PROJECT REPORT

A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION

Submitted by

PNT2022TMID21349

Bhuvanshree B - 917719D015

Ishwarya S - 917719D034

Soumya V - 917719D097

Tharsana S - 917719D102

TABLE OF CONTENTS

1 INTRODUCTION 1

1.1 PROJECT OVERVIEW 1

1.2 PURPOSE 1

2 LITERATURE SURVEY 2

2.1 EXISTING PROBLEM 2

2.2 REFERENCES 2

2.3 PROBLEM STATEMENT DEFINITION 3

3 IDEATION AND PROPOSED SOLUTION 4

3.1 EMPATHY MAP CANVAS 4

3.2 IDEATION & BRAINSTORMING 4

3.3 PROPOSED SOLUTION 7

3.4 PROBLEM SOLUTION FIT 8

4 REQUIREMENT ANALYSIS 8

4.1 FUNCTIONAL REQUIREMENTS 8

4.2 NON FUNCTIONAL REQUIREMENTS 9

5 PROJECT DESIGN 9

5.1 DATA FLOW DIAGRAM 9

5.2 SOLUTION & TECHNICAL ARCHITECTURE 10

5.3 USER STORIES 11

6 PROJECT PLANNING AND SCHEDULING 12

6.1 SPRINT PLANNING AND ESTIMATION 12

6.2 SPRINT DELIVERY SCHEDULE 12

7 CODING & SOLUTIONING 13

8 TESTING 18

8.1 TEST CASES 18

8.2 USER ACCEPTANCE TESTING 19

8.2.1 DEFECT ANALYSIS 19

8.2.2 TEST CASE ANALYSIS 19

9 RESULTS 20

9.1 PERFORMANCE METRICS 20

10 ADVANTAGES & DISADVANTAGES 21

10.1 ADVANTAGES 21

10.2 DISADVANTAGES 21

11 CONCLUSION 22

12 FUTURE SCOPE 23

13 APPENDIX 24

SOURCE CODE 24

GITHUB 27

PROJECT DEMO 27

1. INTRODUCTION

1.1 Project Overview

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. Handwritten character recognition is one of the practically important issues in pattern recognition applications.

The applications of digit recognition include postal mail sorting, bank check processing, form data entry, etc. The heart of the problem lies within the ability to develop an efficient algorithm that can recognize handwritten digits and which is submitted by users by the way of a scanner, tablet, and other digital devices.

1.2 Purpose

The main purpose of this project is to build an automatic handwritten digit recognition method for the recognition of handwritten digit strings. The total project lies with a great computation speed and by a online server where run and compilation done quickly.

To accomplish the recognition task, first, the digits will be segmented into individual digits. Then, a digit recognition module is employed to classify each segmented digit completing the handwritten digit string recognition task.

The better functionality of this project can be license plate verification. Car license plate can be checked and one can set the record rightfully that which car is passing the gate and when by the recognition of characters.

2. LITERATURE SURVEY

2.1 Existing problem

Handwriting recognition tends to have problems when it comes to accuracy. sometimes, characters look very similar, making it hard for a computer to recognise accurately. When letters all connect, it makes it hard for computers to recognise individual characters. In the case of handwriting recognition from photos, there are awkward angles to consider. The angle the photo is taken could obscure the character, making it harder for the computer to identify.

2.2 References

- **Handwritten Digit Recognition System Based on Convolutional Neural Network - August 2020**

Jinze Li; Gongbo Sun; Leiye Yi; Qian Cao; Fusen Liang; Yu Sun

An offline recognition system for handwritten digits based on convolutional neural networks. The application of this system greatly reduce labor costs and improve work efficiency, which is of great significance in many fields.

- **Handwritten Character Recognition of MODI Script using Convolutional Neural Network Based Feature Extraction Method and Support Vector Machine Classifier - October 2020**

Solley Joseph; Jossy George

Implementation of a feature extraction method using CNN autoencoder for MODI script character recognition. This project is to propose a novel feature extraction model by learning a Bag of Features Framework for handwritten text recognition based on Deep Sparse Auto-Encoder. Gives very high accuracy and is better compared to the most accurate MODI character recognition method

- **Handwritten English word recognition using a deep learning based object detection architecture - September 2021**

RiktimMondal, SamirMalakar, ElisaH.BarneySmith, RamSarkar

To propose a handwriting recognition technique to recognize handwritten English text based on a YOLOv3 object recognition model that is lexicon-free and that performs sequential character detection and identification with a low number of training samples. Most of the methods in the literature use lexicon-based approaches and train their models on large datasets having near 50 K word samples to achieve good results. This results in high computational requirements.

- **Handwriting Recognition for Medical Prescriptions using a CNN-Bi-LSTM Model- May 2021, IEEE Explore**

Tavish Jain, Rohan Sharma, Ruchika Malhotra

To develop a technique that is specially trained to recognize medical prescriptions correctly. They have self built a corpus manually containing the terms widely used in the medical domain, commonly used in prescriptions. We then use string matching algorithms, and string distance functions to find the nearest word in the corpus, so that bias is given to medical terms for increasing accuracy of the predicted output.

- **Handwritten Words and Digits Recognition using Deep Learning Based Bag of Features Framework-February 2020**

Najoua Raha, Maroua Tounsi, Tarek M Hamdani, Adel M Alimi

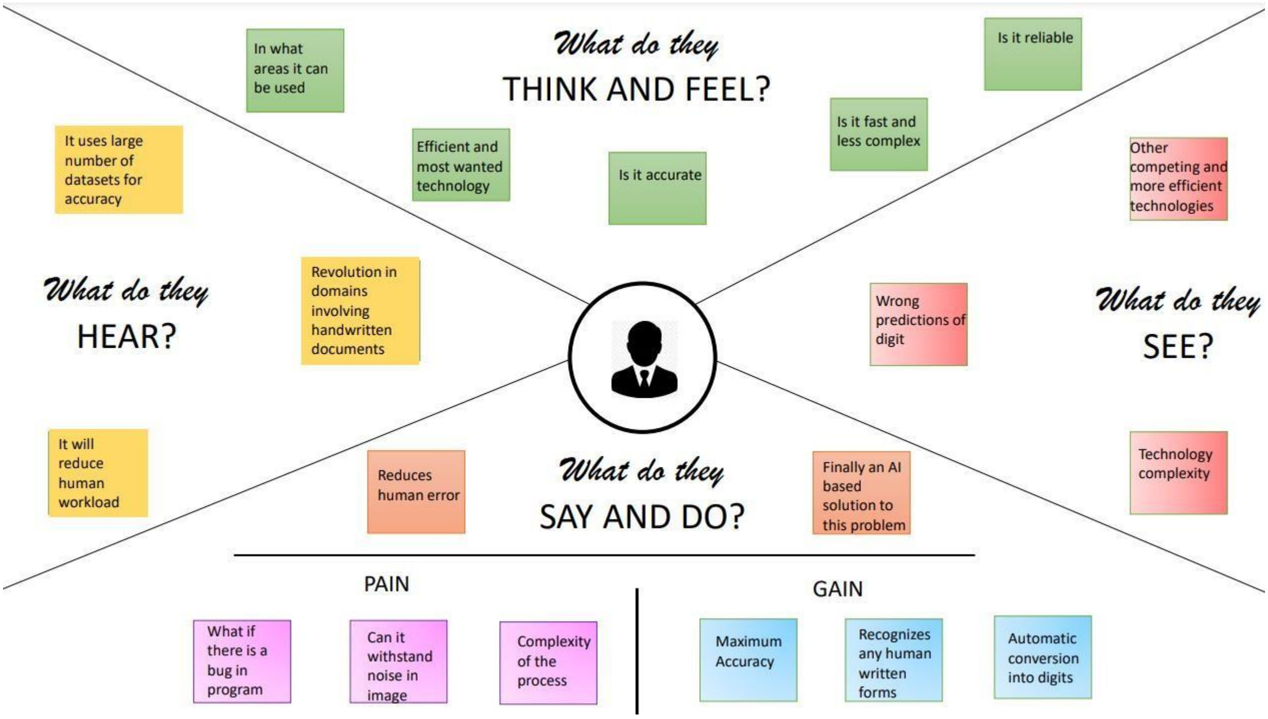
This project is to propose a novel feature extraction model by learning a Bag of Features Framework for handwritten text recognition based on Deep Sparse Auto-Encoder. This method achieves promising recognition on both datasets (IFN/ENIT word images benchmark and MNIST handwritten digits.)

2.3 Problem Statement Definition

The goal of this project is to create a model that will be able to recognize and determine the handwritten digits from its image by using the concepts of Convolution Neural Network. Though the goal is to create a model which can recognize the digits, it can be extended to letters and an individual’s handwriting.

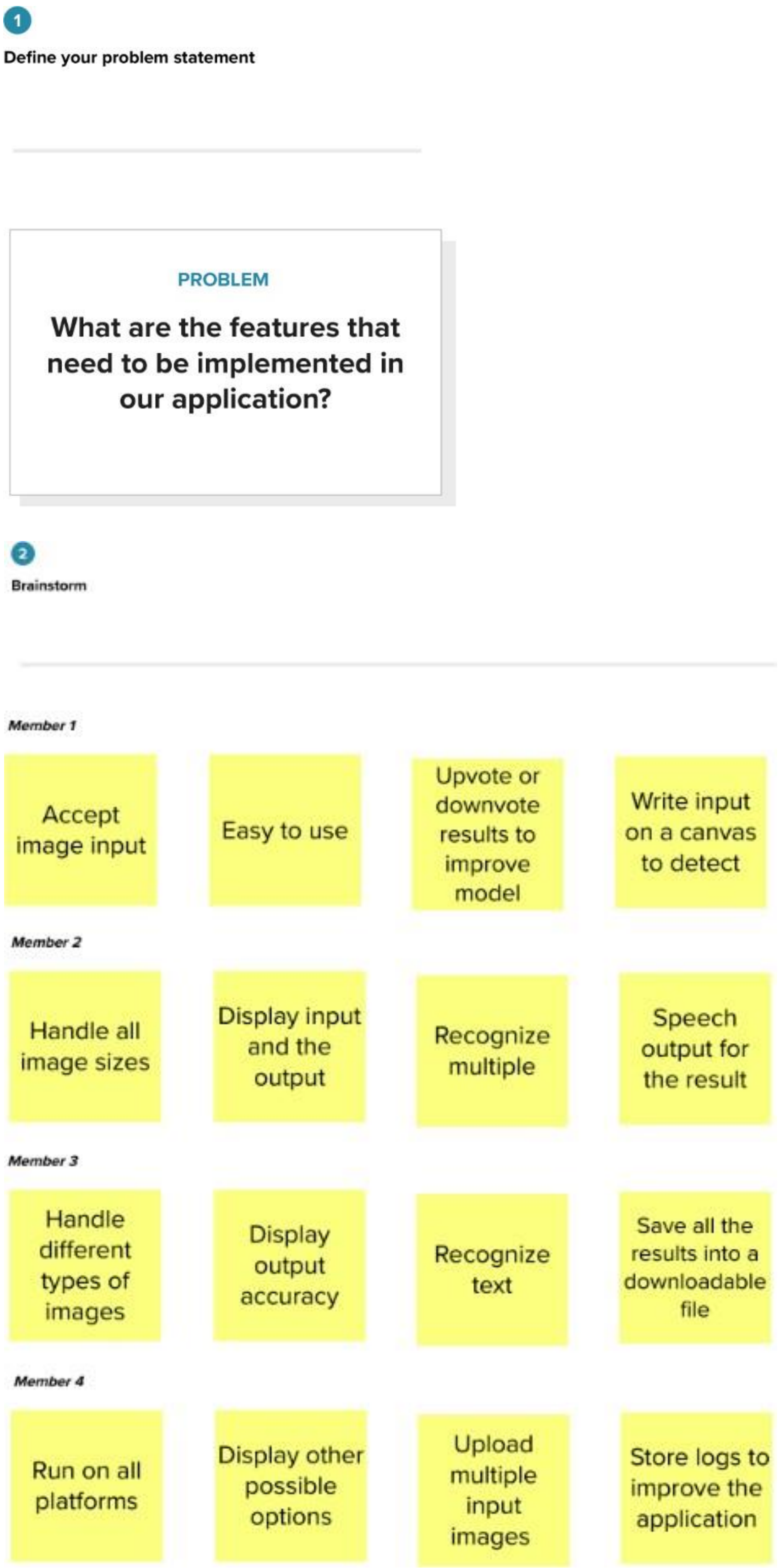
3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas


What do they
THINK AND FEEL?
In what areas it can be used
Efficient and most wanted technology
Is it accurate
Is it fast and less complex
Is it reliable
What do they
HEAR?
It uses large number of datasets for accuracy
Revolution in domains involving handwritten documents
It will reduce human workload
What do they
SEE?
Other competing and more efficient technologies
Wrong predictions of digit
Technology complexity
What do they
SAY AND DO?
Reduces human error
Finally an AI based solution to this problem
PAIN
What if there is a bug in program
Can it withstand noise in image
Complexity of the process
GAIN
Maximum Accuracy
Recognizes any human written forms
Automatic conversion into digits

3.2 Ideation & Brainstorming

Digit recognition system is the working of a machine to train itself or 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 by hand.The uniqueness and variety in the handwriting of different individuals influences the formation and appearance of the digits



1

Define your problem statement

PROBLEM

What are the features that need to be implemented in our application?

2

Brainstorm

Member 1

Accept image input	Easy to use	Upvote or downvote results to improve model	Write input on a canvas to detect

Member 2

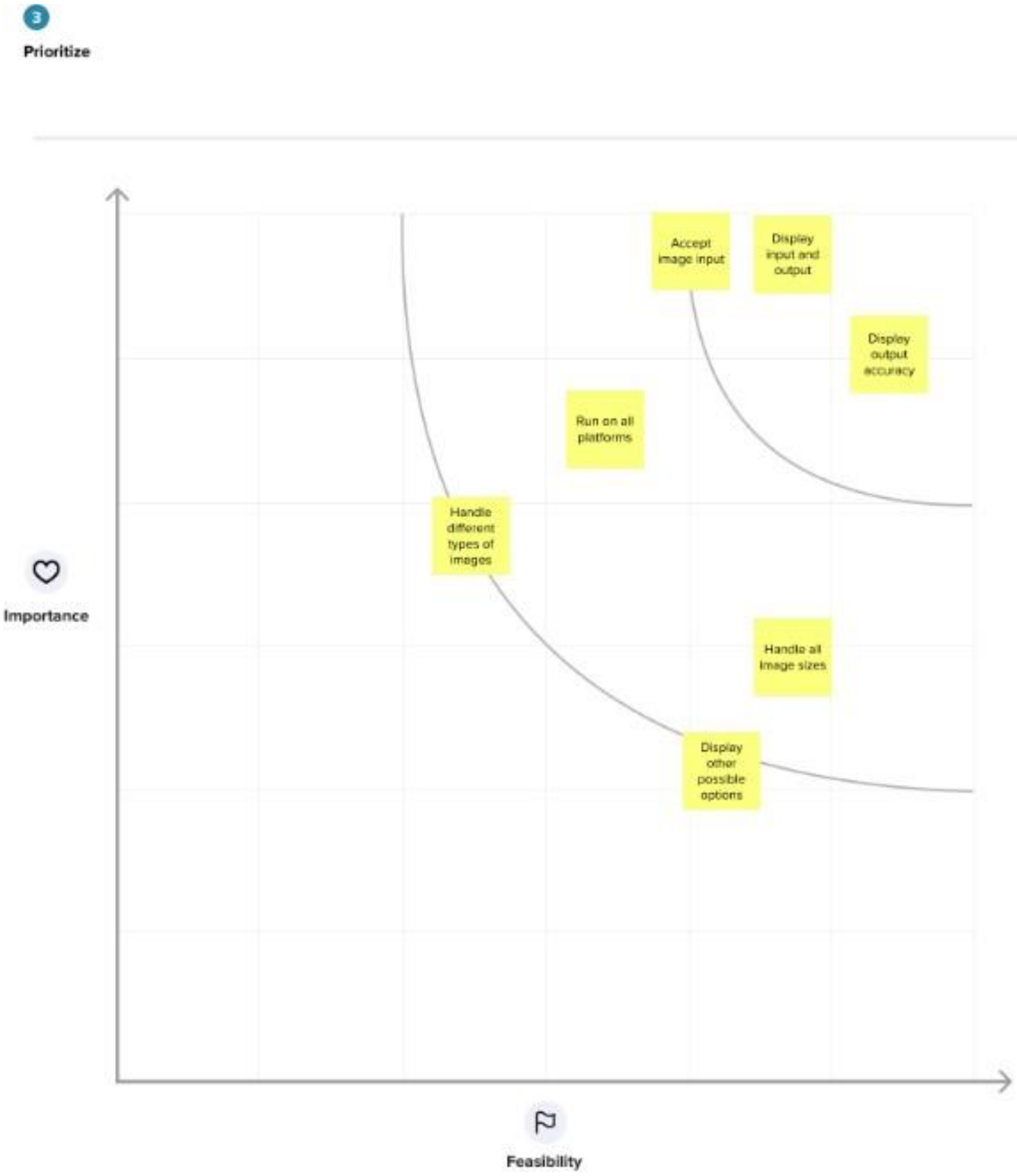
Handle all image sizes	Display input and the output	Recognize multiple	Speech output for the result

Member 3

Handle different types of images	Display output accuracy	Recognize text	Save all the results into a downloadable file

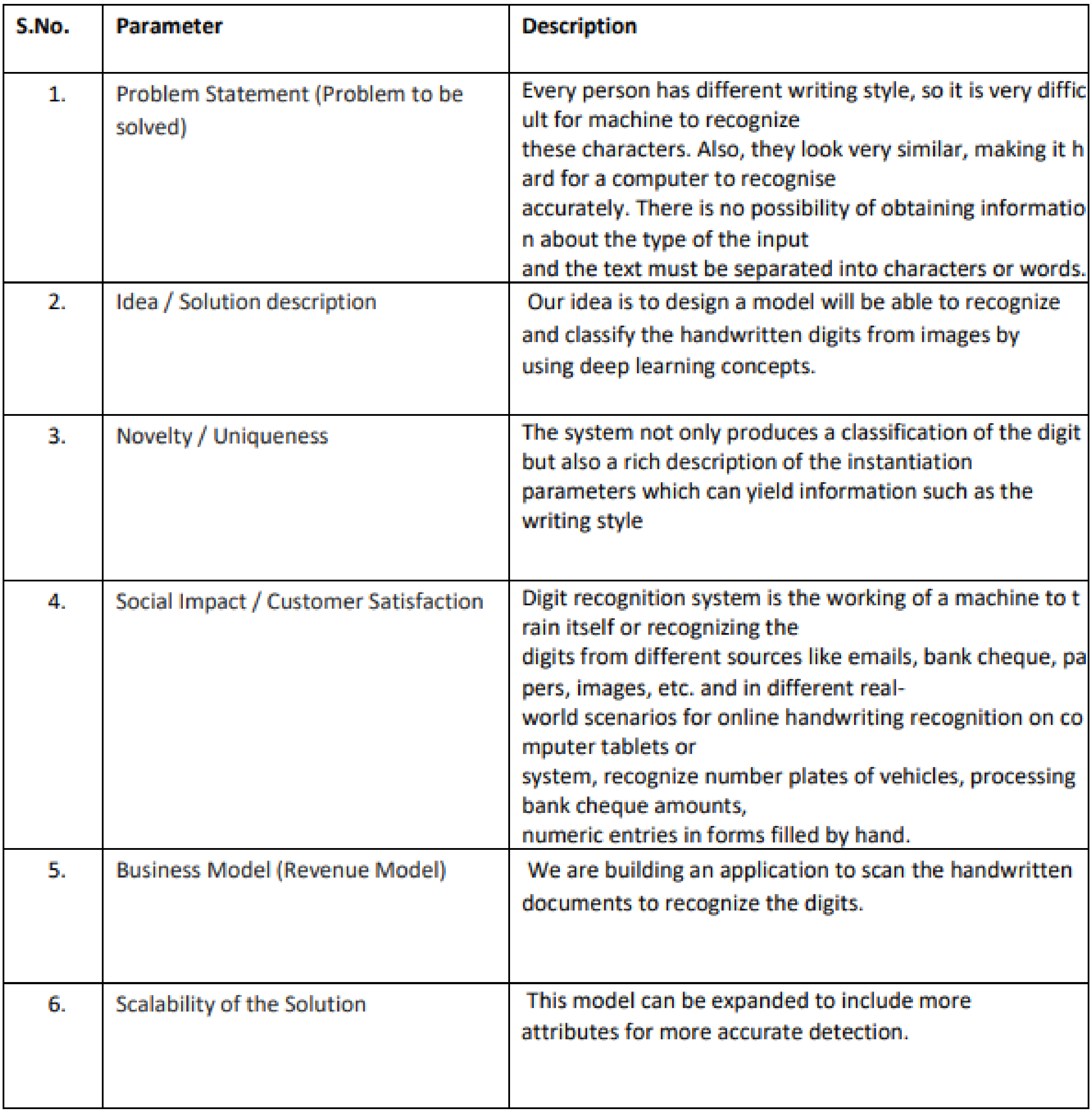
Member 4

Run on all platforms	Display other possible options	Upload multiple input images	Store logs to improve the application



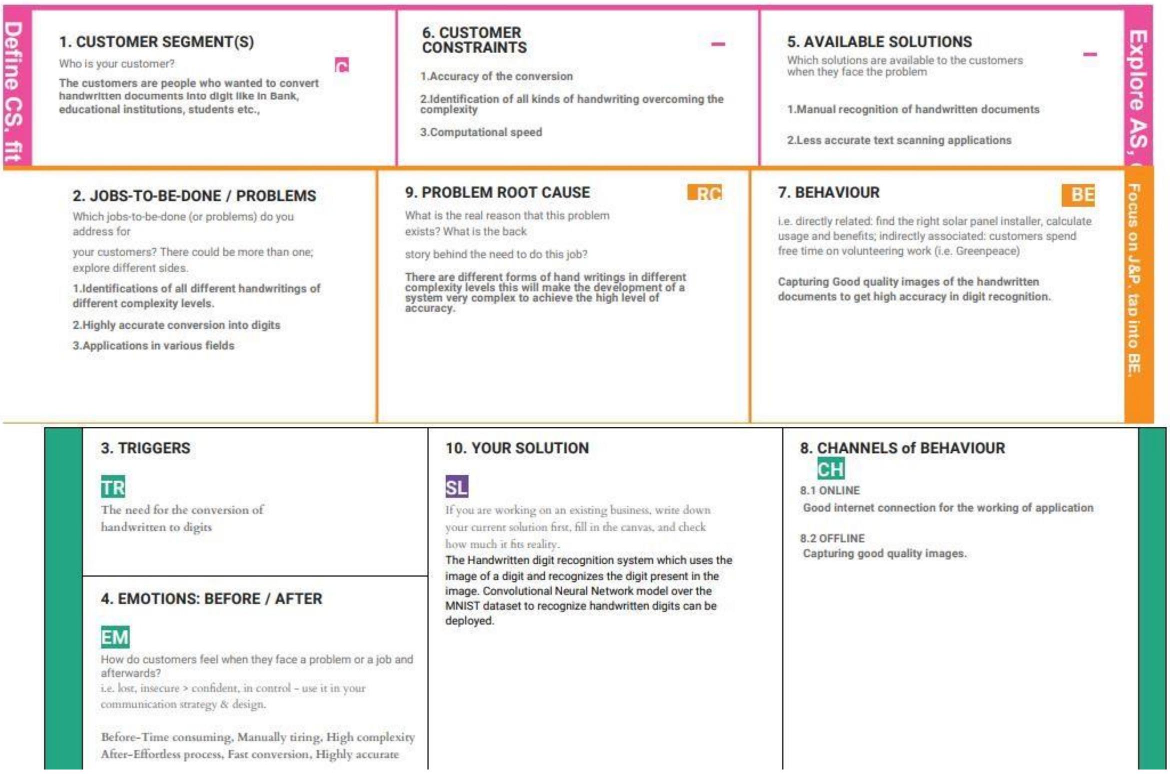
3
Prioritize
Accept image input
Display input and output
Display output accuracy
Run on all platforms
Handle different types of images
Handle all image sizes
Display other possible options
Importance
Feasibility

3.3 Proposed Solution



S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Every person has different writing style, so it is very difficult for machine to recognize these characters. Also, they look very similar, making it hard for a computer to recognise accurately. There is no possibility of obtaining information about the type of the input and the text must be separated into characters or words.
2.	Idea / Solution description	Our idea is to design a model will be able to recognize and classify the handwritten digits from images by using deep learning concepts.
3.	Novelty / Uniqueness	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
4.	Social Impact / Customer Satisfaction	Digit recognition system is the working of a machine to train itself or 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 by hand.
5.	Business Model (Revenue Model)	We are building an application to scan the handwritten documents to recognize the digits.
6.	Scalability of the Solution	This model can be expanded to include more attributes for more accurate detection.

3.4 Problem Solution fit


Define CS, fit
Explore AS,
Focus on J&P, tap into BE,
CS
RC
BE
TR
EM
SL
CH

1. CUSTOMER SEGMENT(S)

Who is your customer?

The customers are people who wanted to convert handwritten documents into digit like in Bank, educational institutions, students etc.,

6. CUSTOMER CONSTRAINTS

1.Accuracy of the conversion

2.Identification of all kinds of handwriting overcoming the complexity

3.Computational speed

5. AVAILABLE SOLUTIONS

Which solutions are available to the customers when they face the problem

1.Manual recognition of handwritten documents

2.Less accurate text scanning applications

2. JOBS-TO-BE-DONE / PROBLEMS

Which jobs-to-be-done (or problems) do you address for

your customers? There could be more than one; explore different sides.

1.Identifications of all different handwritings of different complexity levels.

2.Highly accurate conversion into digits

3.Applications in various fields

9. PROBLEM ROOT CAUSE

What is the real reason that this problem exists? What is the back

story behind the need to do this job?

There are different forms of hand writings in different complexity levels this will make the development of a system very complex to achieve the high level of accuracy.

7. BEHAVIOUR

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)

Capturing Good quality images of the handwritten documents to get high accuracy in digit recognition.

3. TRIGGERS

The need for the conversion of handwritten to digits

4. EMOTIONS: BEFORE / AFTER

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.

Before-Time consuming, Manually tiring, High complexity
After-Effortless process, Fast conversion, Highly accurate

10. YOUR SOLUTION

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.
The Handwritten digit recognition system which uses the image of a digit and recognizes the digit present in the image. Convolutional Neural Network model over the MNIST dataset to recognize handwritten digits can be deployed.

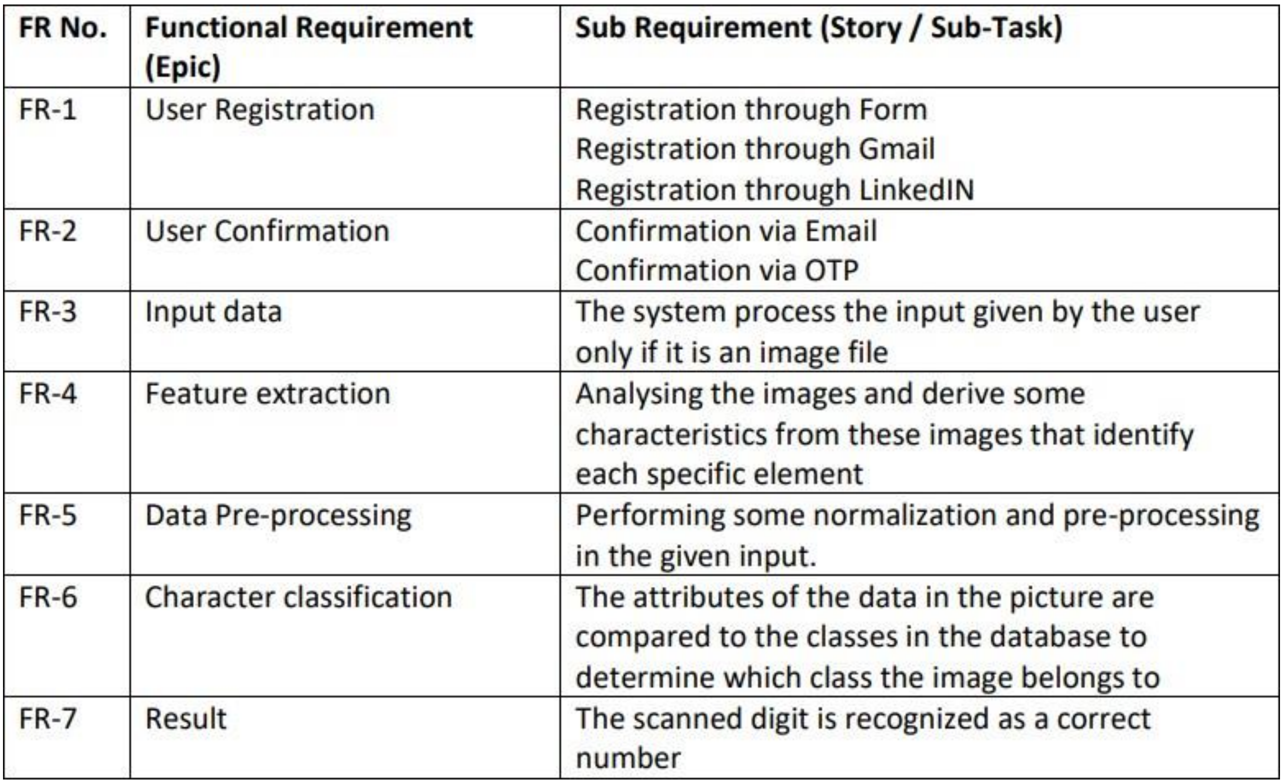
8. CHANNELS of BEHAVIOUR

8.1 ONLINE
Good internet connection for the working of application

8.2 OFFLINE
Capturing good quality images.

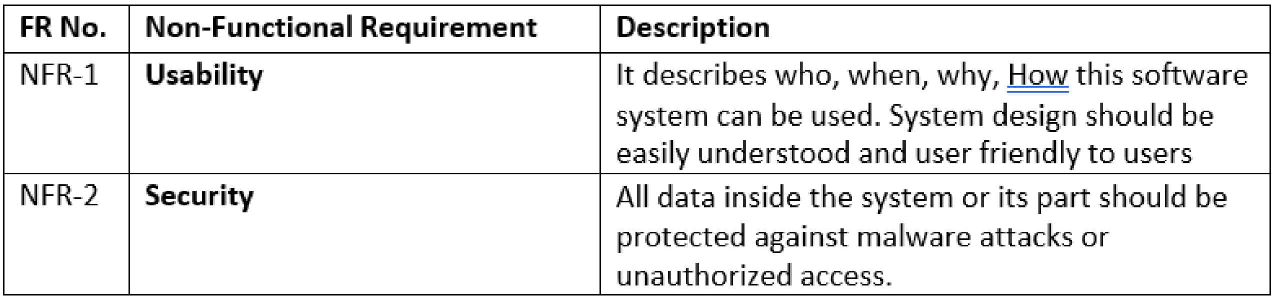
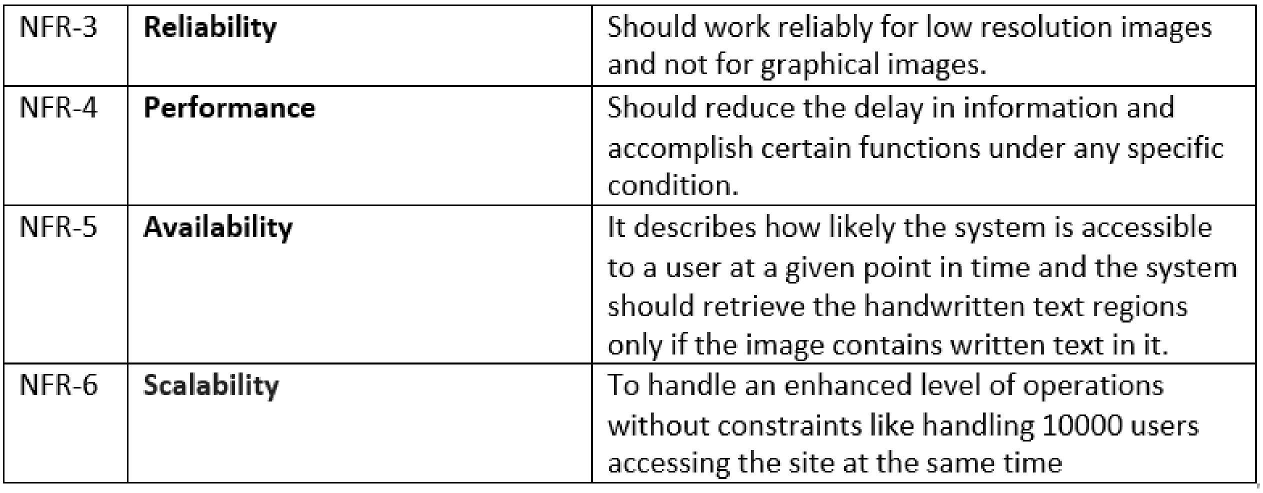
4. REQUIREMENT ANALYSIS

4.1 Functional requirement



FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Input data	The system process the input given by the user only if it is an image file
FR-4	Feature extraction	Analysing the images and derive some characteristics from these images that identify each specific element
FR-5	Data Pre-processing	Performing some normalization and pre-processing in the given input.
FR-6	Character classification	The attributes of the data in the picture are compared to the classes in the database to determine which class the image belongs to
FR-7	Result	The scanned digit is recognized as a correct number

4.2 Non-Functional requirements

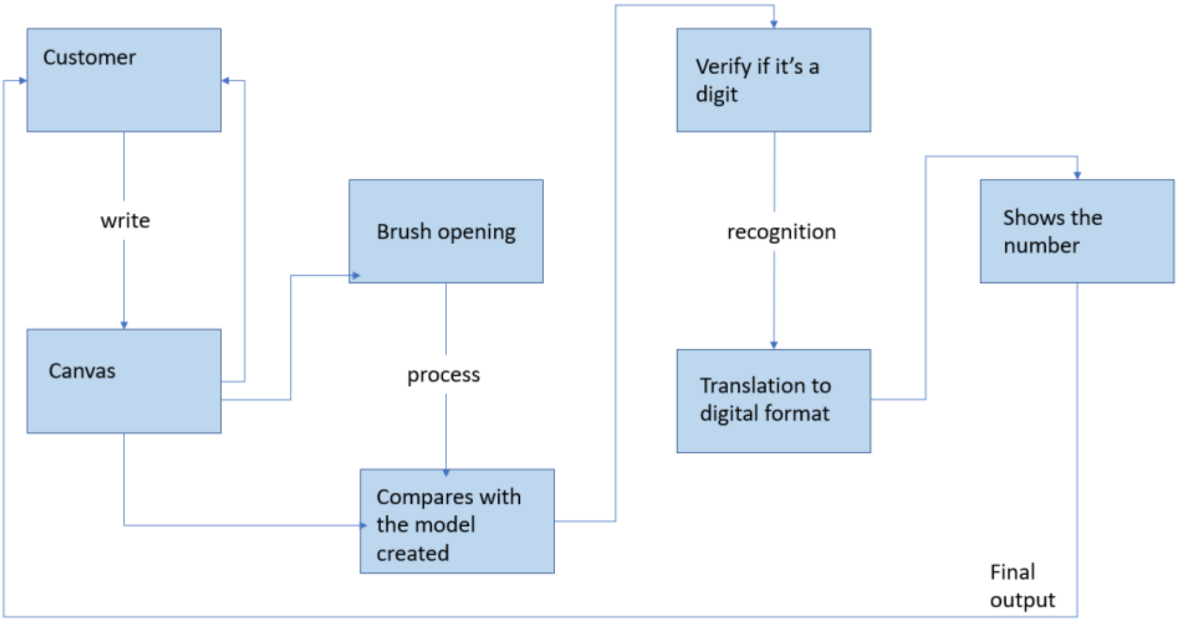



FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	It describes who, when, why, How this software system can be used. System design should be easily understood and user friendly to users
NFR-2	**Security**	All data inside the system or its part should be protected against malware attacks or unauthorized access.
NFR-3	**Reliability**	Should work reliably for low resolution images and not for graphical images.
NFR-4	**Performance**	Should reduce the delay in information and accomplish certain functions under any specific condition.
NFR-5	**Availability**	It describes how likely the system is accessible to a user at a given point in time and the system should retrieve the handwritten text regions only if the image contains written text in it.
NFR-6	**Scalability**	To handle an enhanced level of operations without constraints like handling 10000 users accessing the site at the same time

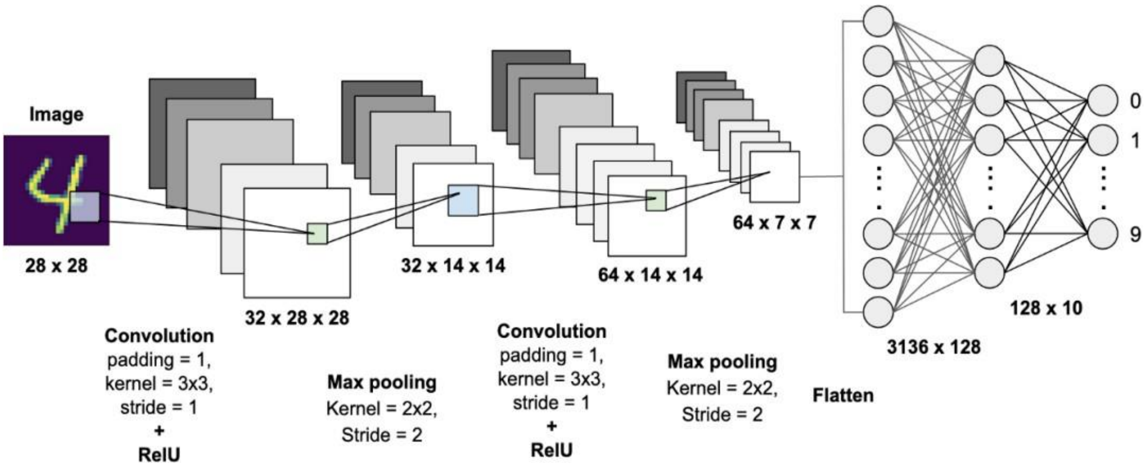
5. PROJECT DESIGN

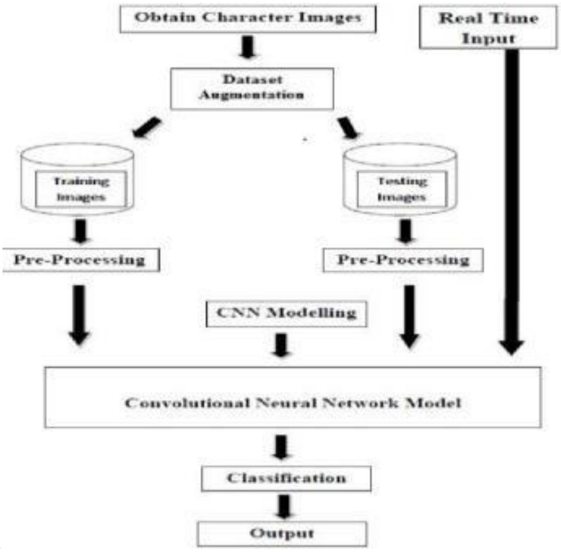
5.1 Data Flow Diagrams

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored. DFD Level 0 (Industry Standard)

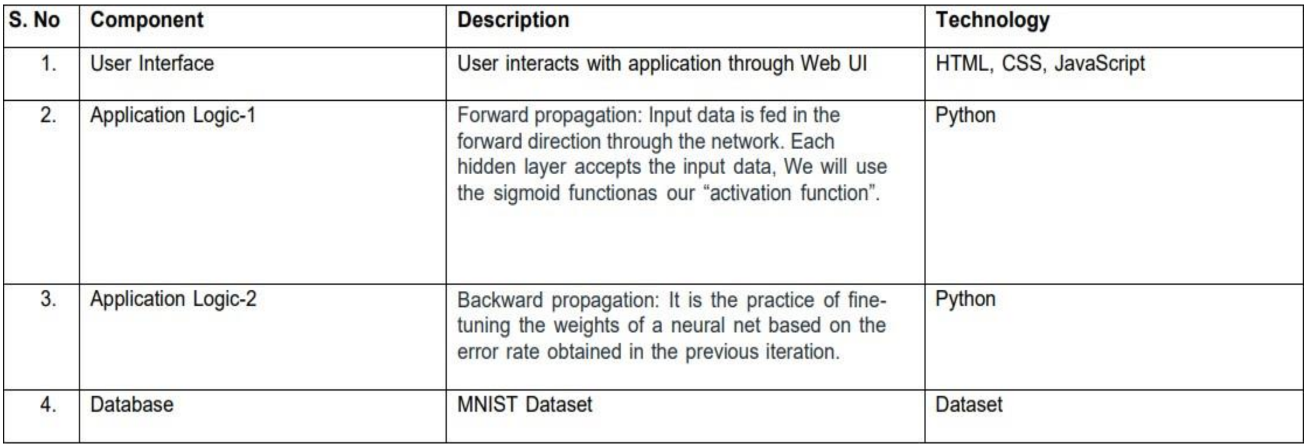


5.2 Solution & Technical Architecture



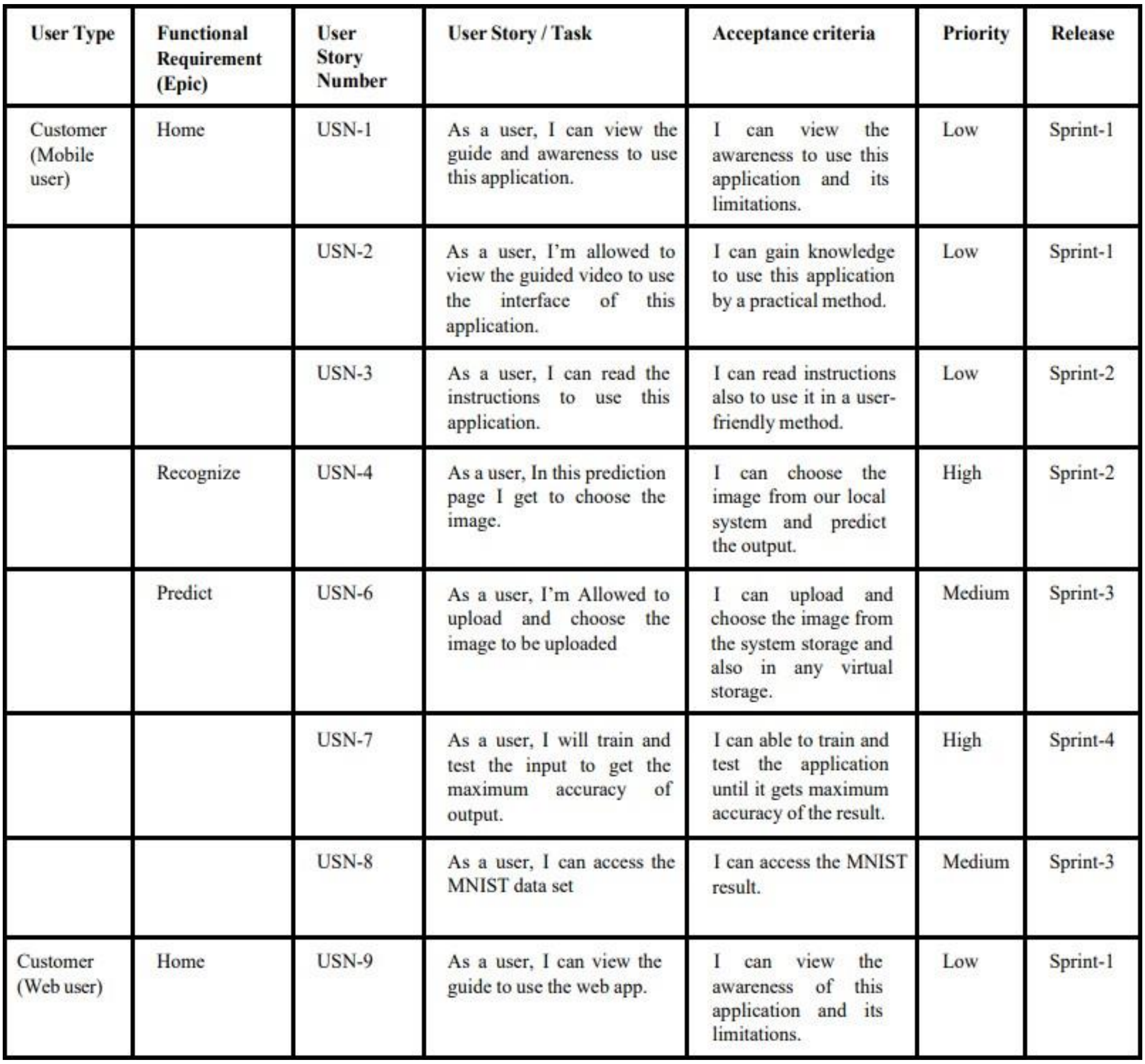


;



S. No	Component	Description	Technology
1.	User Interface	User interacts with application through Web UI	HTML, CSS, JavaScript
2.	Application Logic-1	Forward propagation: Input data is fed in the forward direction through the network. Each hidden layer accepts the input data, We will use the sigmoid functionas our “activation function”.	Python
3.	Application Logic-2	Backward propagation: It is the practice of fine-tuning the weights of a neural net based on the error rate obtained in the previous iteration.	Python
4.	Database	MNIST Dataset	Dataset

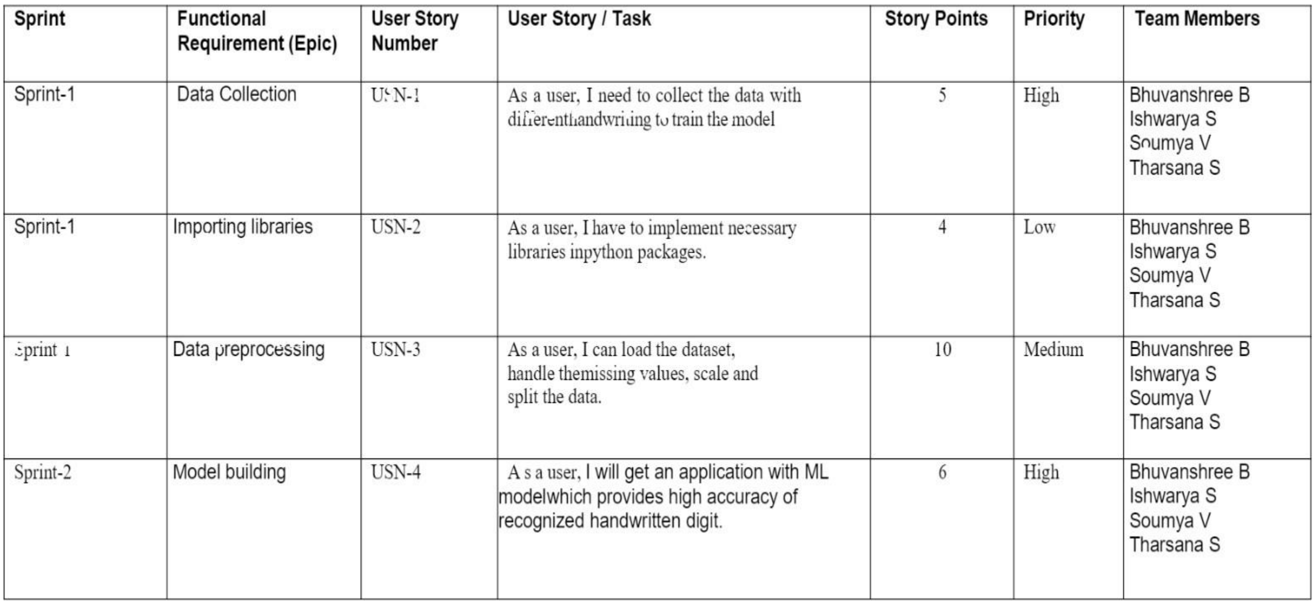
5.3 User Stories



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 awareness to use this application and its limitations.	Low	Sprint-1
		USN-2	As a user, I’m allowed to view the guided video to use the interface of this application.	I can gain knowledge to use this application by a practical method.	Low	Sprint-1
		USN-3	As a user, I can read the instructions to use this application.	I can read instructions also to use it in a user-friendly method.	Low	Sprint-2
	Recognize	USN-4	As a user, In this prediction page I get to choose the image.	I can choose the image from our local system and predict the output.	High	Sprint-2
	Predict	USN-6	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
		USN-7	As a user, I will train and test the input to get the maximum accuracy of output.	I can able to train and test the application until it gets maximum accuracy of the result.	High	Sprint-4
		USN-8	As a user, I can access the MNIST data set	I can access the MNIST result.	Medium	Sprint-3
Customer (Web user)	Home	USN-9	As a user, I can view the guide to use the web app.	I can view the awareness of this application and its limitations.	Low	Sprint-1

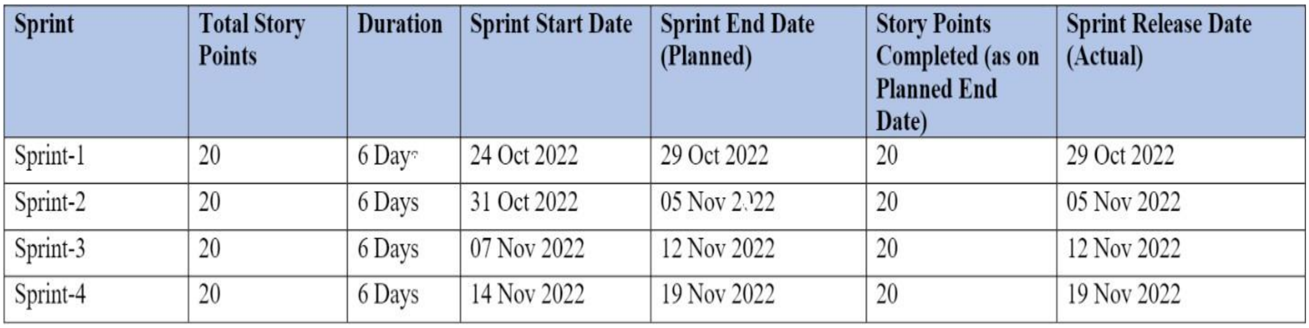
6. PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNING AND ESTIMATION



Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data Collection	USN-1	As a user, I need to collect the data with differenthandwriting to train the model	5	High	Bhuvanshree B Ishwarya S Soumya V Tharsana S
Sprint-1	Importing libraries	USN-2	As a user, I have to implement necessary libraries inpython packages.	4	Low	Bhuvanshree B Ishwarya S Soumya V Tharsana S
Sprint 1	Data preprocessing	USN-3	As a user, I can load the dataset, handle themissing values, scale and split the data.	10	Medium	Bhuvanshree B Ishwarya S Soumya V Tharsana S
Sprint-2	Model building	USN-4	A s a user, I will get an application with ML modelwhich provides high accuracy of recognized handwritten digit.	6	High	Bhuvanshree B Ishwarya S Soumya V Tharsana S

6.2 SPRINT DELIVERY SCHEDULE

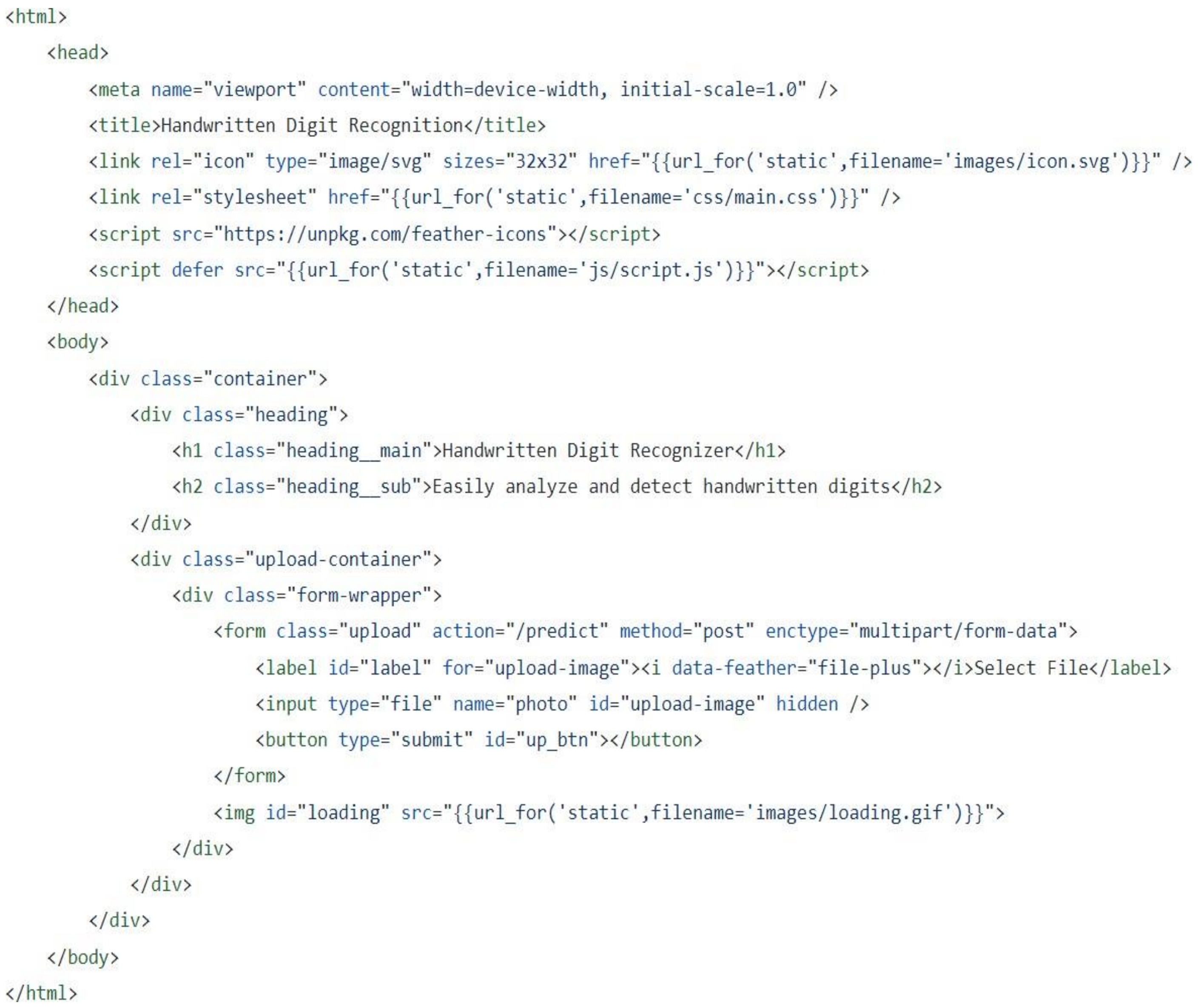


Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022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

7. CODING & SOLUTIONING

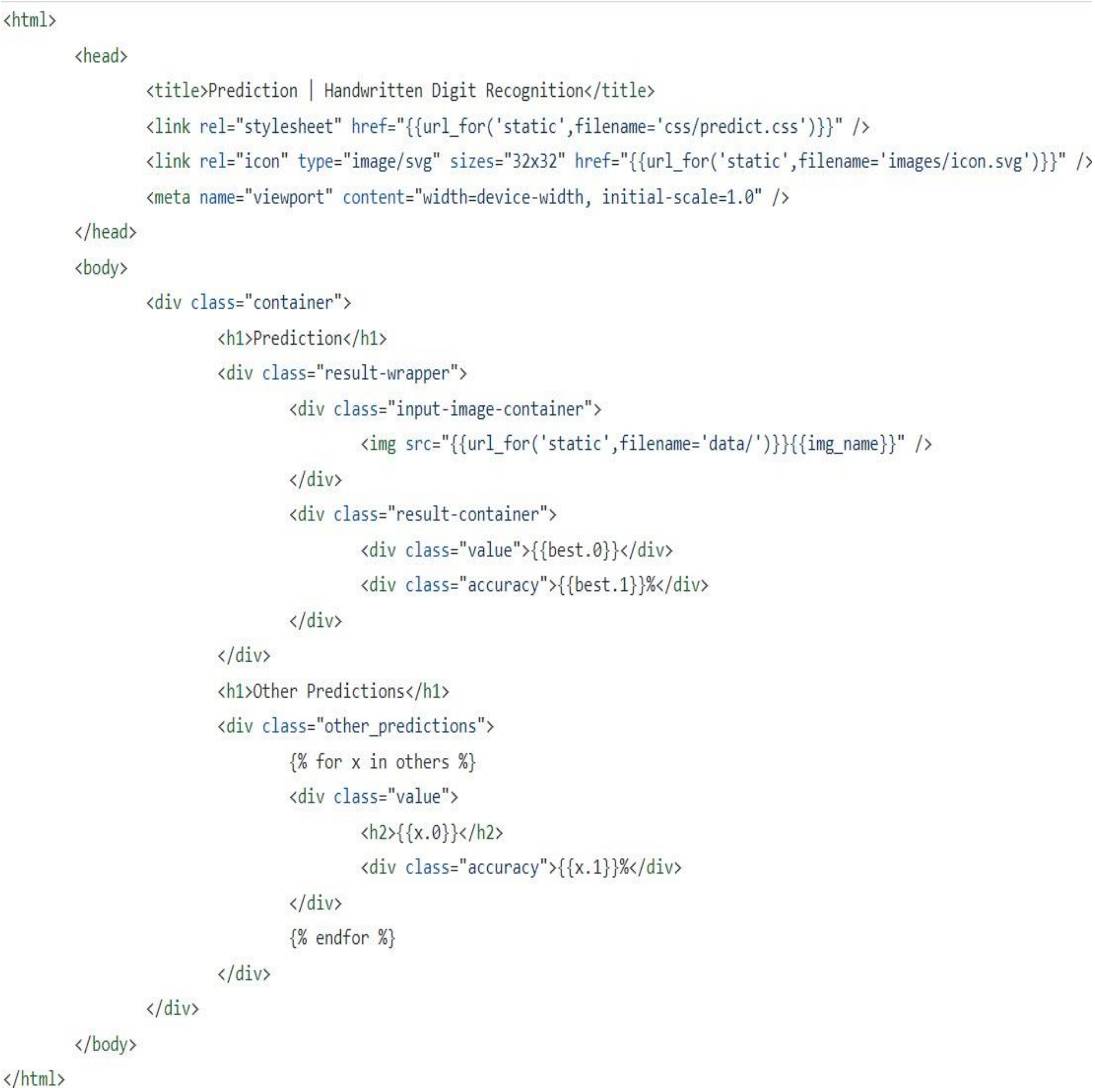
Front end

Home



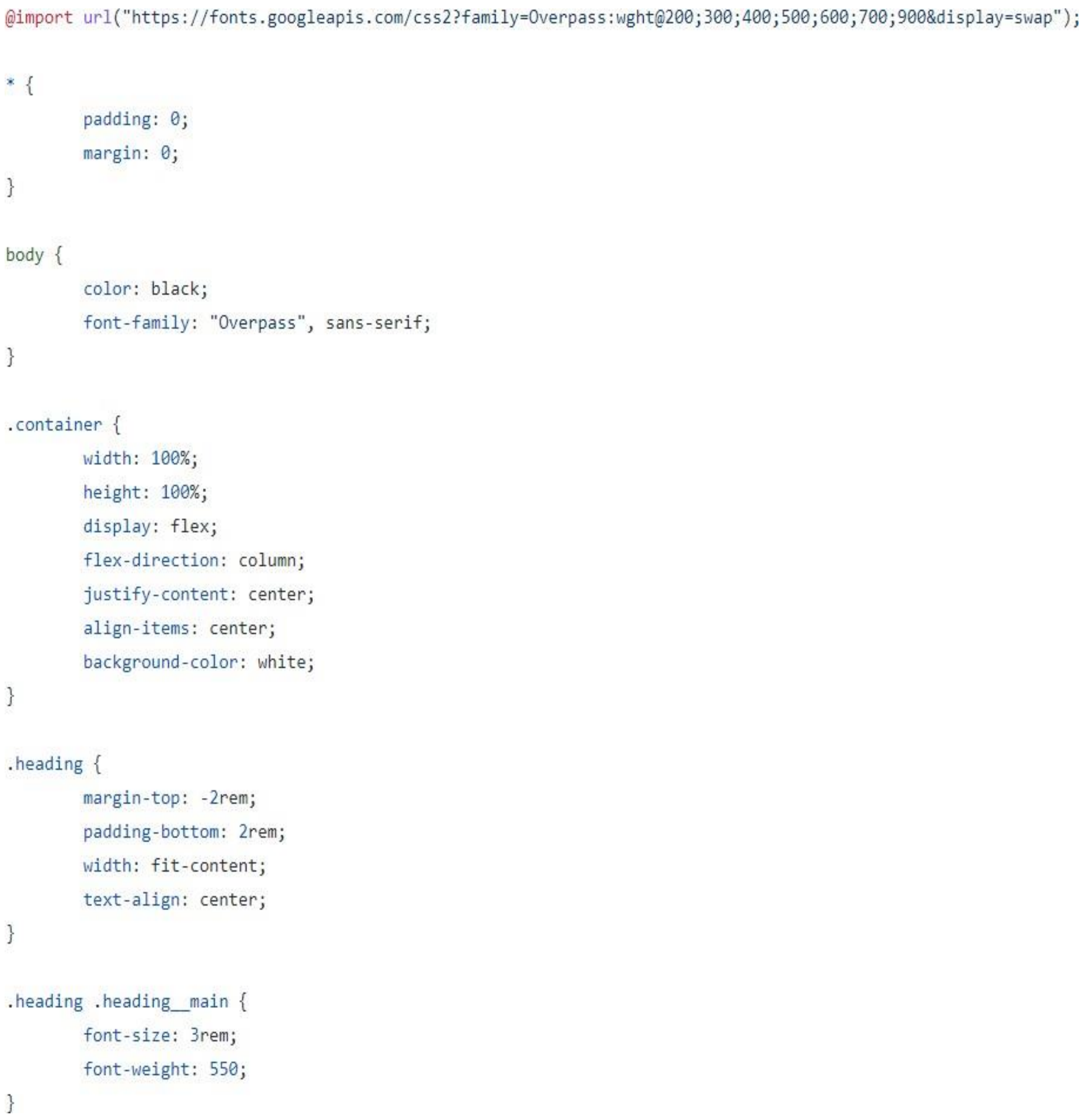
```html
<html>
    <head>
        <meta name="viewport" content="width=device-width, initial-scale=1.0" />
        <title>Handwritten Digit Recognition</title>
        <link rel="icon" type="image/svg" sizes="32x32" href="{{url_for('static',filename='images/icon.svg')}}" />
        <link rel="stylesheet" href="{{url_for('static',filename='css/main.css')}}" />
        <script src="https://unpkg.com/feather-icons"></script>
        <script defer src="{{url_for('static',filename='js/script.js')}}"></script>
    </head>
    <body>
        <div class="container">
            <div class="heading">
                <h1 class="heading__main">Handwritten Digit Recognizer</h1>
                <h2 class="heading__sub">Easily analyze and detect handwritten digits</h2>
            </div>
            <div class="upload-container">
                <div class="form-wrapper">
                    <form class="upload" action="/predict" method="post" enctype="multipart/form-data">
                        <label id="label" for="upload-image"><i data-feather="file-plus"></i>Select File</label>
                        <input type="file" name="photo" id="upload-image" hidden />
                        <button type="submit" id="up_btn"></button>
                    </form>
                    <img id="loading" src="{{url_for('static',filename='images/loading.gif')}}">
                </div>
            </div>
        </div>
    </body>
</html>
```

Predict



```
<html>
        <head>
                <title>Prediction | Handwritten Digit Recognition</title>
                <link rel="stylesheet" href="{{url_for('static',filename='css/predict.css')}}" />
                <link rel="icon" type="image/svg" sizes="32x32" href="{{url_for('static',filename='images/icon.svg')}}" />
                <meta name="viewport" content="width=device-width, initial-scale=1.0" />
        </head>
        <body>
                <div class="container">
                        <h1>Prediction</h1>
                        <div class="result-wrapper">
                                <div class="input-image-container">
                                        <img src="{{url_for('static',filename='data/')}}{{img_name}}" />
                                </div>
                                <div class="result-container">
                                        <div class="value">{{best.0}}</div>
                                        <div class="accuracy">{{best.1}}%</div>
                                </div>
                        </div>
                        <h1>Other Predictions</h1>
                        <div class="other_predictions">
                                {% for x in others %}
                                <div class="value">
                                        <h2>{{x.0}}</h2>
                                        <div class="accuracy">{{x.1}}%</div>
                                </div>
                                {% endfor %}
                        </div>
                </div>
        </body>
</html>
```

Main

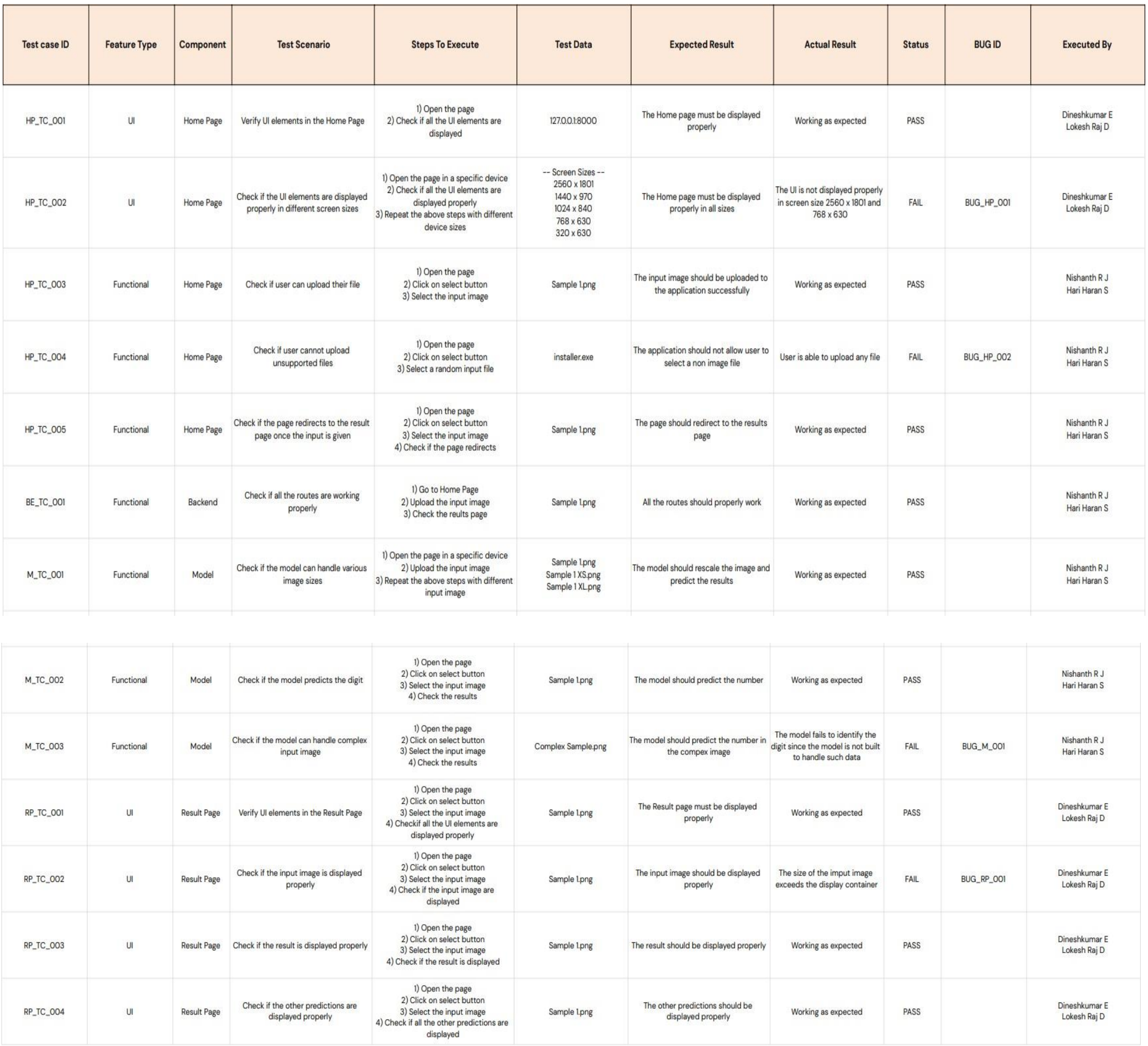


```
@import url("https://fonts.googleapis.com/css2?family=Overpass:wght@200;300;400;500;600;700;900&display=swap");

* {
        padding: 0;
        margin: 0;
}

body {
        color: black;
        font-family: "Overpass", sans-serif;
}

.container {
        width: 100%;
        height: 100%;
        display: flex;
        flex-direction: column;
        justify-content: center;
        align-items: center;
        background-color: white;
}

.heading {
        margin-top: -2rem;
        padding-bottom: 2rem;
        width: fit-content;
        text-align: center;
}

.heading .heading__main {
        font-size: 3rem;
        font-weight: 550;
}
```

```
.heading .heading__sub {
        font-size: 1rem;
        color: rgb(90, 88, 88);
}

.upload-container {
        box-shadow: 0 0 20px rgb(172, 170, 170);
        width: 40rem;
        height: 25rem;
        padding: 1.5rem;
}

.form-wrapper {
        background-color: rgba(190, 190, 190, 0.5);
        width: 100%;
        height: 100%;
        display: flex;
        border: 1px dashed black;
        justify-content: center;
        align-items: center;
}

.form-wrapper #loading {
        display: none;
        position: absolute;
}

.form-wrapper .upload {
        display: flex;
        justify-content: center;
        align-items: center;
        width: 8rem;
        height: -webkit-fit-content;
        height: -moz-fit-content;
        height: fit-content;
        border-radius: 6px;
        color: white;
        background-color: rgb(114, 96, 182);
        box-shadow: 0 5px 10px rgb(146, 135, 247);
}

.form-wrapper .upload #up_btn {
        display: none;
}

.form-wrapper .upload label {
        font-size: 1rem;
        font-weight: 600;
        color: white;
        height: 100%;
        width: 100%;
        padding: 10px;
        display: block;
}

.form-wrapper .upload svg {
        height: 15px;
        width: auto;
        padding-right: 8px;
        margin-bottom: -2px;
}

@media screen and (max-width: 700px) {
        .upload-container {
                height: 20rem;
                width: 18rem;
                margin-top: 3.5rem;
                margin-bottom: -8rem;
        }

        .heading .heading__main {
                margin-top: -6rem;
                font-size: 2rem;
                padding-bottom: 1rem;
        }
}
```

8. TESTING

8.1 TEST CASES

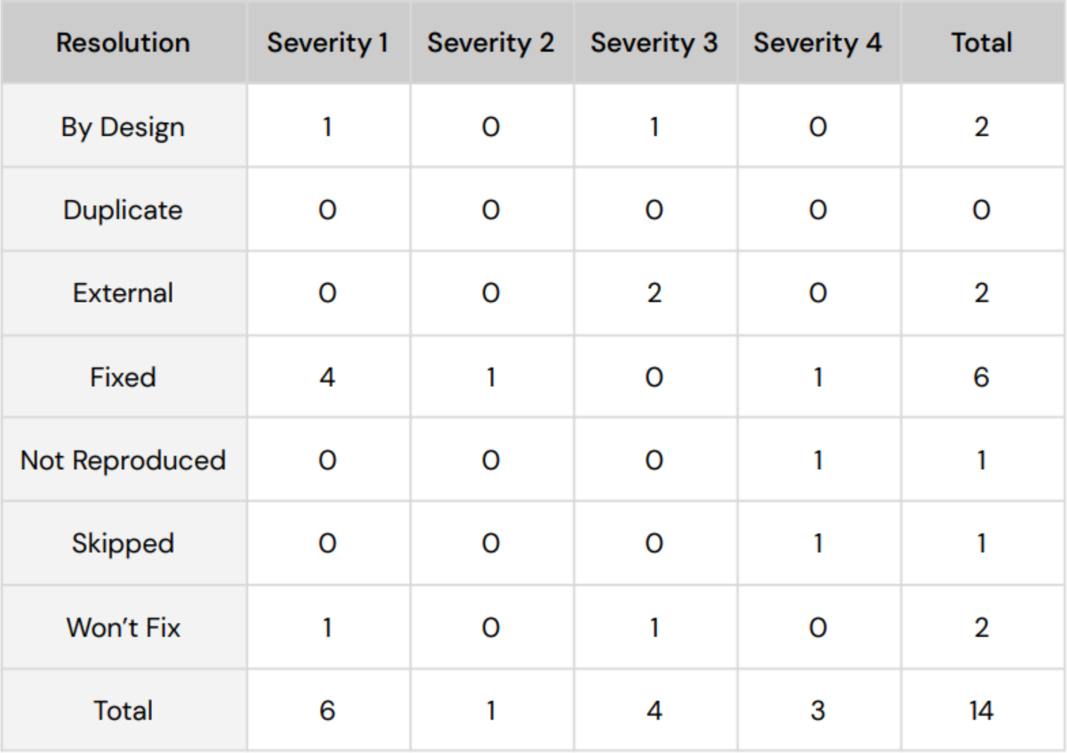


Test case ID	Feature Type	Component	Test Scenario	Steps To Execute	Test Data	Expected Result	Actual Result	Status	BUG ID	Executed By
HP_TC_001	UI	Home Page	Verify UI elements in the Home Page	1) Open the page 2) Check if all the UI elements are displayed	127.0.0.1:8000	The Home page must be displayed properly	Working as expected	PASS		Dineshkumar E Lokesh Raj D
HP_TC_002	UI	Home Page	Check if the UI elements are displayed properly in different screen sizes	1) Open the page in a specific device 2) Check if all the UI elements are displayed properly 3) Repeat the above steps with different device sizes	-- Screen Sizes -- 2560 x 1801 1440 x 970 1024 x 840 768 x 630 320 x 630	The Home page must be displayed properly in all sizes	The UI is not displayed properly in screen size 2560 x 1801 and 768 x 630	FAIL	BUG_HP_001	Dineshkumar E Lokesh Raj D
HP_TC_003	Functional	Home Page	Check if user can upload their file	1) Open the page 2) Click on select button 3) Select the input image	Sample 1.png	The input image should be uploaded to the application successfully	Working as expected	PASS		Nishanth R J Hari Haran S
HP_TC_004	Functional	Home Page	Check if user cannot upload unsupported files	1) Open the page 2) Click on select button 3) Select a random input file	installer.exe	The application should not allow user to select a non image file	User is able to upload any file	FAIL	BUG_HP_002	Nishanth R J Hari Haran S
HP_TC_005	Functional	Home Page	Check if the page redirects to the result page once the input is given	1) Open the page 2) Click on select button 3) Select the input image 4) Check if the page redirects	Sample 1.png	The page should redirect to the results page	Working as expected	PASS		Nishanth R J Hari Haran S
BE_TC_001	Functional	Backend	Check if all the routes are working properly	1) Go to Home Page 2) Upload the input image 3) Check the reults page	Sample 1.png	All the routes should properly work	Working as expected	PASS		Nishanth R J Hari Haran S
M_TC_001	Functional	Model	Check if the model can handle various image sizes	1) Open the page in a specific device 2) Upload the input image 3) Repeat the above steps with different input image	Sample 1.png Sample 1 XS.png Sample 1 XL.png	The model should rescale the image and predict the results	Working as expected	PASS		Nishanth R J Hari Haran S
M_TC_002	Functional	Model	Check if the model predicts the digit	1) Open the page 2) Click on select button 3) Select the input image 4) Check the results	Sample 1.png	The model should predict the number	Working as expected	PASS		Nishanth R J Hari Haran S
M_TC_003	Functional	Model	Check if the model can handle complex input image	1) Open the page 2) Click on select button 3) Select the input image 4) Check the results	Complex Sample.png	The model should predict the number in the compex image	The model fails to identify the digit since the model is not built to handle such data	FAIL	BUG_M_001	Nishanth R J Hari Haran S
RP_TC_001	UI	Result Page	Verify UI elements in the Result Page	1) Open the page 2) Click on select button 3) Select the input image 4) Checkif all the UI elements are displayed properly	Sample 1.png	The Result page must be displayed properly	Working as expected	PASS		Dineshkumar E Lokesh Raj D
RP_TC_002	UI	Result Page	Check if the input image is displayed properly	1) Open the page 2) Click on select button 3) Select the input image 4) Check if the input image are displayed	Sample 1.png	The input image should be displayed properly	The size of the imput image exceeds the display container	FAIL	BUG_RP_001	Dineshkumar E Lokesh Raj D
RP_TC_003	UI	Result Page	Check if the result is displayed properly	1) Open the page 2) Click on select button 3) Select the input image 4) Check if the result is displayed	Sample 1.png	The result should be displayed properly	Working as expected	PASS		Dineshkumar E Lokesh Raj D
RP_TC_004	UI	Result Page	Check if the other predictions are displayed properly	1) Open the page 2) Click on select button 3) Select the input image 4) Check if all the other predictions are displayed	Sample 1.png	The other predictions should be displayed properly	Working as expected	PASS		Dineshkumar E Lokesh Raj D

8.2 USER ACCEPTANCE TESTING

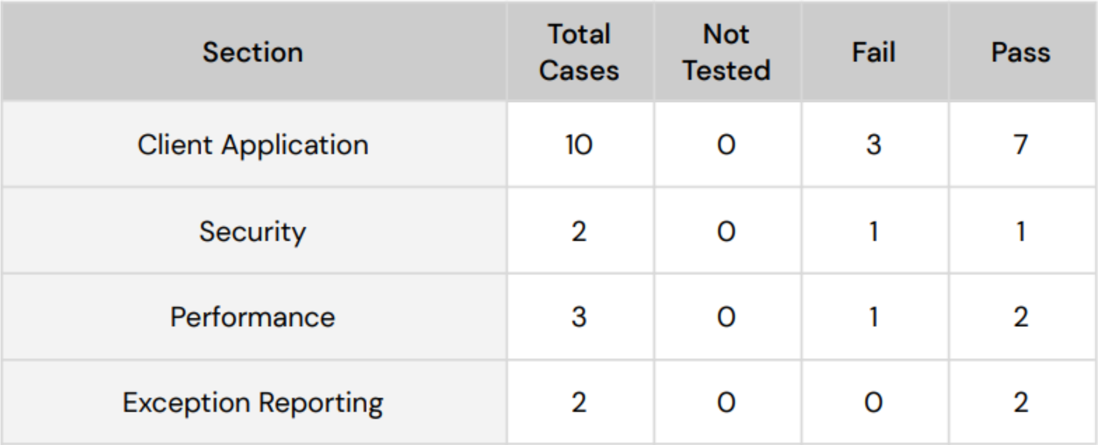
User Acceptance Testing (UAT) explains the test coverage and open issues of the Handwritten Digit Recognition project at the time of the release to UAT

8.2.1 DEFECT ANALYSIS



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

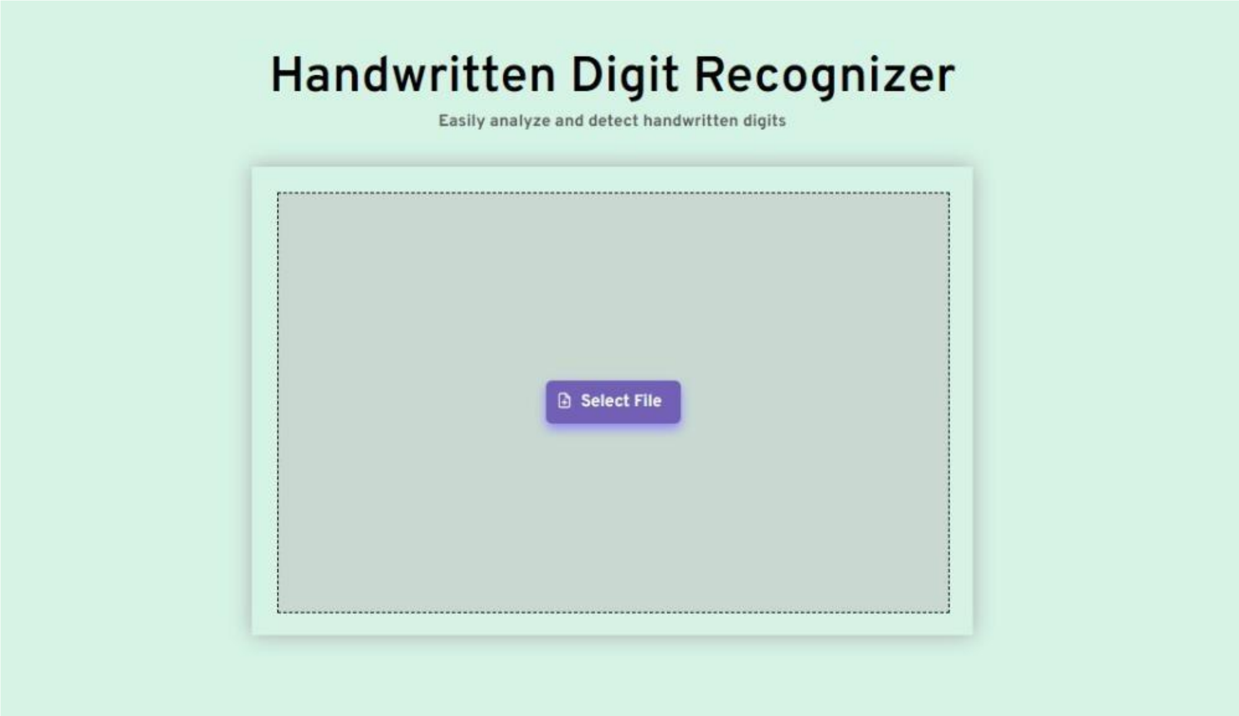
8.2.2 TEST CASE ANALYSIS



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

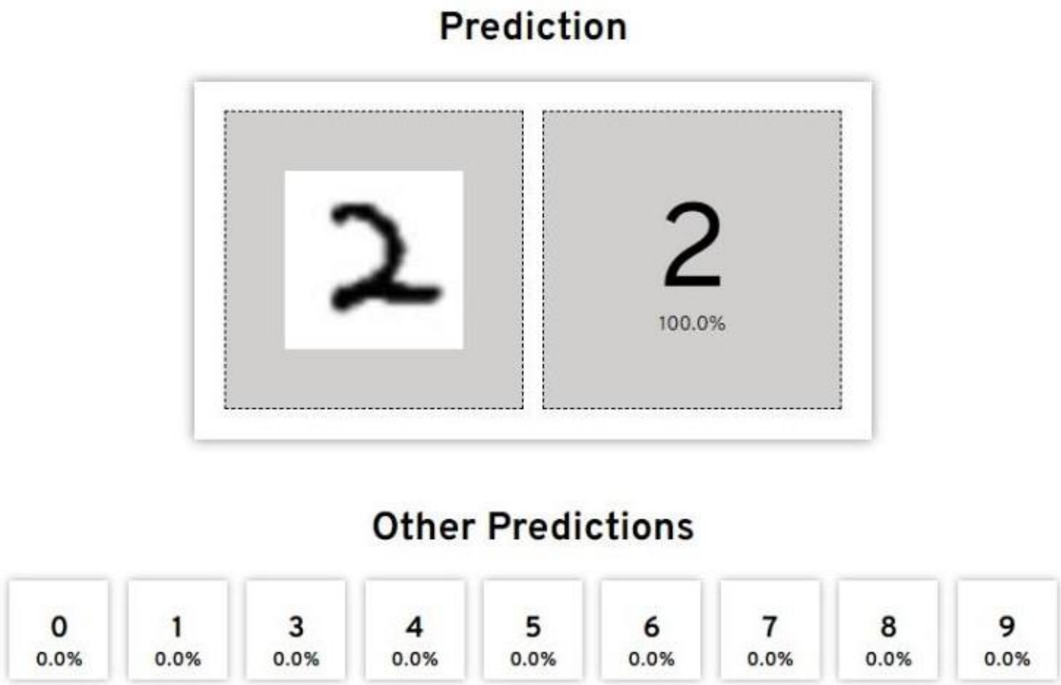
9. RESULTS

9.1 PERFORMANCE METRICS



Handwritten Digit Recognizer

Easily analyze and detect handwritten digits



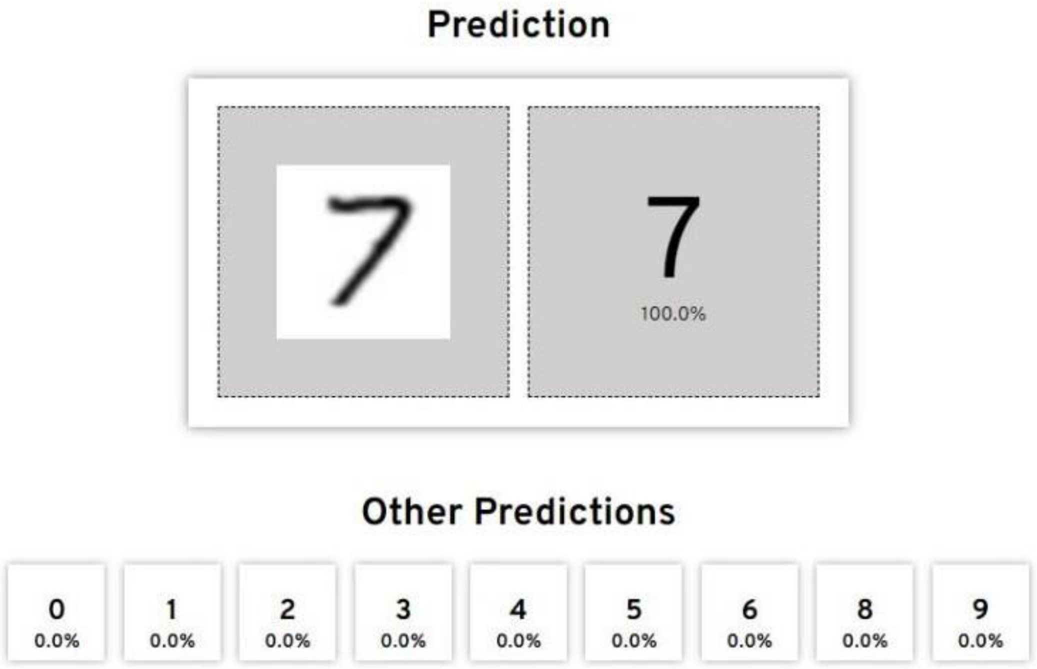
Prediction

2

100.0%

Other Predictions

0	1	3	4	5	6	7	8	9
0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%



Prediction

7

100.0%

Other Predictions

0	1	2	3	4	5	6	8	9
0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%

10. ADVANTAGES & DISADVANTAGES

- 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

11. CONCLUSION

This project demonstrated a web application that uses machine learning to recognise handwritten numbers. Flask, HTML, CSS, JavaScript, and a few other technologies were used to create this project. The model predicts the handwritten digit using a CNN network. During testing, the model achieved a 99.61% recognition rate. The proposed project is scalable and can easily handle a huge number of users. Since it is a web application, it is compatible with any device that can run a browser. This project is extremely useful in real-world scenarios such as recognizing number plates of vehicles, processing bank cheque amounts, numeric entries in forms filled up by hand (tax forms) and so on. There is so much room for improvement, which can be implemented in subsequent versions.

12. FUTURE SCOPE

Some of the improvements that can be made to this project are as follows:

- Add support to detect from digits multiple images and save the results
- Add support to detect multiple digits
- Improve model to detect digits from complex images
- Add support to different languages to help users from all over the world

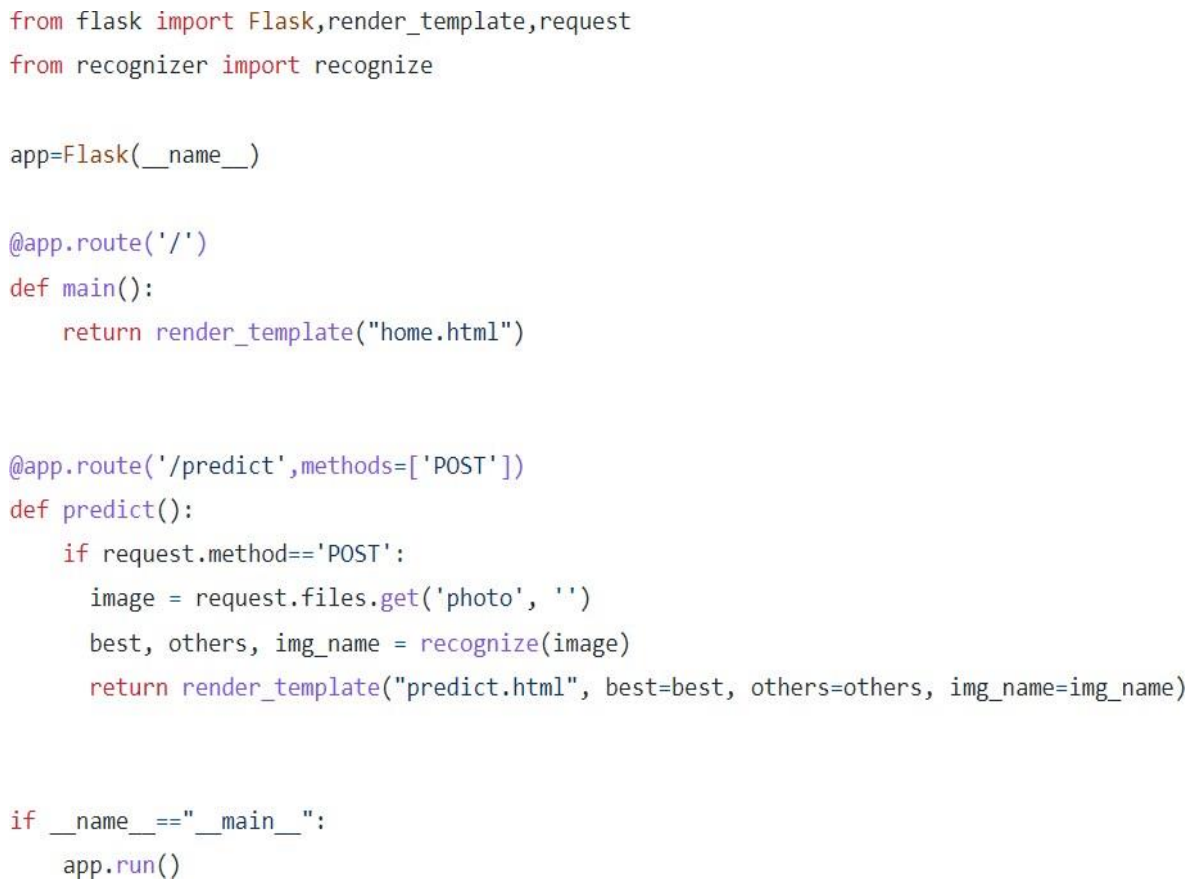
The task of handwritten digit recognition, using a classifier, has great importance and use such as online handwriting recognition on computer tablets, recognize zip codes on mail for postal mail sorting, processing bank check amounts, numeric entries in forms filled up by hand (for example - tax forms)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

APPENDIX

SOURCE CODE

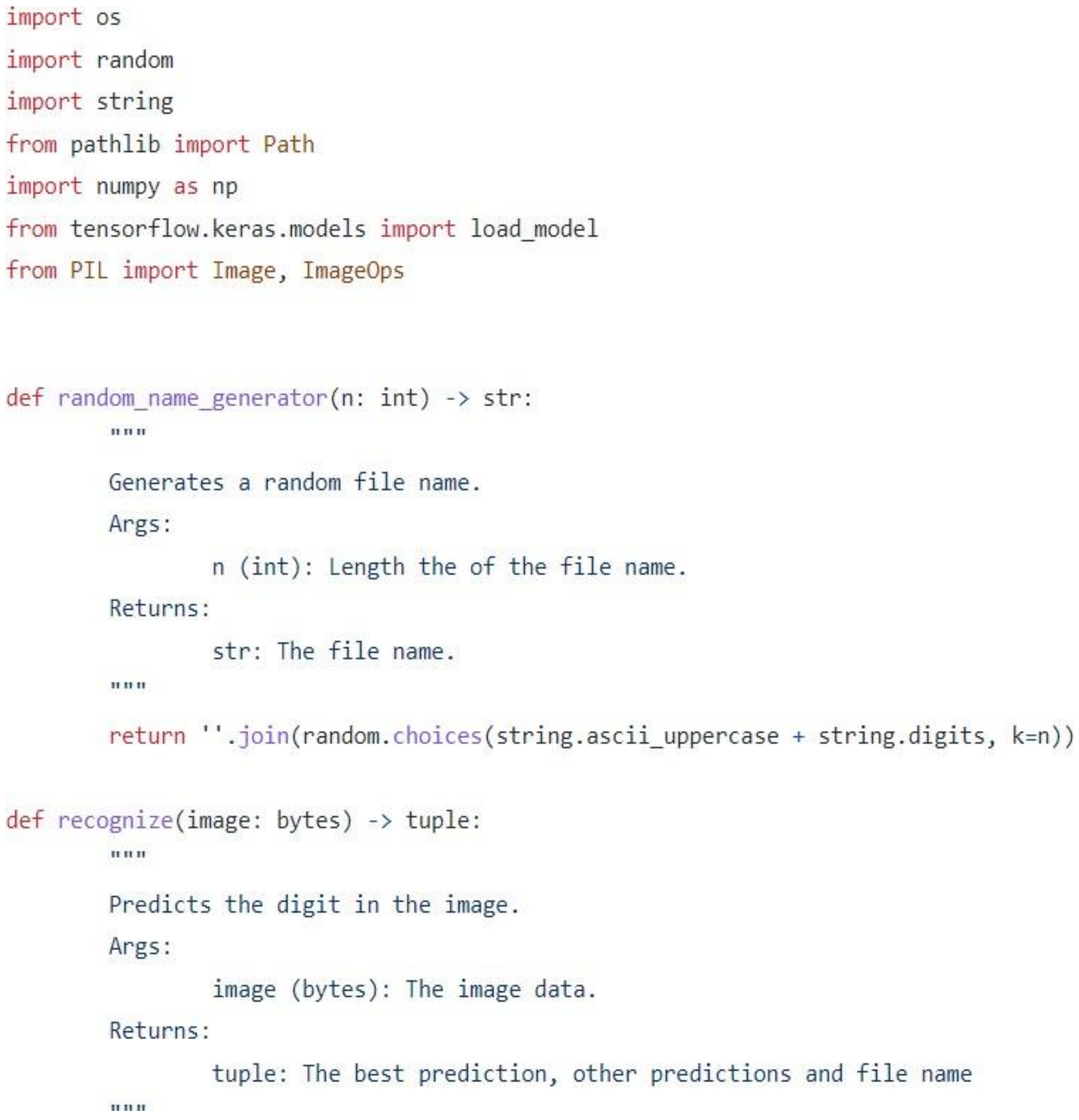
Back end

App

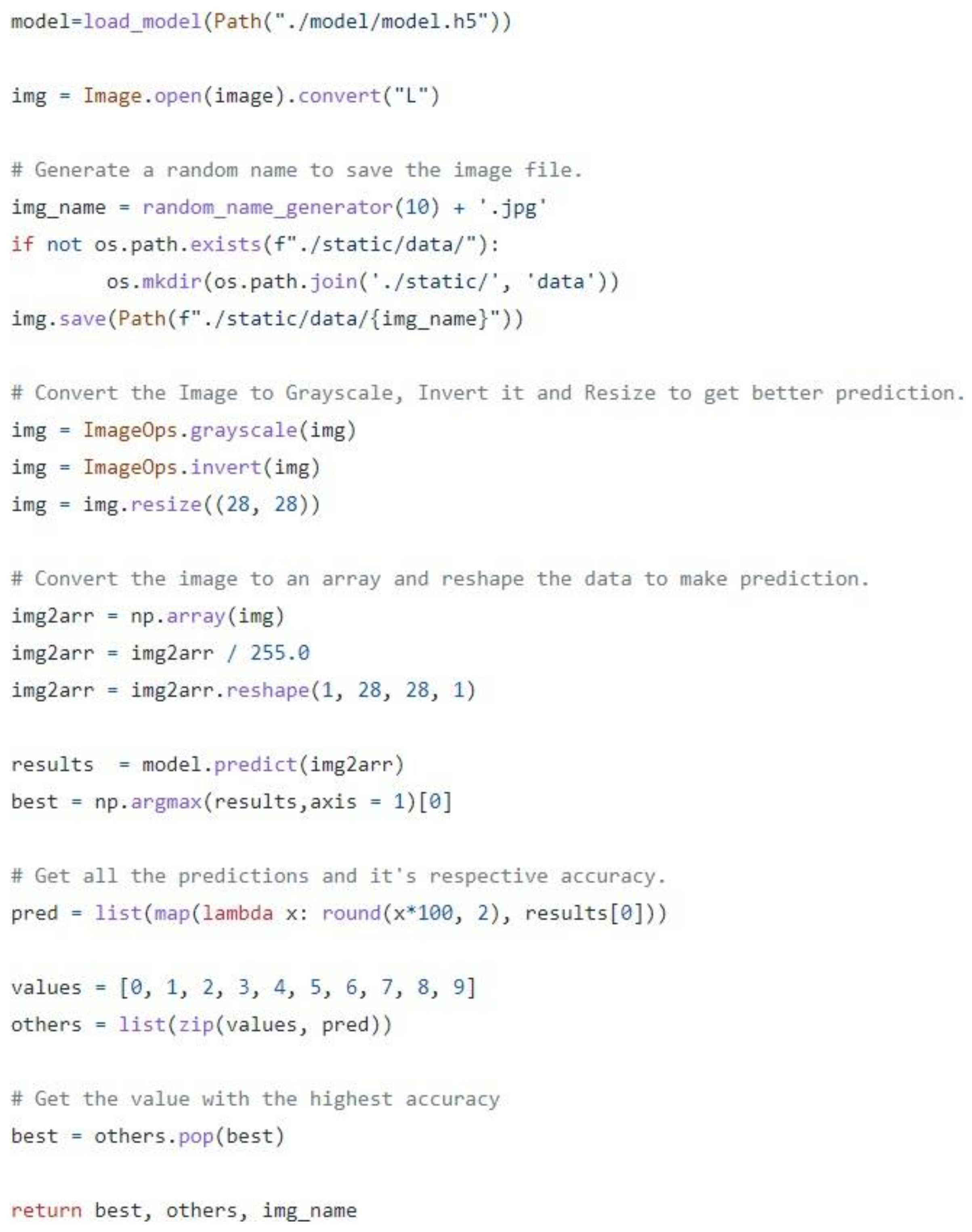


```
from flask import Flask,render_template,request
from recognizer import recognize

app=Flask(__name__)

@app.route('/')
def main():
    return render_template("home.html")


@app.route('/predict',methods=['POST'])
def predict():
    if request.method=='POST':
      image = request.files.get('photo', '')
      best, others, img_name = recognize(image)
      return render_template("predict.html", best=best, others=others, img_name=img_name)


if __name__=="__main__":
    app.run()
```

Recognizer



```
import os
import random
import string
from pathlib import Path
import numpy as np
from tensorflow.keras.models import load_model
from PIL import Image, ImageOps



def random_name_generator(n: int) -> str:
        """
        Generates a random file name.
        Args:
                n (int): Length the of the file name.
        Returns:
                str: The file name.
        """
        return ''.join(random.choices(string.ascii_uppercase + string.digits, k=n))

def recognize(image: bytes) -> tuple:
        """
        Predicts the digit in the image.
        Args:
                image (bytes): The image data.
        Returns:
                tuple: The best prediction, other predictions and file name
        """
```



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

img = Image.open(image).convert("L")

# Generate a random name to save the image file.
img_name = random_name_generator(10) + '.jpg'
if not os.path.exists(f"./static/data/"):
        os.mkdir(os.path.join('./static/', 'data'))
img.save(Path(f"./static/data/{img_name}"))

# Convert the Image to Grayscale, Invert it and Resize to get better prediction.
img = ImageOps.grayscale(img)
img = ImageOps.invert(img)
img = img.resize((28, 28))

# Convert the image to an array and reshape the data to make prediction.
img2arr = np.array(img)
img2arr = img2arr / 255.0
img2arr = img2arr.reshape(1, 28, 28, 1)

results  = model.predict(img2arr)
best = np.argmax(results,axis = 1)[0]

# Get all the predictions and it's respective accuracy.
pred = list(map(lambda x: round(x*100, 2), results[0]))

values = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
others = list(zip(values, pred))

# Get the value with the highest accuracy
best = others.pop(best)

return best, others, img_name
```

GITHUB

https://github.com/IBM-EPBL/IBM-Project-10235-1659117866

PROJECT DEMO

https://github.com/IBM-EPBL/IBM-Project-10235-1659117866/blob/main/Final%20deliverables/Project%20Demo/Project_Demo.mp4