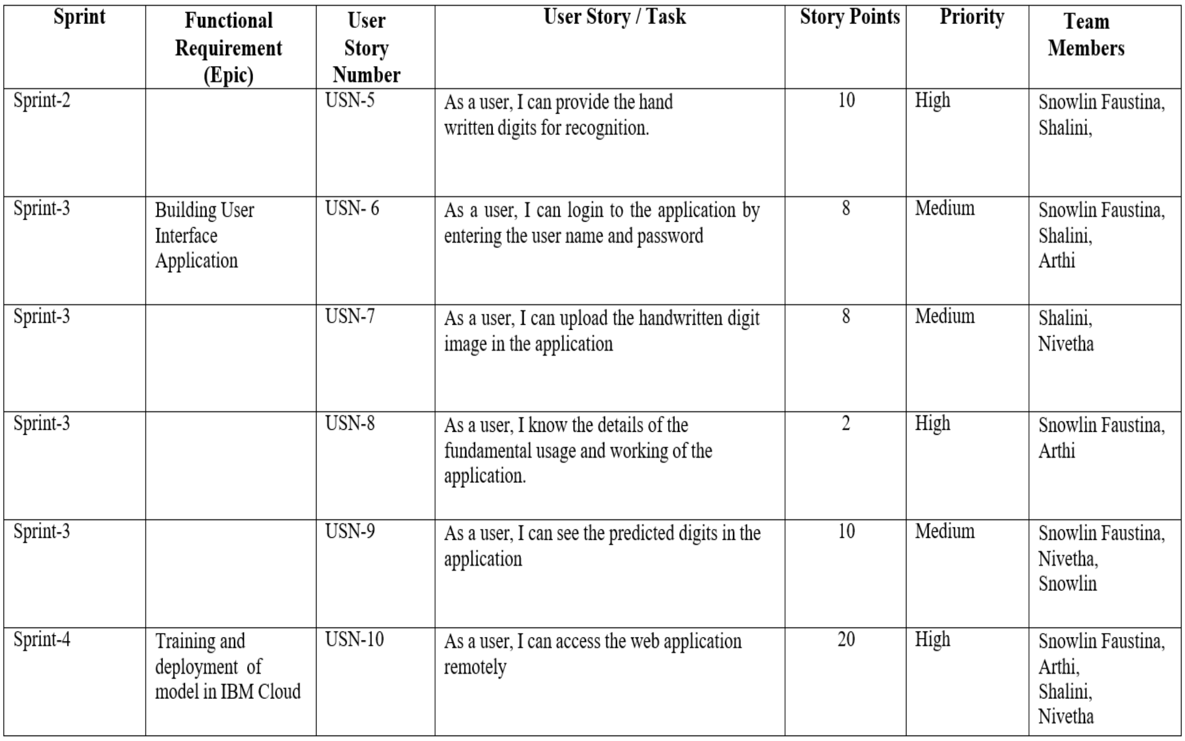
CHAPTER 6

PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNING AND ESTIMATION



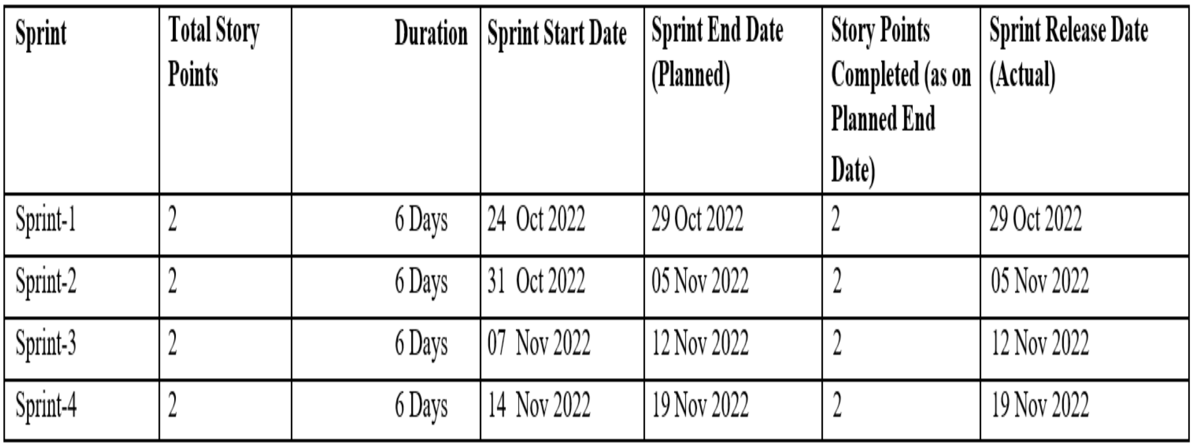
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data Collection & preprocessing	USN-1	As a user, I can upload any image and do the pre-processing steps involved in it.	10	High	Arthi, Nivetha, Shalini
Sprint-1		USN-2	As a user, the image can be uploaded at any resolution.	10	High	Snowlin Faustina, Shalini
Sprint-2	Building the Machinelearning model	USN-3	As a user, I can build an application using ML model which will provide higher accuracy for the recognized handwritten digit	3	Medium	Nivetha, Arthi
Sprint-2		USN-4	As a user, I can pass the handwritten image of the digit for recognizing the digit.	2	Medium	Arthi, Nivetha



Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-2		USN-5	As a user, I can provide the hand written digits for recognition.	10	High	Snowlin Faustina, Shalini,
Sprint-3	Building User Interface Application	USN- 6	As a user, I can login to the application by entering the user name and password	8	Medium	Snowlin Faustina, Shalini, Arthi
Sprint-3		USN-7	As a user, I can upload the handwritten digit image in the application	8	Medium	Shalini, Nivetha
Sprint-3		USN-8	As a user, I know the details of the fundamental usage and working of the application.	2	High	Snowlin Faustina, Arthi
Sprint-3		USN-9	As a user, I can see the predicted digits in the application	10	Medium	Snowlin Faustina, Nivetha, Snowlin
Sprint-4	Training and deployment of model in IBM Cloud	USN-10	As a user, I can access the web application remotely	20	High	Snowlin Faustina, Arthi, Shalini, Nivetha

`

6.2 SPRINT DELIVERY SCHEDULE

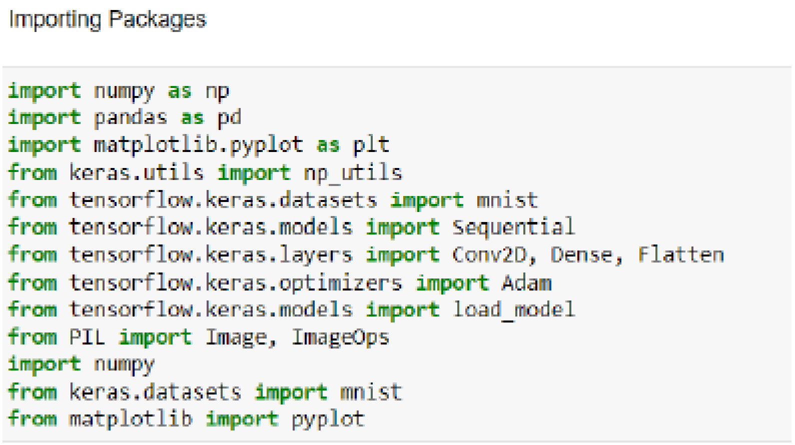


Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	2	6 Days	24 Oct 2022	29 Oct 2022	2	29 Oct 2022
Sprint-2	2	6 Days	31 Oct 2022	05 Nov 2022	2	05 Nov 2022
Sprint-3	2	6 Days	07 Nov 2022	12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

`

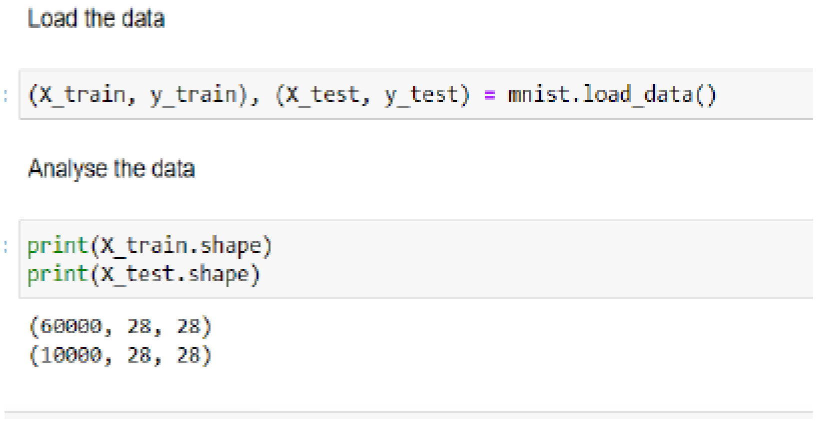
CHAPTER 7

CODING & SOLUTIONING



Importing Packages

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from keras.utils import np_utils
from tensorflow.keras.datasets import mnist
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, Dense, Flatten
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.models import load_model
from PIL import Image, ImageOps
import numpy
from keras.datasets import mnist
from matplotlib import pyplot
```



Load the data

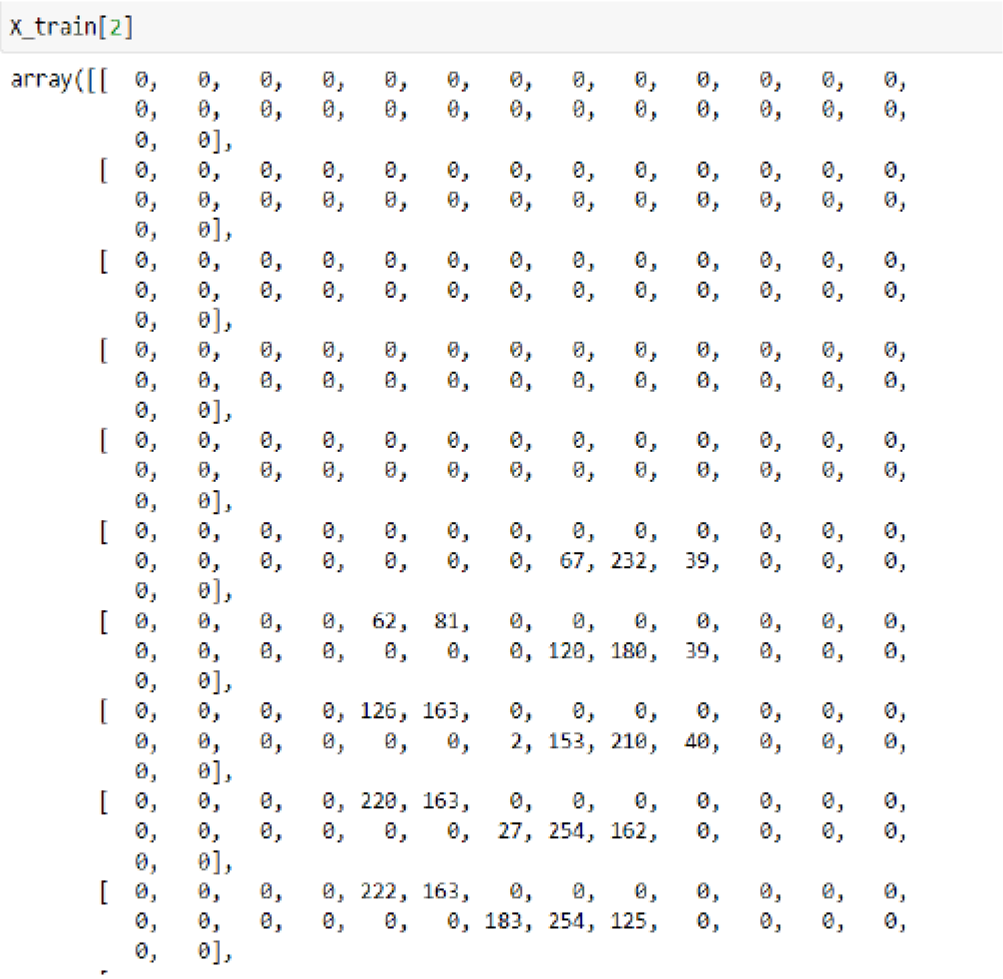
```
(X_train, y_train), (X_test, y_test) = mnist.load_data()
```

Analyse the data

```
print(X_train.shape)
print(X_test.shape)
```

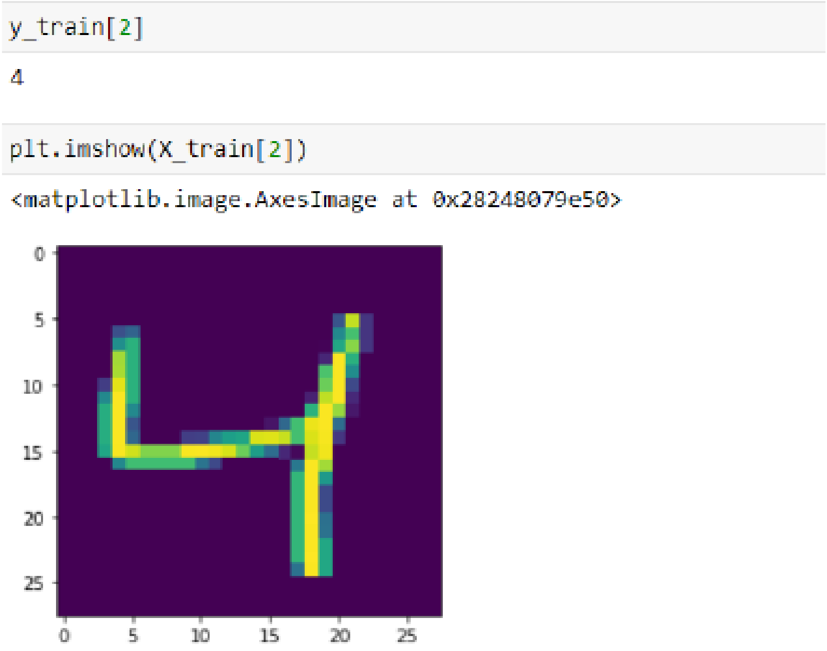
```
(60000, 28, 28)
(10000, 28, 28)
```

`



```
X_train[2]
```

```
array([[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
        [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
        [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
        [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
        [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
        [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,  67, 232,  39,   0,   0,   0,
          0,   0],
        [  0,   0,   0,   0,  62,  81,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0, 120, 180,  39,   0,   0,   0,
          0,   0],
        [  0,   0,   0,   0, 126, 163,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   2, 153, 210,  40,   0,   0,   0,
          0,   0],
        [  0,   0,   0,   0, 220, 163,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,  27, 254, 162,   0,   0,   0,   0,
          0,   0],
        [  0,   0,   0,   0, 222, 163,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0, 183, 254, 125,   0,   0,   0,   0,
          0,   0],
```



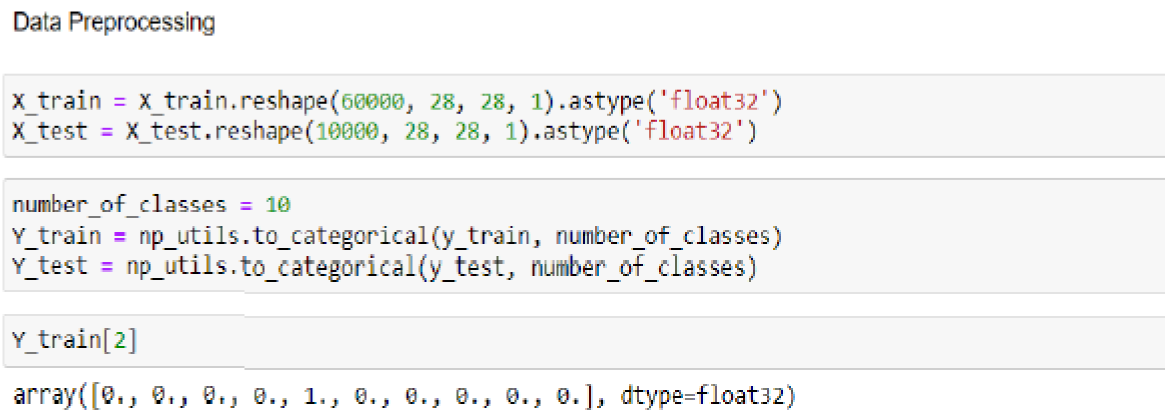
```
y_train[2]
```

```
4
```

```
plt.imshow(X_train[2])
```

```
<matplotlib.image.AxesImage at 0x28248079e50>
```

`



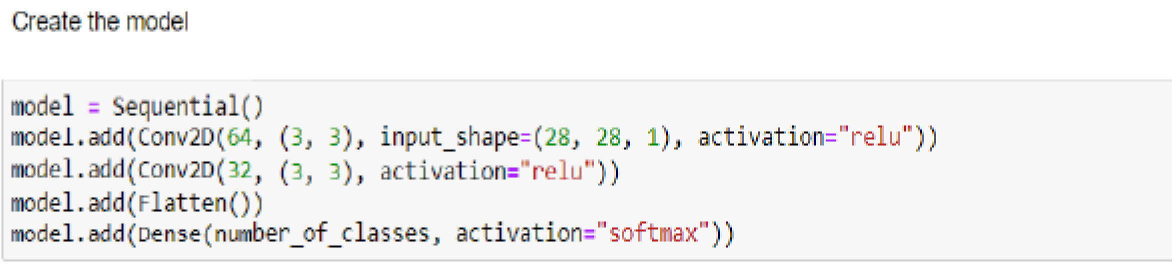
Data Preprocessing

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

```
number_of_classes = 10
Y_train = np_utils.to_categorical(y_train, number_of_classes)
Y_test = np_utils.to_categorical(y_test, number_of_classes)
```

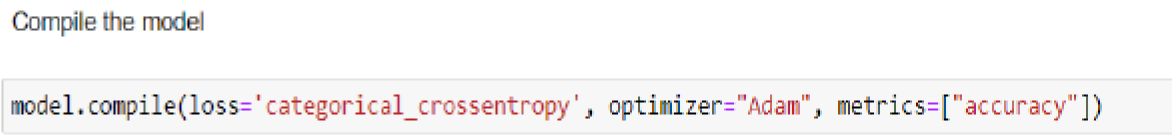
```
Y_train[2]
```

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



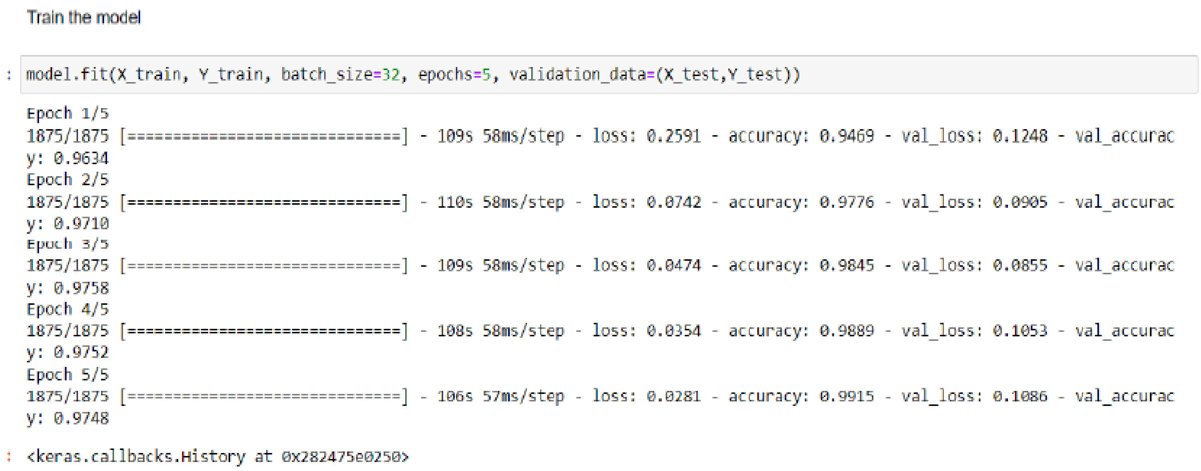
Create the model

```
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"))
```



Compile the model

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

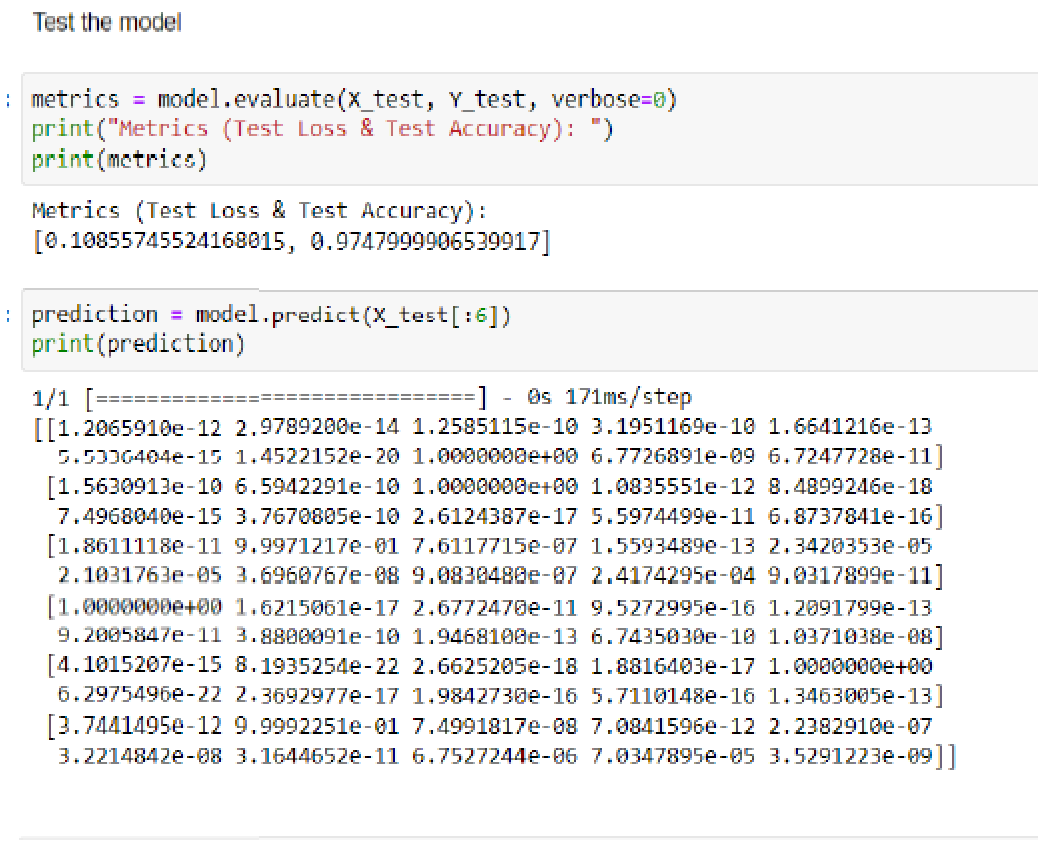


Train the model

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

```
Epoch 1/5
1875/1875 [==============================] - 109s 58ms/step - loss: 0.2591 - accuracy: 0.9469 - val_loss: 0.1248 - val_accurac
y: 0.9634
Epoch 2/5
1875/1875 [==============================] - 110s 58ms/step - loss: 0.0742 - accuracy: 0.9776 - val_loss: 0.0905 - val_accurac
y: 0.9710
Epoch 3/5
1875/1875 [==============================] - 109s 58ms/step - loss: 0.0474 - accuracy: 0.9845 - val_loss: 0.0855 - val_accurac
y: 0.9758
Epoch 4/5
1875/1875 [==============================] - 108s 58ms/step - loss: 0.0354 - accuracy: 0.9889 - val_loss: 0.1053 - val_accurac
y: 0.9752
Epoch 5/5
1875/1875 [==============================] - 106s 57ms/step - loss: 0.0281 - accuracy: 0.9915 - val_loss: 0.1086 - val_accurac
y: 0.9748
```

```
: <keras.callbacks.History at 0x282475e0250>
```



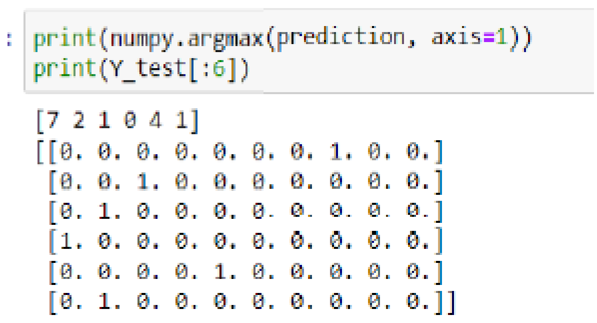
Test the model

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

```
Metrics (Test Loss & Test Accuracy):
[0.10855745524168015, 0.9747999906539917]
```

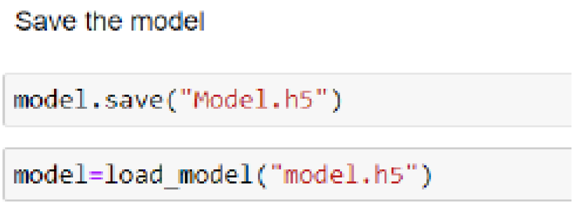
```
prediction = model.predict(X_test[:6])
print(prediction)
```

```
1/1 [==============================] - 0s 171ms/step
[[1.2065910e-12 2.9789200e-14 1.2585115e-10 3.1951169e-10 1.6641216e-13
  5.5336404e-15 1.4522152e-20 1.0000000e+00 6.7726891e-09 6.7247728e-11]
 [1.5630913e-10 6.5942291e-10 1.0000000e+00 1.0835551e-12 8.4899246e-18
  7.4968040e-15 3.7670805e-10 2.6124387e-17 5.5974499e-11 6.8737841e-16]
 [1.8611118e-11 9.9971217e-01 7.6117715e-07 1.5593489e-13 2.3420353e-05
  2.1031763e-05 3.6960767e-08 9.0830480e-07 2.4174295e-04 9.0317899e-11]
 [1.0000000e+00 1.6215061e-17 2.6772470e-11 9.5272995e-16 1.2091799e-13
  9.2005847e-11 3.8800091e-10 1.9468100e-13 6.7435030e-10 1.0371038e-08]
 [4.1015207e-15 8.1935254e-22 2.6625205e-18 1.8816403e-17 1.0000000e+00
  6.2975496e-22 2.3692977e-17 1.9842730e-16 5.7110148e-16 1.3463005e-13]
 [3.7441495e-12 9.9992251e-01 7.4991817e-08 7.0841596e-12 2.2382910e-07
  3.2214842e-08 3.1644652e-11 6.7527244e-06 7.0347895e-05 3.5291223e-09]]
```



```
print(numpy.argmax(prediction, axis=1))
print(Y_test[:6])
```

```
[7 2 1 0 4 1]
[[0. 0. 0. 0. 0. 0. 0. 1. 0. 0.]
 [0. 0. 1. 0. 0. 0. 0. 0. 0. 0.]
 [0. 1. 0. 0. 0. 0. 0. 0. 0. 0.]
 [1. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 1. 0. 0. 0. 0. 0.]
 [0. 1. 0. 0. 0. 0. 0. 0. 0. 0.]]
```



Save the model

```
model.save("Model.h5")
```

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

`

7.1 DATABASE SCHEMA

The MNIST database (Modified National Institute of Standards and Technology database) is a large database of handwritten digits. The database is also widely used for training and testing in the field of machine learning. It was created by "re-mixing" the samples from NIST's original datasets. The creators felt that since NIST's training dataset was taken from American Census Bureau employees, while the testing dataset was taken from American high school students, it was not well-suited for machine learning experiments. Furthermore, the black and white images from NIST were normalized to fit into a 28x28 pixel bounding box and anti-aliased, which introduced gray scale levels.

`

CHAPTER 8

TESTING

8.1 TEST CASES

Test caseID	Feature Type	Component	Test Scenario	Expected Result	Actual Result	Status
HP_TC_001	UI	Home Page	Verify UI elements inthe Home Page	The home page must be displayed properly	Working as expected	PASSED
BE_TC_001	UI	Back End	Check if all the routes are working properly	All the routes must work properly	Working as expected	PASSED
RP_TC_001	Functional	Result Page	Proper display of result page	The result page must be displayed properly	Working as expected	PASSED

`

8.2 USER ACCEPTANCE TESTING

8.2.1 DEFECT ANALYSIS

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Total
By Design	1	0	1	0	2
Duplicate	0	0	0	0	0
External	0	0	2	0	2
Fixed	4	1	0	1	6
Not Reproduced	0	0	0	1	1
Skipped	0	0	0	1	1
Total	5	1	3	3	12

`

8.2.2 TEST CASE ANALYSIS

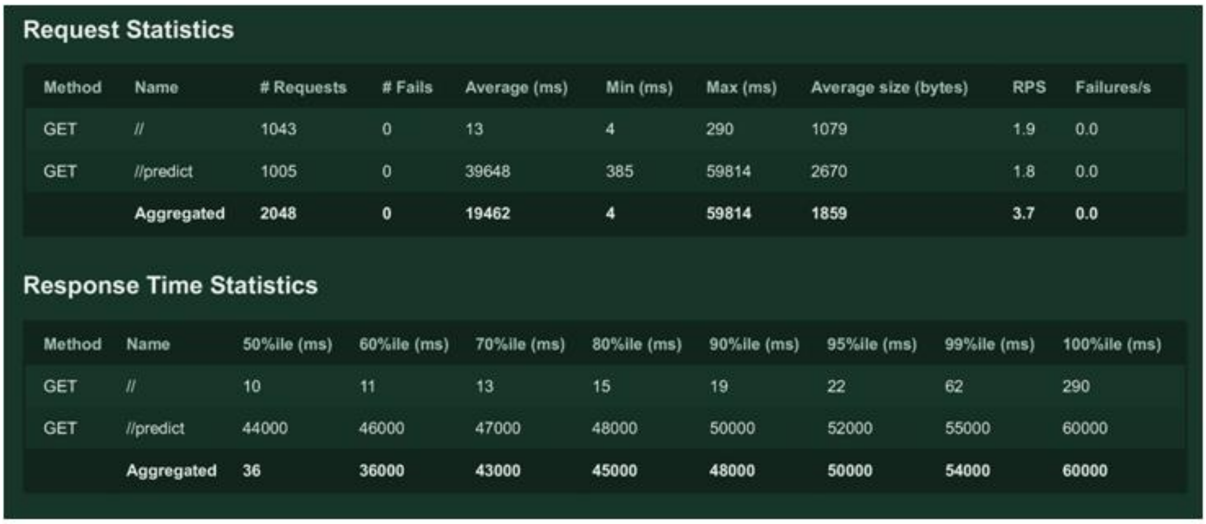
Section	Total Cases	Not Tested	Fail	Pass
Client Application	10	0	3	7
Security	2	0	1	1
Performance	3	0	1	2
Exception Reporting	2	0	0	2

`

CHAPTER 9

RESULTS

9.1 PERFORMANCE METRICS



Request Statistics

Method	Name	# Requests	# Fails	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	RPS	Failures/s
GET	//	1043	0	13	4	290	1079	1.9	0.0
GET	//predict	1005	0	39648	385	59814	2670	1.8	0.0
	Aggregated	**2048**	**0**	**19462**	**4**	**59814**	**1859**	**3.7**	**0.0**

Response Time Statistics

Method	Name	50%ile (ms)	60%ile (ms)	70%ile (ms)	80%ile (ms)	90%ile (ms)	95%ile (ms)	99%ile (ms)	100%ile (ms)
GET	//	10	11	13	15	19	22	62	290
GET	//predict	44000	46000	47000	48000	50000	52000	55000	60000
	Aggregated	**36**	**36000**	**43000**	**45000**	**48000**	**50000**	**54000**	**60000**

`

CHAPTER 10

ADVANTAGED AND DISADVANTAGES

ADVANTAGES

- Digit Recognition plays an important role in the modern world.
- It can solve more complex problems and makes the human job easier. This type of system can be widely used in the world to recognize zip code or postal code for mail sorting.
- In banking sector too where more handwritten numbers are involved like account number, figure of cash and checks.
- The system not only produces a classification of the digit but also a rich description of the instantiation parameters which can yield information such as writing styles.

DISADVANTAGES

- This is not done in real time as a person writes therefore not appropriate for immediate text input.

SOCIAL IMPACT

- Postal department and courier services can easily find the digits written.
- Old people who will have eye sight issues with handwritten digits.

BUSINESS IMPACT

Baking sector and Postal sector by providing the services.

`

APPLICATIONS

- Detection of vehicle numbers
- Banks for reading cheques
- Post offices for arranging letters.

`

CHAPTER 11

CONCLUSION

This project demonstrated a web application that is used to recognise handwritten numbers. The model predicts the handwritten digit when the user gives the input. 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 more room for improvement, which can be implemented in subsequent versions.

`

CHAPTER 12

FUTURE SCOPE

FUTURE SCOPE

The task of handwritten digit recognition, using a classifier, has great importance and use such as—online handwritten recognition on computer tablets, recognize zip codes on mail for postal mail sorting, processing bank check accounts, numeric entries in forms filled up by hand(for example-tax forms) and so on.

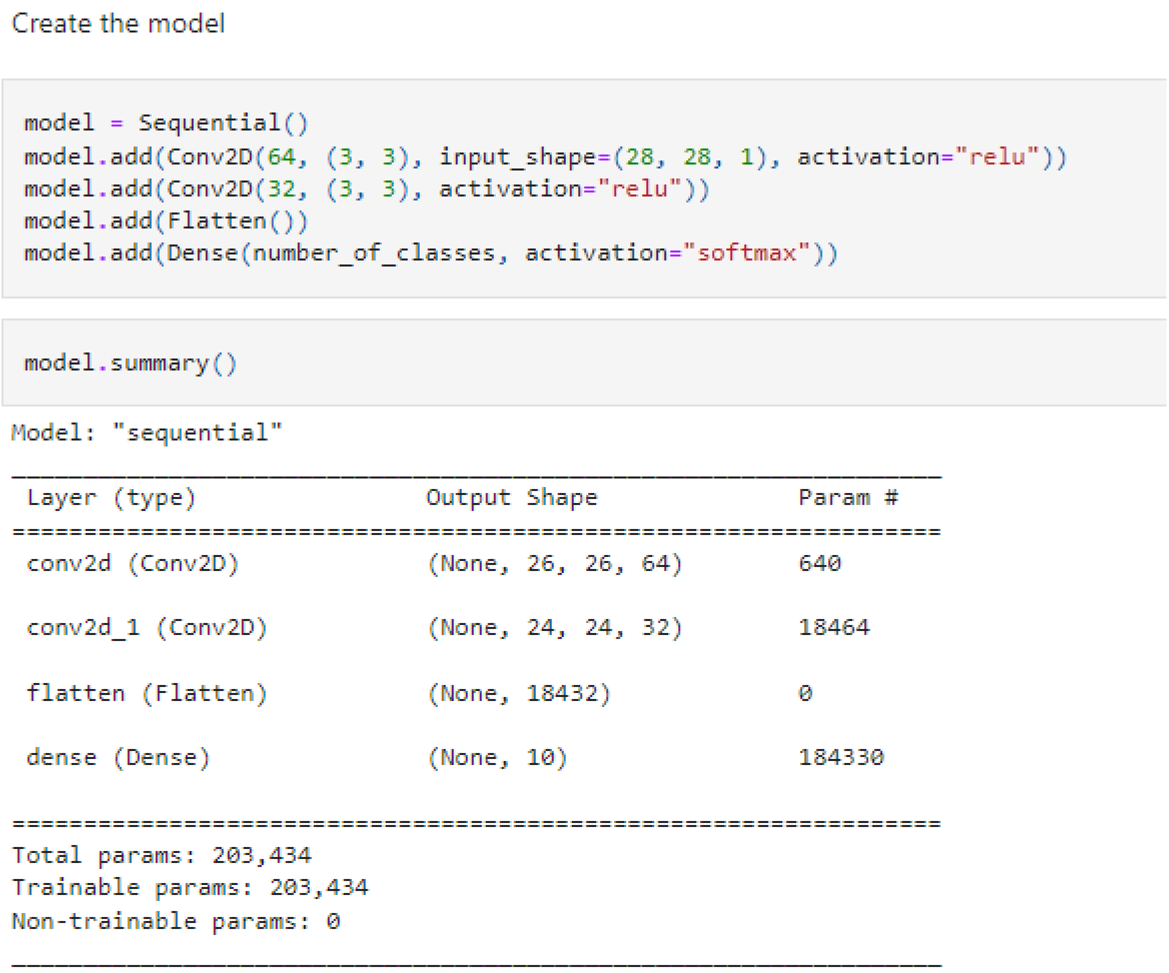
`

CHAPTER 13

APPENDIX

SOURCE CODE

MODEL CREATION

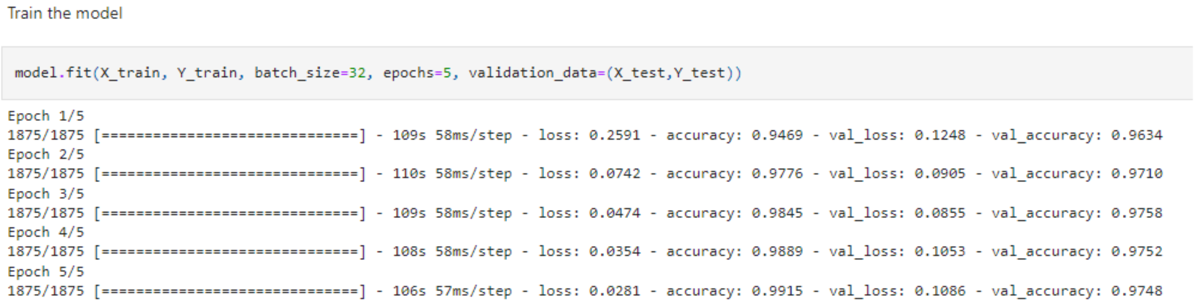


Create the model

```
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.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
_________________________________________________________________
```



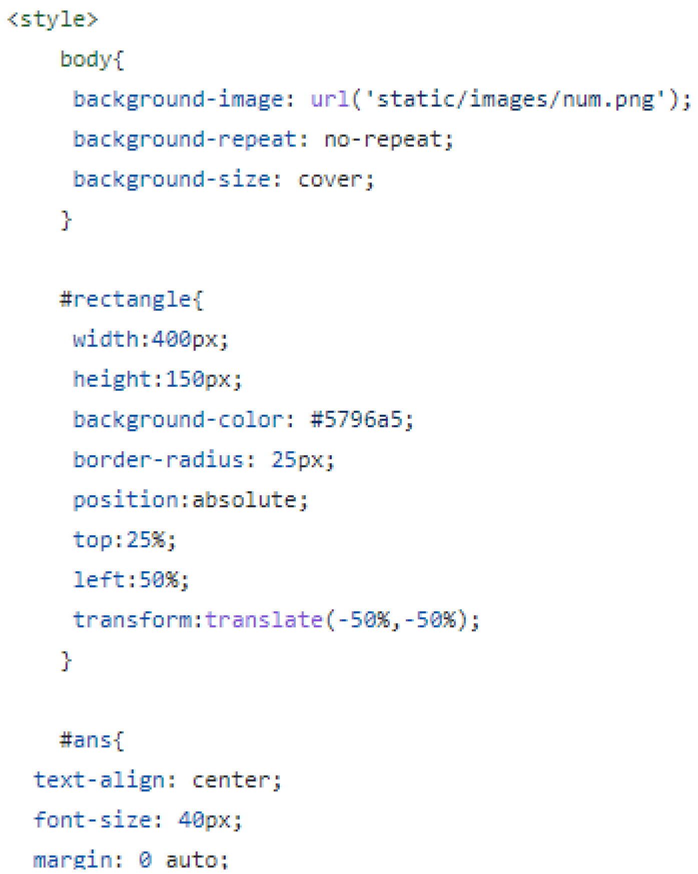
Train the model

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

```
Epoch 1/5
1875/1875 [==============================] - 109s 58ms/step - loss: 0.2591 - accuracy: 0.9469 - val_loss: 0.1248 - val_accuracy: 0.9634
Epoch 2/5
1875/1875 [==============================] - 110s 58ms/step - loss: 0.0742 - accuracy: 0.9776 - val_loss: 0.0905 - val_accuracy: 0.9710
Epoch 3/5
1875/1875 [==============================] - 109s 58ms/step - loss: 0.0474 - accuracy: 0.9845 - val_loss: 0.0855 - val_accuracy: 0.9758
Epoch 4/5
1875/1875 [==============================] - 108s 58ms/step - loss: 0.0354 - accuracy: 0.9889 - val_loss: 0.1053 - val_accuracy: 0.9752
Epoch 5/5
1875/1875 [==============================] - 106s 57ms/step - loss: 0.0281 - accuracy: 0.9915 - val_loss: 0.1086 - val_accuracy: 0.9748
```

`

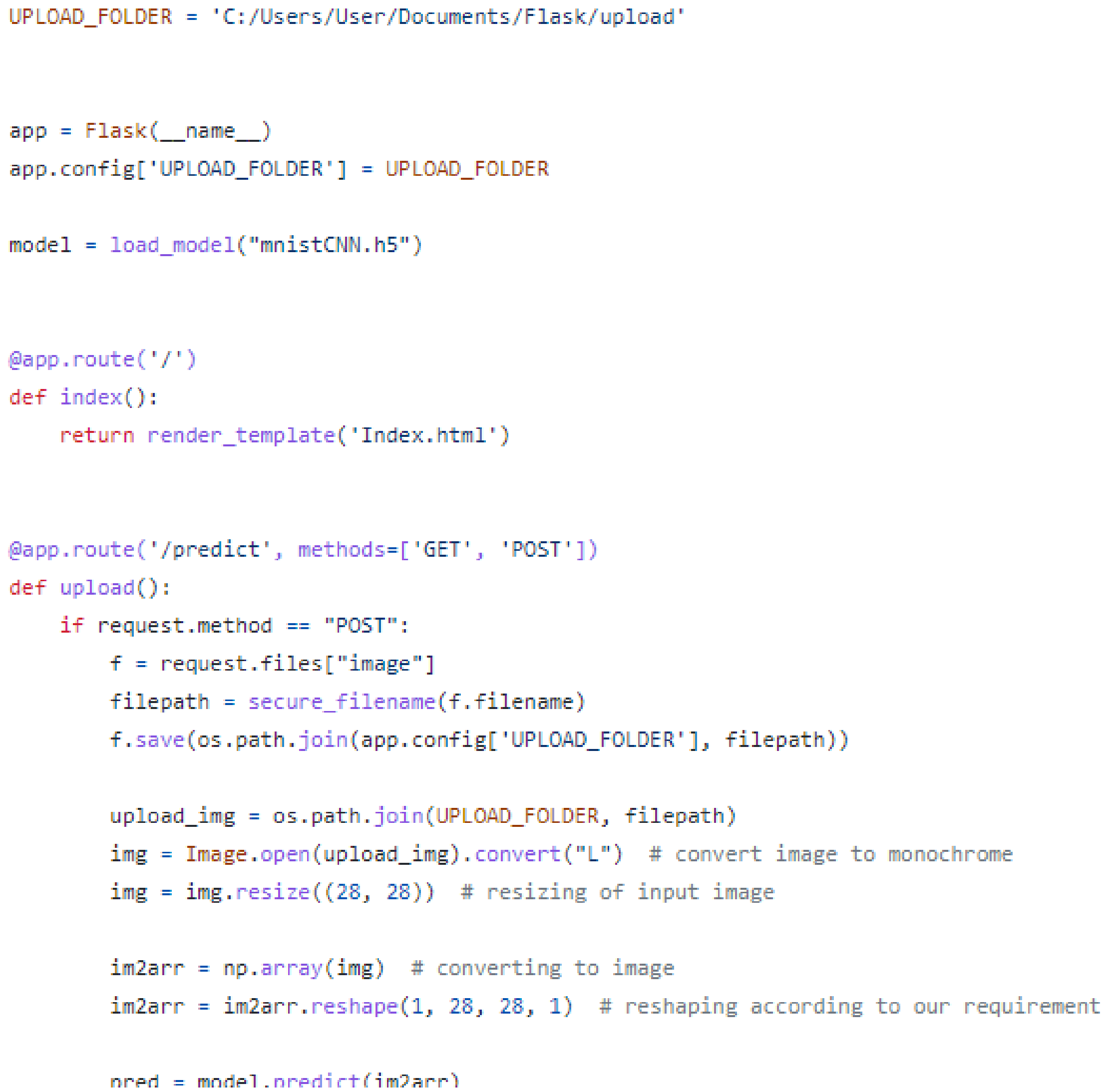
HOME PAGE (HTML)



```
<style>
    body{
     background-image: url('static/images/num.png');
     background-repeat: no-repeat;
     background-size: cover;
    }

    #rectangle{
     width:400px;
     height:150px;
     background-color: #5796a5;
     border-radius: 25px;
     position:absolute;
     top:25%;
     left:50%;
     transform:translate(-50%,-50%);
    }

    #ans{
  text-align: center;
  font-size: 40px;
  margin: 0 auto;
```

`

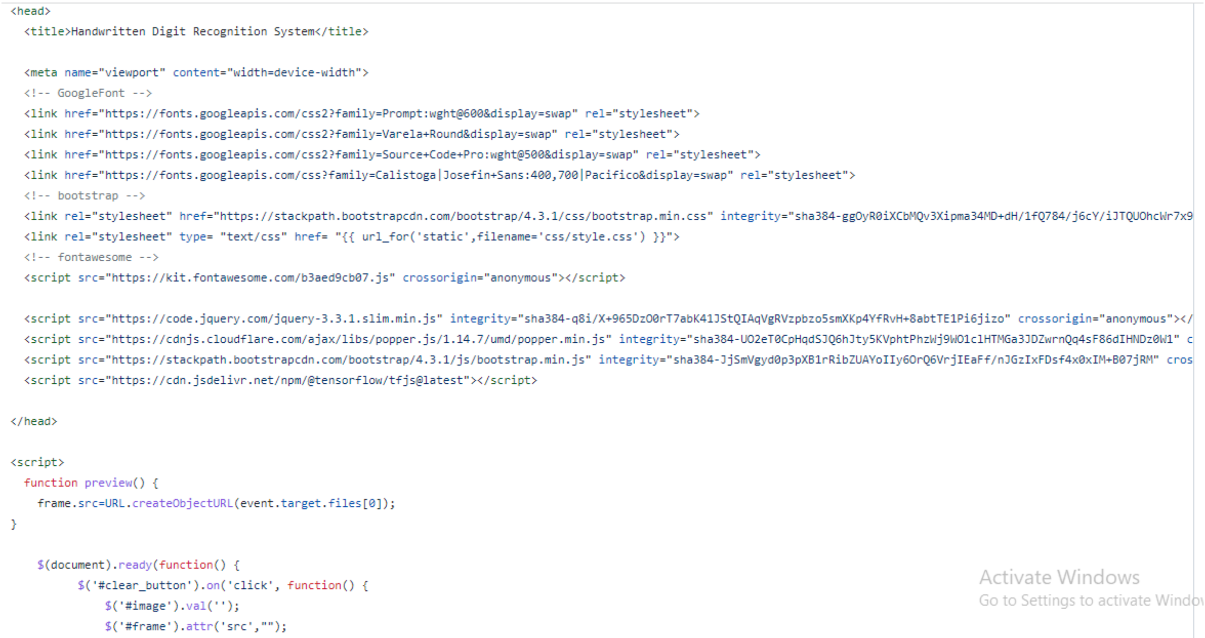
app.py



```
UPLOAD_FOLDER = 'C:/Users/User/Documents/Flask/upload'


app = Flask(__name__)
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER

model = load_model("mnistCNN.h5")


@app.route('/')
def index():
    return render_template('Index.html')


@app.route('/predict', methods=['GET', 'POST'])
def upload():
    if request.method == "POST":
        f = request.files["image"]
        filepath = secure_filename(f.filename)
        f.save(os.path.join(app.config['UPLOAD_FOLDER'], filepath))

        upload_img = os.path.join(UPLOAD_FOLDER, filepath)
        img = Image.open(upload_img).convert("L")  # convert image to monochrome
        img = img.resize((28, 28))  # resizing of input image

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

        pred = model.predict(im2arr)
```

`

Index.html



```
<head>
  <title>Handwritten Digit Recognition System</title>

  <meta name="viewport" content="width=device-width">
  <!-- GoogleFont -->
  <link href="https://fonts.googleapis.com/css2?family=Prompt:wght@600&display=swap" rel="stylesheet">
  <link href="https://fonts.googleapis.com/css2?family=Varela+Round&display=swap" rel="stylesheet">
  <link href="https://fonts.googleapis.com/css2?family=Source+Code+Pro:wght@500&display=swap" rel="stylesheet">
  <link href="https://fonts.googleapis.com/css?family=Calistoga|Josefin+Sans:400,700|Pacifico&display=swap" rel="stylesheet">
  <!-- bootstrap -->
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9
  <link rel="stylesheet" type= "text/css" href= "{{ url_for('static',filename='css/style.css') }}">
  <!-- fontawesome -->
  <script src="https://kit.fontawesome.com/b3aed9cb07.js" crossorigin="anonymous"></script>

  <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo" crossorigin="anonymous"></
  <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js" integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1" c
  <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js" integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM" cros
  <script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@latest"></script>

</head>

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

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

GITHUB LINK

https://github.com/IBM-EPBL/IBM-Project-1547-1658396674

DEMO LINK

https://drive.google.com/file/d/1xOy4-UO20GChdS1GBacHppJLKOaW5IMn/view?usp=sharing