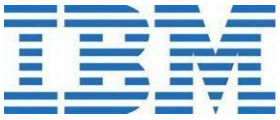
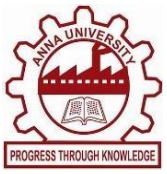




PROJECT REPORT

A Novel Method for Handwritten Digit Recognition System

Submitted by,
PNT2022TMID35735

KRITHIKSHAN K	-	2019503528
VIJAY RAGHAV S	-	2019503061
NITISH KUMAR K M	-	2019503540
RISHIKESH D	-	2019503575

DEPARTMENT OF COMPUTER TECHNOLOGY

MADRAS INSTITUTE OF TECHNOLOGY, ANNA UNIVERSITY

CHENNAI-600044.

Table of Content

CHAPTER 1
INTRODUCTION

1.1 Project Overview:

Handwriting recognition is one of the compelling researches being done because every person in this world has their own way of writing. It is the ability of a computer to automatically recognize and understand handwritten numbers and letters. Advances in the fields of science and technology are digitizing everything to reduce human effort. Therefore, many real-time applications require handwritten digit recognition. The MNIST record is widely used for this recognition process and contains 70000 handwritten digits. Train these images using an artificial neural network to create a deep learning model. A web application is created that allows the user to draw numbers that the model parses, and the recognized results are returned to her UI.

1.2 Purpose:

Handwritten digit recognition is the ability of a computer to recognize digits handwritten by humans. It is one of the leading applications of pattern recognition and machine learning. Despite some limitations, handwriting recognition plays an important role in the modern world and has become a very powerful technology supporting many applications, including automating the sorting of letters and bank checks to make them more complex. We are at the forefront of solving critical problems and empowering people to work.

CHAPTER 2

LITERATURE SURVEY

2.1 Existing Problem:

A fundamental problem with handwritten digit recognition is that the style, size, alignment, margins, and width of handwritten digits vary from person to person and are not always the same. Also, similarities between digits such as 1 and 7, 5 and 6, 3 and 8, 2 and 5, 2 and 7, etc., can cause problems in digit identification. Number structure and appearance. There is also an oblique angle to consider when recognizing handwriting from a photo. The angle at which the photo is taken can obscure the character and make it difficult for a computer to identify.

2.2 References:

- **CHARACTER RECOGNITION USING ARTIFICIAL NEURAL NETWORK**

  Pranjali Pohankar, Namrata Taralkar, Snehalata Karmare, Smita Kulkarni

  International Journal of Electronics Communication and Computer Engineering 5 (4), 2014

- **OFFLINE HANDWRITTEN DIGIT RECOGNITION USING NEURAL NETWORK**

  Sumedha B Hallale, Geeta D Salunke

  International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering 2 (9), 4373-4377, 2013

- **RECOGNITION OF HANDWRITTEN TEXT: ARTIFICIAL NEURAL NETWORK APPROACH**

  Apash Roy, NR Manna

  International Journal of Advanced and Innovative Research (2278-7844) 2 (9), 2013

- **OFFLINE HANDWRITTEN CHARACTER RECOGNITION TECHNIQUES USING NEURAL NETWORK: A REVIEW**

  Vijay Laxmi Sahu, Babita Kubde

  International journal of science and Research (IJSR) 2 (1), 87-94, 2013

- **A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION WITH NEURAL NETWORKS**

  Malothu Nagu, N Vijay Shankar, K. Annapurna

  International Journal of Computer Science and Information Technologies (IJCSIT), Vol. 2 (4) 1685-1692, 2011

- **HANDWRITTEN DIGITS RECOGNITION WITH ARTIFICIAL NEURAL NETWORK**

  K. Islam, G. Mujtaba, R.G. Raj, H.F. Nweke

  2017 International Conference on Engineering Technology and Technopreneurship (ICE2T)

- **CHARACTER RECOGNITION TECHNIQUE USING NEURAL NETWORK**

  Harshal Bodade, Amit Sahu

  International Journal of Engineering Research and Applications 3 (2), 1778-1783, 2013

- **HANDWRITTEN CHARACTER RECOGNITION USING NEURAL NETWORK**

  Chirag I Patel, Ripal Patel, Palak Patel

  International Journal of Scientific & Engineering Research 2 (5), 1-6, 2011

- **RECOGNITION OF HANDWRITTEN TEXT: ARTIFICIAL NEURAL NETWORK APPROACH**

  Apash Roy, NR Manna

  International Journal of Advanced and Innovative Research (2278-7844) 2 (9), 2013

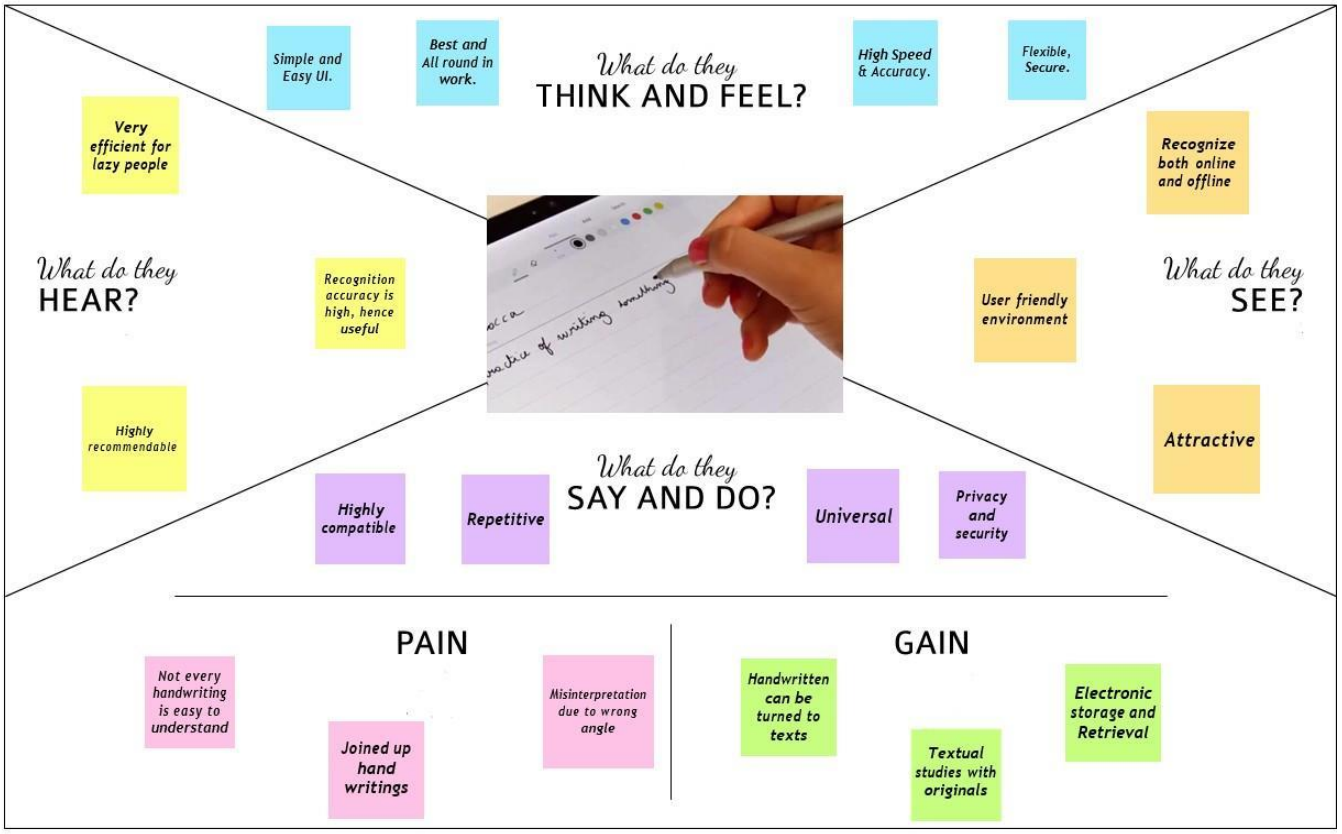
2.3 Problem Statement Definition:

Handwritten digit recognition is the ability of computers to recognize digits handwritten by humans. The problem is challenging because of unique handwritings possessed by every individual - which means any given digit can be written in somewhat different shapes and sizes. A model powerful enough to generalize the core characteristics of each digit can be used to tackle this problem. MNIST Dataset contains images of over 10000 digits written by many different people. A Convolutional Neural Network trained on this dataset can be used to recognize handwritten digits.

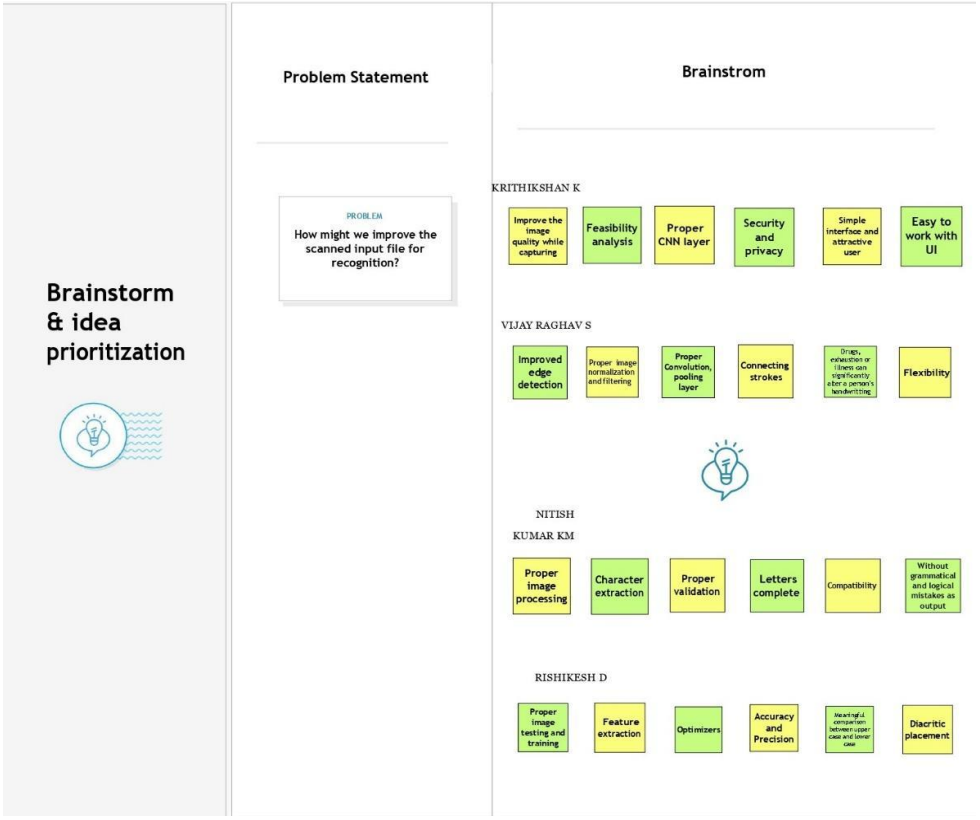
CHAPTER 3

IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas:



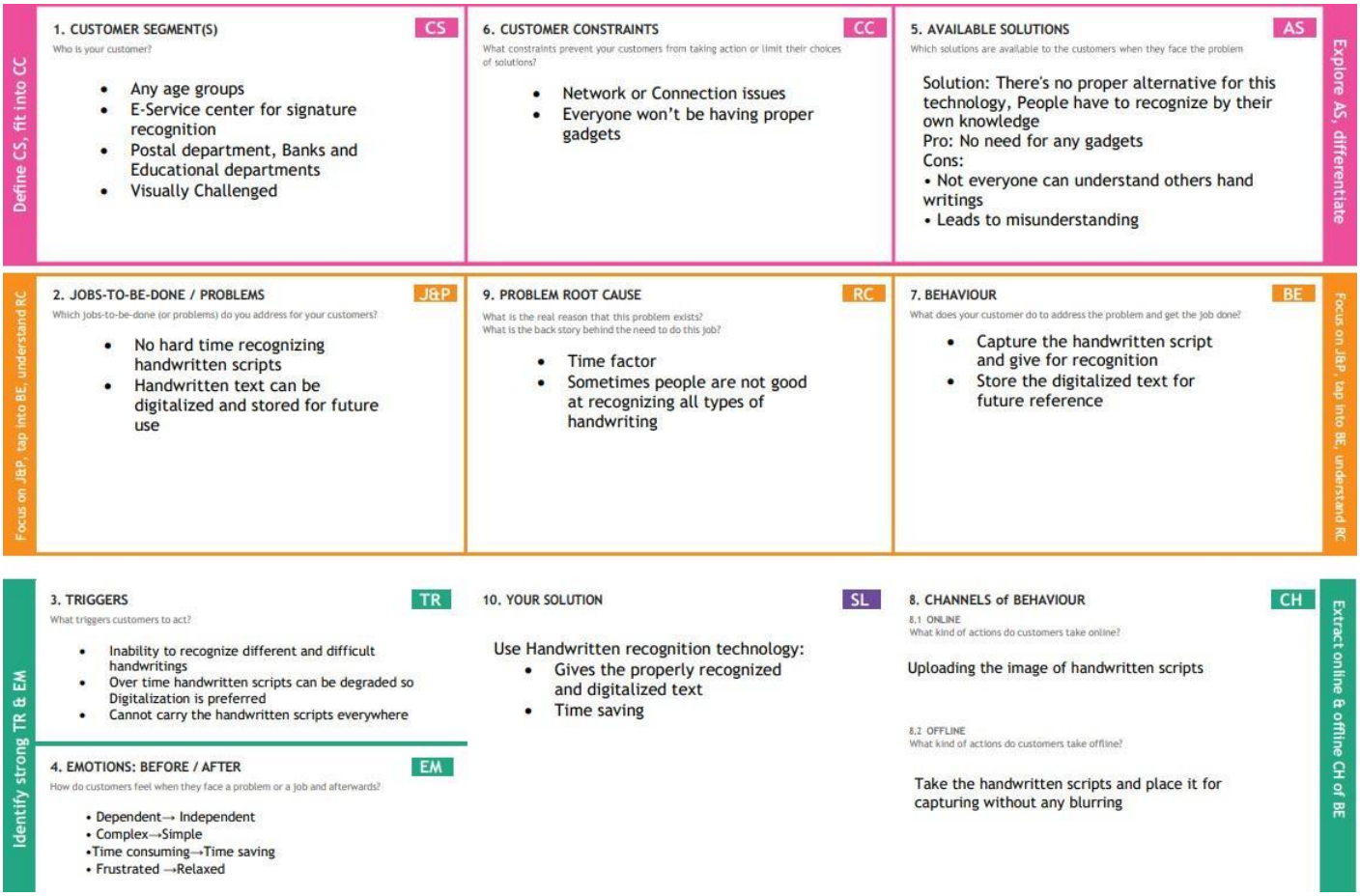
3.2 Ideation & Brainstorming:



3.3 Proposed Solution:

S.No.	Parameter	Description
1.	Problem Statement (Problem to besolved)	● This is a collection of thousands of handwritten pictures used to train classification models using Machine Learning techniques. ● As a part of this problem statement, we will train a multilayer perceptron using Tensorflow-v2 to recognize the handwritten digits.
2.	Idea / Solution description	● 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.
3.	Novelty / Uniqueness	● 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). ● This has been a topic of boundless-research in the field of deep learning.
4.	Social Impact / Customer Satisfaction	● 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 the writing style. ● The generative models can perform recognition driven segmentation.
5.	Business Model (Revenue Model)	● Input module ● Image processing module Segmentation ● module Feature extraction module ● Data set training module Classification module
6.	Scalability of the Solution	● The accuracy of the result for the training data set is 99.98%, and 99.40% with 50% noise by using MNIST. Even we can improve this model to achieve the better resultsby training different types of datasets.

3.4 Problem Solution fit:



CHAPTER 4

REQUIREMENT ANALYSIS

4.1 Functional requirements:

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	The product essentially converts handwritten digits to digital form.	The user is first asked to draw a number on the canvas, and the model that is built is then used to compare the data and provide an output in digitized form.
FR-2	Recognizing the handwrittendigit and displaying.	Recognizing the handwritten digit and displaying.
FR-3	Import dataset file directly to the program from a command that will download the dataset from its website. Save the dataset file in the same directory as the program	Installing packages and applications.
FR-4	Build a Neural Network with a number of nodes in the input layer equal to the number of pixels in the arrays	Nil
FR-5	Activating the Neural Network	Packages – tensorflow

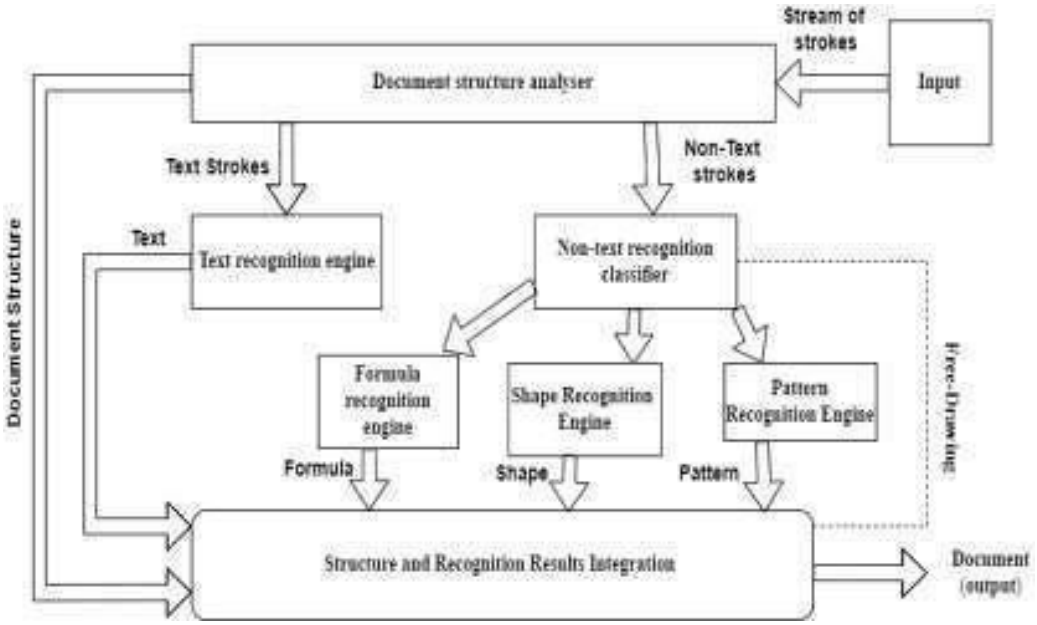
4.2 Non-Functional requirements:

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	System design should be easily understood and user friendly to users. Furthermore, users of all skill levels should be able to navigate it without problems.
NFR-2	Security	The system should automatically be able to authenticate all users with their unique username and password
NFR-3	Performance	Should reduce the delay in information when hundreds of requests are given.
NFR-4	Availability	Information is restricted to each users limited access
NFR-5	Scalability	The system should be able to handle 10000 usersaccessing the site at the same time

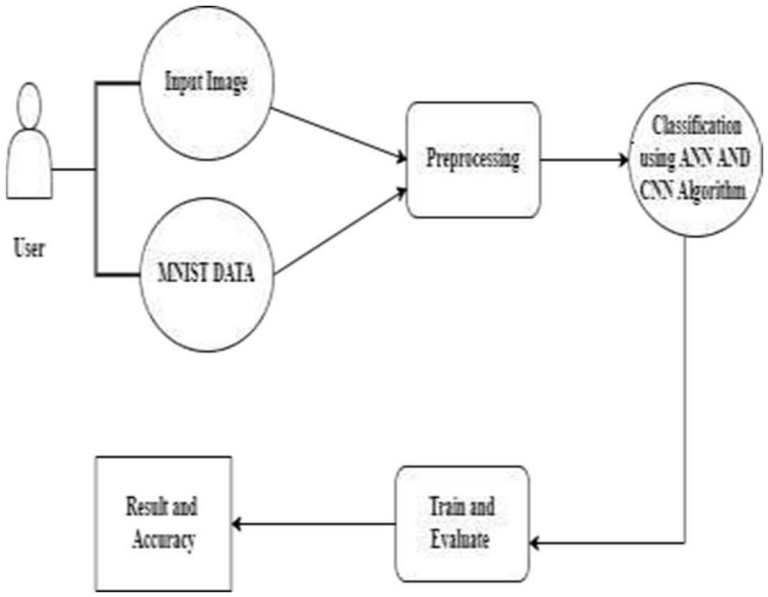
CHAPTER 5

PROJECT DESIGN

5.1 Data Flow Diagram:

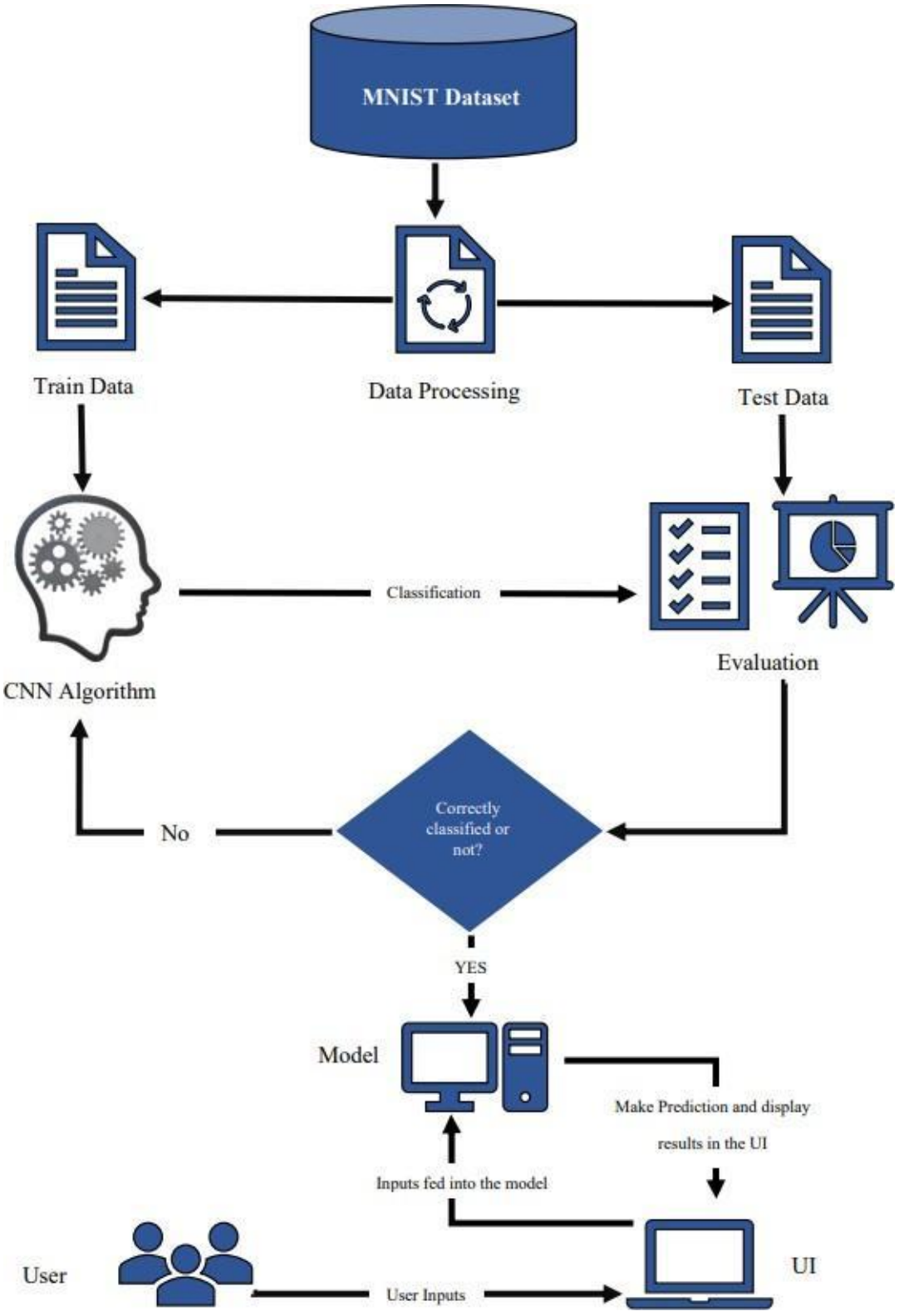


Simplified Diagram:

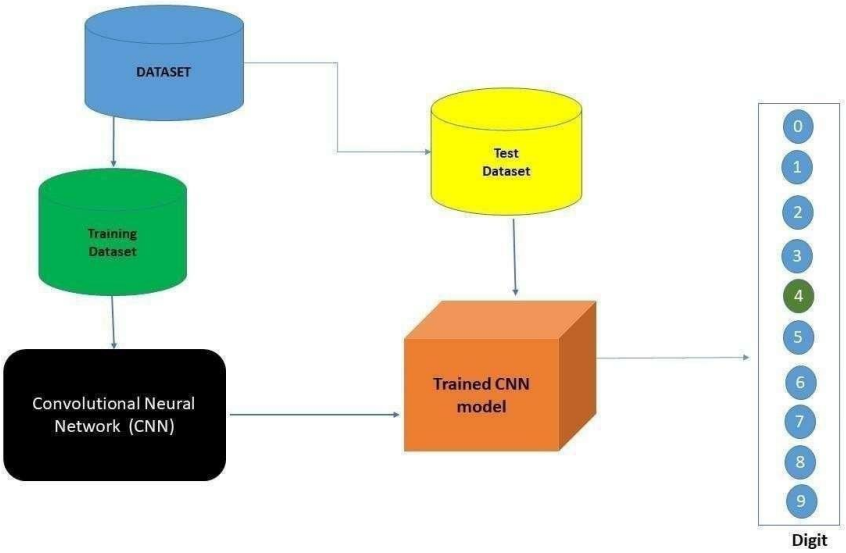


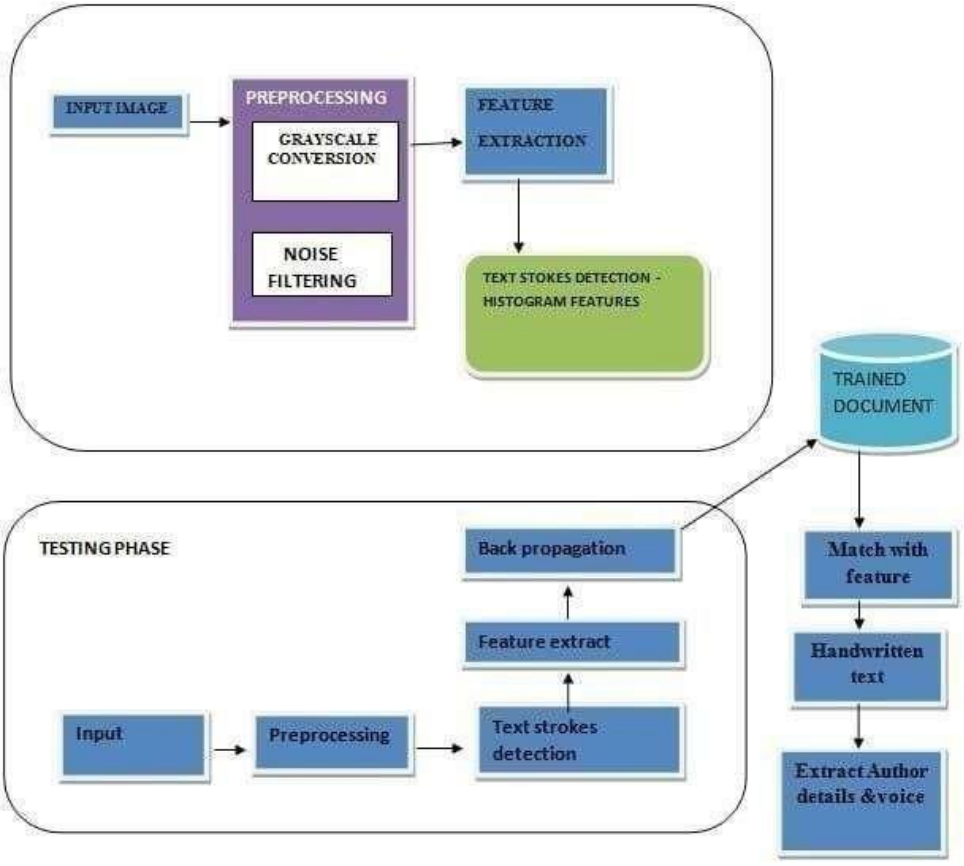
5.2 Solution & Technical Architecture:

Solution Architecture:



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 and awareness to use this application.	I can view the awarenessto use this application and its limitations.	Low	Sprint-1
		USN-2	As a user, I'm allowed to view the guidedvideo to use the interface of this application.	I can gain knowledge touse this application by apractical method.	Low	Sprint-1
		USN-3	As a user, I can read the instructions to usethis application.	I can read instructionsalso to use it in a user-friendly method.	Low	Sprint-2
	Recognize	USN-4	As a user, In this prediction page I get tochoose the image.	I can choose the imagefrom our local system and predict the output.	High	Sprint-2
	Predict	USN-5	As a user, I'm Allowed to upload andchoose the image to be uploaded	I can upload and choosethe image from the system storage and alsoin any virtual storage.	Medium	Sprint-3
		USN-6	As a user, I will train and test the input toget the maximum accuracy of output.	I can able to train and test the application untilit gets maximum accuracy of the result.	High	Sprint-4
		USN-7	As a user, I can access the MNIST data set	I can access the MNISTdata set to produce the accurate result.	Medium	Sprint-3

Customer (Web user)	Home	USN-8	As a user, I can view the guide to use the web app.	I can view the awareness of this application and its limitations.	High	Sprint-1
	Recognize	USN-9	As a user, I can use the web application virtually anywhere.	I can use the application portably anywhere.	High	Sprint-1
		USN-10	As it is an open source, can use it costfreely.	I can use it without any payment to be paid for it to access.	Medium	Sprint-2
		USN-11	As it is a web application, it is installation free	I can use it without the installation of the application or any software.	Medium	Sprint-4
	Predict	USN-12	As a user, I'm Allowed to upload and choose the image to be uploaded	I can upload and choose the image from the system storage and also in any virtual storage.	Medium	Sprint-3

CHAPTER 6

PROJECT PLANNING & SCHEDULING

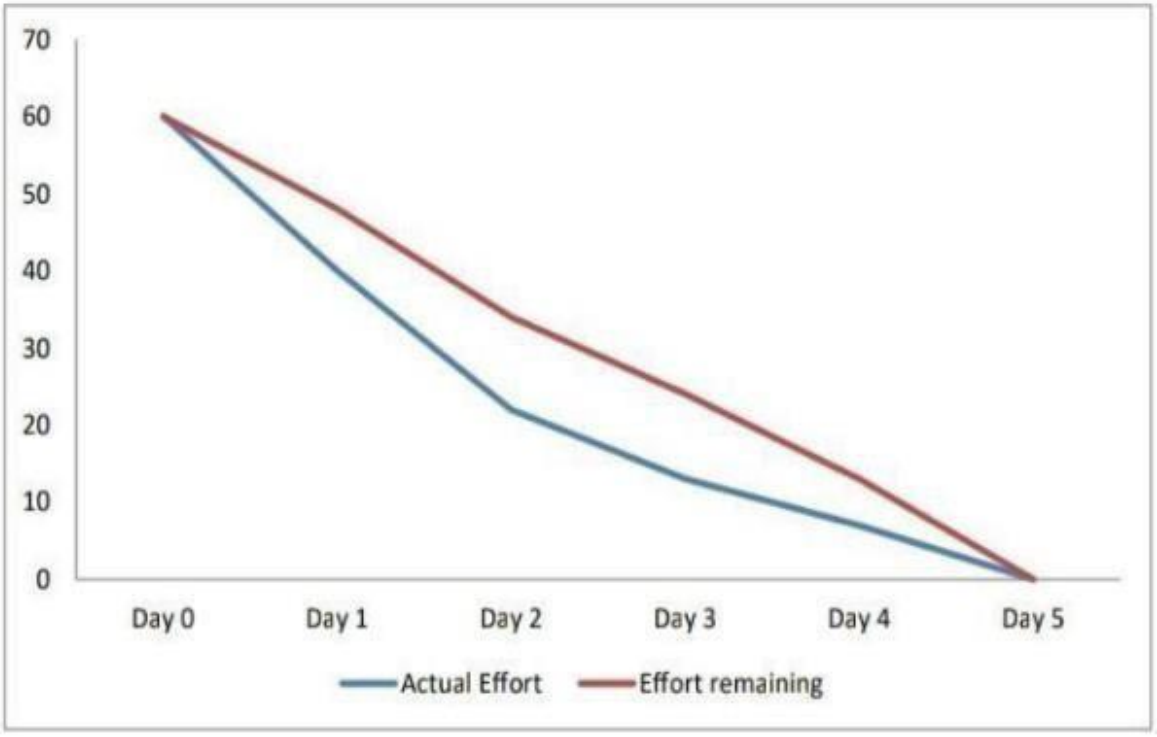
6.1 Sprint Planning & Estimation:

Sprint	Functional Requirement (Epic)	US Number	User Story/Task	Points	Priority	Team members
Sprint-1	Home	USN-1	As a user, I can view the guide and awareness to use this application.	1	Medium	Krithikshan K Vijay Raghav S Nitish Kumar K M Rishikesh D
Sprint-1		USN-2	As a user, I'm allowed to view the guided video to use the interface of this application.	3	High	Krithikshan K Vijay Raghav S Nitish Kumar K M Rishikesh D
Sprint-1		USN-3	As a user, I can read the instructions to use this application.	2	Low	Krithikshan K Vijay Raghav S Nitish Kumar K M Rishikesh D
Sprint-2	Recognize	USN-4	As a user, In this recognition page I get to choose the image.	4	High	Krithikshan K Vijay Raghav S Nitish Kumar K M Rishikesh D
Sprint-3	Predict	USN-5	As a user, I'm Allowed to upload and choose the image to be uploaded	3	Low	Krithikshan K Vijay Raghav S Nitish Kumar K M Rishikesh D
Sprint-3		USN-6	As a user, I will train and test the input to get the maximum accuracy of output.	4	High	Krithikshan K Vijay Raghav S Nitish Kumar K M Rishikesh D
Sprint-3		USN-7	As a user, I can access the MNIST data set	2	Medium	Krithikshan K Vijay Raghav S Nitish Kumar K M Rishikesh D

6.2 Sprint Delivery Schedule:

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

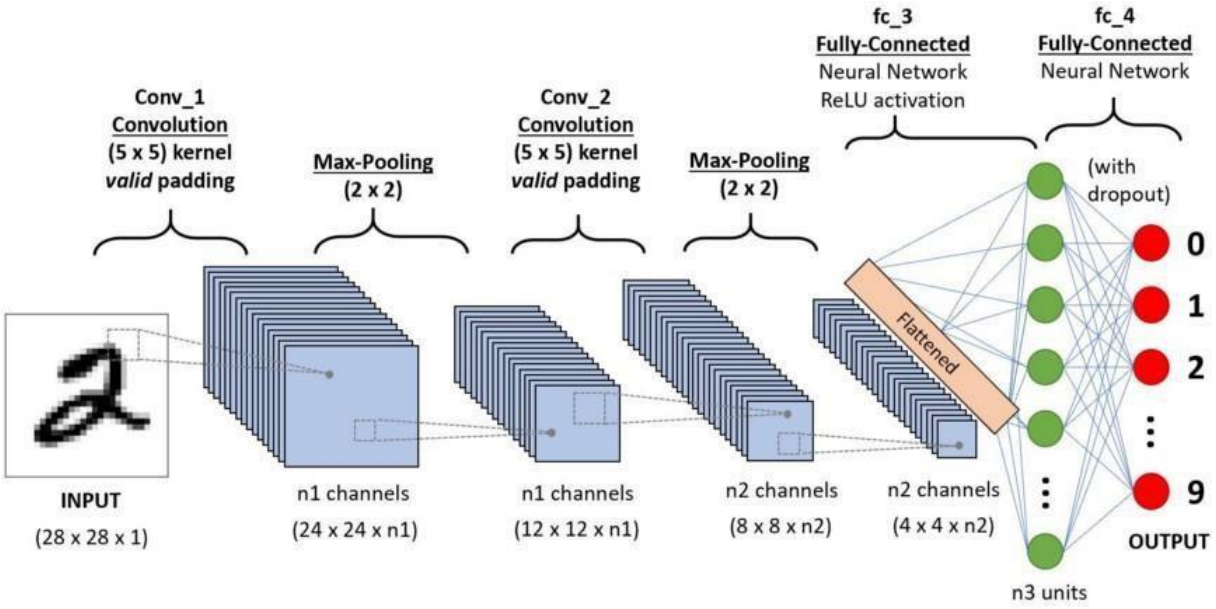
6.3 Reports from JIRA:



CHAPTER 7

CODING & SOLUTIONING

In deep learning, a convolutional neural network (CNN/ConvNet) is a class of deep neural networks, most commonly applied to analyze visual imagery. Now when we think of a neural network we think about matrix multiplications but that is not the case with ConvNet. It uses a special technique called Convolution. Now in mathematics convolution is a mathematical operation on two functions that produces a third function that expresses how the shape of one is modified by the other.



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

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

print('===========')
print(Y_train[0])


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(no_classes,activation='softmax'))

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

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

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

7.1 Feature 2

The flask object implements a WSGI application and acts as the central object. It is passed the name of the module or package of the application. Once it is created it will act as a central registry for the view functions, the URL rules, template configuration and much more.

The name of the package is used to resolve resources from inside the package or the folder the module is contained in depending on if the package parameter resolves to an actual python package (a folder with an _init_.py file inside) or a standard module (just a .py file).

```python
from flask import Flask, render_template, request, jsonify
import numpy as np
from tensorflow import keras
import cv2
import base64

# Initialize flask app
app = Flask(__name__)

# Load prebuilt model
model = keras.models.load_model('digit.h5')

@app.route('/', methods=['GET'])
def home():
    return render_template('home.html')

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

@app.route('/contact')
def contact():
    return render_template('contact.html')
# Handle GET request
@app.route('/drawing', methods=['GET'])
def drawing():
    return render_template('drawing.html')

# Handle POST request
@app.route('/drawing', methods=['POST'])
```

CHAPTER 8

TESTING

8.1 Test Cases:

Defect Analysis:

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	8	3	2	3	16
Duplicate	0	1	0	3	4
External	4	1	0	5	10
Fixed	13	4	3	19	39
Not Reproduced	1	1	0	1	3
Skipped	1	0	1	1	3
Won't Fix	0	0	2	3	5
Totals	27	10	8	35	80

Test Case Analysis:

Section	Total Cases	Not Tested	Fail	Pass
Register	20	0	5	15
Login	30	0	6	24
Redirect to recognition page	68	0	10	58
User input (Drawing)	10	0	0	10
Output prediction	70	0	59	11
Final Model Output	70	0	54	16
Security	20	0	2	18

8.2 User Acceptance Testing:

Test case ID	Feature type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Comments	TC for Automation(Y/N)	BUG ID	Executed By
Detection 01	User Interface webpage	Register Page	Verify user is able to see the Registration page	Active server and internet connection with frontend code for Detection	1. Enter Website URL and Search the URL 2. Display the Register Page to the user	http://127.0.0.1:5000	Register Page will be display withthe Process of Front end	Worked as expected	Pass	User can view the registration page	Yes	--	PNT2022TMID35735 Team
Detection 02	User interface webpage	Register page	Verify user is able to register with user credentials	Active server and internet connection with frontendcode for Detection	1. Enter Website URLand Search the URL 2. Display the Register Page to the User 3. Able to register in registration page	http://127.0.0.1:5000	Successfully registered	Worked as expected	Pass	User can register	Yes	--	PNT2022TMID35735 Team
Detection 03	User interface webpage	Login page	Verify user is able to see the login page	Active server and internet connection with frontendcode for Detection HTML Search Tag with the valid URL	1. Enter Website URL and .Search the URL 2. Display the login Page to the user	http://127.0.0.1:5000	Login Page will be display with the Process of Front end	Worked as expected	Pass	User can view the login page	Yes	--	PNT2022TMID35735 Team
Detection 04	User interface webpage	Login page	Verify user is able to login using user credentials	Active server and internet connection with frontendcode for Detection HTML Search Tag with the valid URL	1. Enter Website URL and Search the URL 2. Display the Login Page to the User 3. Able to login in login page	http://127.0.0.1:5000	Successful login	Worked as expected	Pass	User can login into web app	Yes	--	PNT2022TMID35735 Team
Detection 04	Conversion	Detection Page	Verify user is able to see the recognition page	Active server and internet connection	1. Enter URL (https://127.0.0.1:5000) and check 2. The URL will redirect to theconversion page	http://127.0.0.1:5000	Able to see Recognize page	Worked as expected	Pass	User can see the recognition page	Yes	--	PNT2022TMID35735 Team
Detection 06	Prediction	Detection Page	Verify user can able to draw the digit	Active server and internet connection	1. Enter URL (https://127.0.0.1:5000) and check 2. The URL will redirect to theconversion page 3.Draw the digit	http://127.0.0.1:5000	Able to draw the digit	Worked as expected	Pass	User can draw the digit in the recognition page	Yes	--	PNT2022TMID35735 Team
Detection 07	Prediction	Detection Page	Verify the user able to get the predicted result of the digit drawn	Active server and internet connection	1. Enter URL (https://127.0.0.1:5000) and check 2. The URL will redirect to theconversion page 3.Draw the digit 4.Get predicted output	http://127.0.0.1:5000	Able to get the predicted digit	Worked as expected	Pass	User can get the predicted digit as output	Yes	--	PNT2022TMID35735 Team

CHAPTER 9

RESULTS

Locust Test Report

During: 11/12/2022, 7:05:40 AM - 11/12/2022, 7:14:47 AM

Target Host: http://127.0.0.1:5000/

Script: locust.py

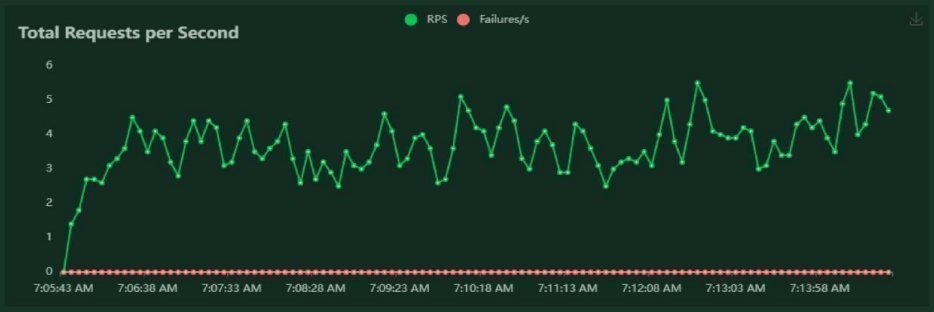
Request Statistics

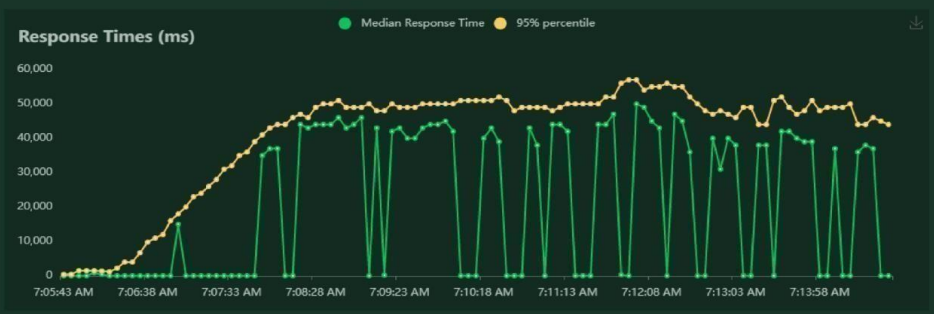
Method	Name	# Requests	# Fails	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	RPS	Failures/s
GET	//	1043	0	13	4	290	1079	1.9	0.0
GET	//predict	1005	0	39648	385	59814	2670	1.8	0.0
	Aggregated	**2048**	**0**	**19462**	**4**	**59814**	**1859**	**3.7**	**0.0**

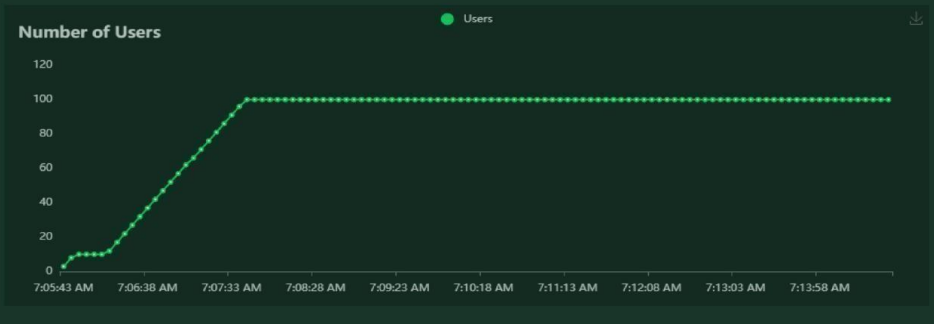
Response Time Statistics

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

Charts







CHAPTER 10
ADVANTAGES & DISADVANTAGES

Advantages:

- It reduces human effort and labor cost.
- It saves times for arranging and sorting huge amount of data
- Only requires far less physical space than the storage of the physical copies.
- Recognising multiple digits on a single frame using sequential model in Keras.
- Data storage, for an example, there are many files, contracts and some personal records that contains some handwritten digits.
- This can be used for sorting through mail by postal code

Disadvantages:

- The system build is complex and holds difficulty
- The handwriting of every individual varies which proves to be a challenge for the system to predict
- Possible unemployment of labor that is typical of technology growth
- The accuracy is not guarantees and there are risks of errors

CHAPTER 11

CONCLUSION

Our project Handwritten Digit Recognition's main purpose is to build an automatic handwritten digit recognition method for the recognition of handwritten digit strings. This project demonstrated a web application that uses machine learning to recognize handwritten numbers. It is done using MNIST dataset and Deep Learning algorithm- multilayer CNN, the use of Keras with Tensorflow that grant the absolute best accuracy. Flask, HTML, CSS, JavaScript, and a few other technologies were used to build this project. 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. Digit recognition is called as the “Hello World of Deep Learning” because it illustrates the basic concepts of Deep learning and it gives a great way to develop more advanced techniques of deep learning. There is so much room for improvement, which can be implemented in subsequent versions.

CHAPTER 12

FUTURE SCOPE

In the future, we can study different architectures of CNNs, namely hybrid CNNs, i.e. CNN-RNN and CNN-HMM models, as well as domain-specific detection systems. The future development of applications based on deep learning and machine learning algorithms is virtually limitless. This project is far from finished and there is a lot of room for improvement. Some of the improvements that can be made to this project are:

- Added support for digitizing multiple images and saving results
- Added support for recognizing multiple digits
- Improve the model to recognize digits from complex images
- Add support for different languages to help users around the world

This project has endless possibilities and can always be improved and made better. Implementing this concept in the real world will benefit multiple industries, reduce the workload for many workers, and improve overall work efficiency.

Source Code:

App:

```
from flask import Flask, render_template, request, jsonify
import numpy as np
from tensorflow import keras
import cv2
import base64
# Initialize flask app
app = Flask(__name__)
# Load prebuilt model
model = keras.models.load_model('digit.h5')
@app.route('/', methods=['GET'])
def home():
    return render_template('home.html')
@app.route('/about')

# Handle GET request
@app.route('/drawing',
methods=['GET']) def drawing():
    return render_template('drawing.html') #
Handle POST request
@app.route('/drawing', methods=['POST'])
def canvas():
    # Recieve base64 data from the user form
    canvasdata = request.form['canvasimg']
    encoded_data = request.form['canvasimg'].split(',')[1]

    # Decode base64 image to python array
```

```
    nparr = np.fromstring(base64.b64decode(encoded_data), np.uint8)
    img = cv2.imdecode(nparr, cv2.IMREAD_COLOR)

    # Convert 3 channel image (RGB) to 1 channel image
    (GRAY) gray_image = cv2.cvtColor(img,
    cv2.COLOR_BGR2GRAY)

    # Resize to (28, 28)
    gray_image = cv2.resize(gray_image, (28, 28), interpolation=cv2.INTER_LINEAR)

    # Expand to numpy array dimenstion to (1, 28, 28)
    img = np.expand_dims(gray_image, axis=0)

    try:
        prediction = np.argmax(model.predict(img))
        print(f"Prediction Result : {str(prediction)}")
        return render_template('drawing.html', response=str(prediction), canvasdata=canvasdata, success=True)
    except Exception as e:
        return render_template('drawing.html', response=str(e), canvasdata=canvasdata)

if_name__== '_main_':
    app.run(debug=True)
```

Model:

```
from tensorflow import keras
import numpy as np
from keras.datasets import mnist
from keras.models import Sequential
from keras import layers
from keras.layers import Dense, Flatten, Conv2D
from keras.optimizer_v1 import Adam
from keras.utils import np_utils
import matplotlib.pyplot as plt
(X_train, Y_train), (X_test, Y_test) = mnist.load_data()
print('*'*20)
print(X_train.shape)
```

```
print(X_test.shape)
print('*'*20)
print(X_train[0])
print('===============')
print(Y_train)
print('_____')
plt.imshow(X_train[0])
X_train = X_train.reshape(60000,28,28,1).astype('float32')
X_test = X_test.reshape(10000,28,28,1).astype('float32')
no_classes=10
Y_train = np_utils.to_categorical(Y_train,no_classes)
Y_test = np_utils.to_categorical(Y_test,no_classes)
print('===========')
print(Y_train[0])
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(no_classes,activation='softmax'))
model.compile(loss='categorical_crossentropy', optimizer='Adam', metrics=['accuracy'])
model.fit(X_train, Y_train, validation_data=(X_test, Y_test),epochs=5, batch_size=32)
metrics= model.evaluate(X_test,Y_test,verbose=0)
print("Metrics")
print(metrics)
prediction=model.predict(X_test[:4])
print(prediction)
print(np.argmax(prediction,axis=1))
print(Y_test[:4])
model.save('digit.h5')
```

HTML:

Homepage:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Digit Recognition</title>
    <link rel="stylesheet" href="/static/css/neww.css">
<link rel="shortcut icon" href="/logos/quill-drawing-a-line.png"
type="image/x-icon">
<style>
        #secDiv{
            position: absolute;
            color:beige ;
            font-family: Verdana, Geneva, Tahoma, sans-serif;
            top: 50%;
            left: 50%;
            transform: translate(-50%,-50%);
            position: absolute;
            width: calc(70% - 20%);
           background-color: rgba(223, 214, 214, 0.25);
           filter: blur(150%);

        }
    </style>

</head>
<body>
    <div class="full-page">
        <div class="navbar">
            <div>
                <a href='/'>Handwritten Digit Recognition System</a>
            </div>
            <nav>
                <ul id='MenuItems'>
                    <li><a href='/drawing'>Recognize</a></li>
                    <li><a href='/contact'>Contact</a></li>
                    <li><a href="/about">About</a></li>
                </ul>
            </nav>
        </div>
    </div>
    <div id="secDiv">
        <p>The handwritten digit recognition is the ability of computers to
recognize human handwritten digits. This handwritten digit recognition is done
using MNIST dataset and Deep Learning algorithm- multilayer CNN, the use of
Keras with Tensorflow that grant the absolute best accuracy. <br>
```

Drawing Page:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Number Recognizer</title>
   <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" integrity="sha384-
JcKb8q3iqJ61gNV9KGb8thSsNjpSL0n8PARn9HuZOnIxN0hoP+VmmDGMN5t9UJ0Z"
crossorigin="anonymous">
   <style>
        body {
                background-color: rgb(170, 170, 170);
        }
      .tools {
         width:400px;
         position:absolute;
         left: 50%;
         transform: translate(-50%,-100%);
      }
      .tools button{
         border: 2px solid rgb(0, 0, 0);
         width: 400px;
         border-radius: 0px;
      }
      .container.mt-4{
         /* background-color: green; */
         top: 50%;
         position: absolute;
         left: 50%;
         transform: translate(-50%,-30%);
         /* box-shadow: black 0 0 50px; */
      }
      #send{
         border-radius: 0 0 8px 8px;
      }

      .navbar.text-light.bg-info{
         background-color: #96fff8 !important;
      }

   </style>
</head>
<body>
   <nav class="navbar text-light bg-info" >
      <div class="mx-auto"><h1 style="font-family: arial; color:black">HANDWRITTEN DIGIT
RECOGNIZER</h1></div>
   </nav>
   {% if response %}
      {% if success %}
         <p class="alert alert-success m-4 text-center" style="background-color:#fcfc9f">PREDICTED
```

```
DIGIT: {{ response }}</p>
      {% else %}
         <p class="alert alert-success m-4 text-center">{{ response }}</p>
      {% endif %}
   {% endif %}
   <div class="container mt-4" >
      <div class="row" style="width: 280px; margin: auto;" >
         <div class="tools d-flex justify-content-between">
            <button class="btn" id="clear" style="background-color:#fcbcff;
color:black">Clear</button>
            <button class="btn" id="pastel" onclick="pastel()" style="background-color:#a5ffcd;
color:black">Pastel</button>
            <button class="btn" id="eraser" onclick="eraser()" style="background-
color:#c899ff">Eraser</button>
         </div>
      </div>
      <form action="/drawing" method="POST" onsubmit="canvastoimage()">
         <div class="row">
            <div class="col d-flex justify-content-center">
               <input type="hidden" id="canvasimg" name="canvasimg">
               <input type="hidden" id="canvasdata" name="canvasdata" value="{{ canvasdata }}">
               <canvas id="canvas" width="400" height="280"></canvas>
            </div>
         </div>
         <div class="row">
            <div class="col d-flex justify-content-center">
               <button style="width: 400px; background-color:#a8b2ff; color:black" class="btn"
id="send" type="submit">DETECT NUMBER</button>
            </div>
         </div>
      </form>
   </div>

   <script>
      // Canvas Drawing
      window.addEventListener('load', ()=>{
         const canvas = document.querySelector('#canvas');
         const context = canvas.getContext('2d');

         const canvasdata = document.querySelector('#canvasdata').value;
         if (canvasdata){
            const image = new Image();
            image.onload = ()=>{
               context.drawImage(image, 0, 0);
            };
            image.src = canvasdata;
         } else {
            context.fillStyle = "black";
            context.fillRect(0, 0, canvas.width, canvas.height);
         }

         let radius = 10;
         let start = 0;
```

```
let end = Math.PI * 2;
let dragging = false;

context.lineWidth = radius * 2;
context.lineCap = 'round';

const putPoint = (e)=>{
    if (dragging){
        context.fillStyle = "black";
        context.strokeStyle = "black";
        context.lineTo(e.offsetX, e.offsetY);
        context.stroke();
        context.beginPath();
        context.arc(e.offsetX, e.offsetY, radius, start, end);
        context.fill();
        context.beginPath();
        context.moveTo(e.offsetX, e.offsetY);
    }
}

const engage = (e)=>{
    dragging = true;
    putPoint(e);
}

const disengage = ()=>{
    dragging = false;
    context.beginPath();
}

canvas.addEventListener('mousedown', engage);
canvas.addEventListener('mousemove', putPoint);
canvas.addEventListener('mouseup', disengage);

const clear = document.querySelector('#clear');
clear.addEventListener('click', ()=>{
    const canvas = document.querySelector('#canvas');
    const context = canvas.getContext('2d');
    context.filter = 'invert(0)';
    context.fillStyle = "black";
    context.fillRect(0, 0, canvas.width, canvas.height);
});

// Pastel
const pastel = document.querySelector('#pastel');
pastel.addEventListener('click', ()=>{
    const canvas = document.querySelector('#canvas');
    const context = canvas.getContext('2d');
    context.filter = 'invert(1)';
    radius = 10;
});

// Eraser
```

```
            const eraser = document.querySelector('#eraser');
            eraser.addEventListener('click', ()=>{
                const canvas = document.querySelector('#canvas');
                const context = canvas.getContext('2d');
                context.filter = 'invert(0)';
                radius = 30;
            });
        });

        // Canvas Submit
        const canvastoimage = ()=>{
            const canvas = document.querySelector('#canvas');
            document.getElementById('canvasimg').value = canvas.toDataURL();
        };
    </script>
    <script src="https://code.jquery.com/jquery-3.5.1.min.js" integrity="sha256-
9/aliU8dGd2tb6OSsuzixeV4y/faTqgFtohetphbbj0=" crossorigin="anonymous"></script>
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"
integrity="sha384-
B4gt1jrGC7Jh4AgTPSdUtOBvfO8shuf57BaghqFfPlYxofvL8/KUEfYiJOMMV+rV"
crossorigin="anonymous"></script>
</body>
</html>
```

GitHub Link:

https://github.com/IBM-EPBL/IBM-Project-5148-1658748838

Project Demo Link:

https://github.com/IBM-EPBL/IBM-Project-5148-1658748838/blob/main/Final%20Deliverables/Demo%20Video/IBM-DIGIT_RECOGNIZER.mp4