Project Report

A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

IBM-Project-11068-1659258981

TEAM ID: PNT2022TMID00205

Submitted by

UPPU ABHIRAM GANESH	**312319104179**
VASANTH MANI VASI V	**312319104182**
TULASI RAMA RAJU C	**312319104176**
SAMUDRALA SAI PRANEETH	**312319104138**
SANJAY KUMAR S	**312319104145**

INDEX

A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

1. INTRODUCTION

1.1 Project Overview

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

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

1.2 Purpose

Handwritten digits are not perfect and can be made with many different flavors. The handwritten digit recognition is the solution to this problem which uses the image of a digit and recognizes the digits present in the image.

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

2. LITERATURE SURVEY

2.1 Existing Problem

The fundamental problem with handwritten digit recognition is that handwritten digits do not always have the same size, width, orientation, and margins since they vary from person to person. Additionally, there would be issues with identifying the numbers because of similarities between numbers like 1 and 7, 5 and 6, 3 and 8, 2 and 7, etc. Finally, the individuality and variation of each individual's handwriting influence the structure and appearance of the digits.

2.2 References

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

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

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

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

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

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

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

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

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

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

- Pashine, Samay and Dixit, Ritik and Kushwah, Rishika

In this study, they developed three deep and machine learning-based models for handwritten digit recognition using MNIST datasets. To determine which model was the most accurate, they compared them based on their individual properties. Support vector machines are among the simplest classifiers, making them faster than other algorithms and providing the highest training accuracy rate in this situation. However, due to their simplicity, SVMs cannot categorize complicated and ambiguous images as accurately as MLP and CNN algorithms can. In their research, they discovered that CNN produced the most precise outcomes for handwritten digit recognition. This led them to the conclusion that CNN is the most effective solution for all types of prediction issues, including those using picture data. Next, by comparing the execution times of the algorithms, they determined that increasing the number of epochs without changing the configuration of the algorithm is pointless due to the limitation of a certain model, and they discovered that beyond a certain number of epochs, the model begins over-fitting the dataset and provides biased predictions.

2.3 Problem Statement Definition

The following are the limitations that computers encounter while attempting to detect handwritten numbers:

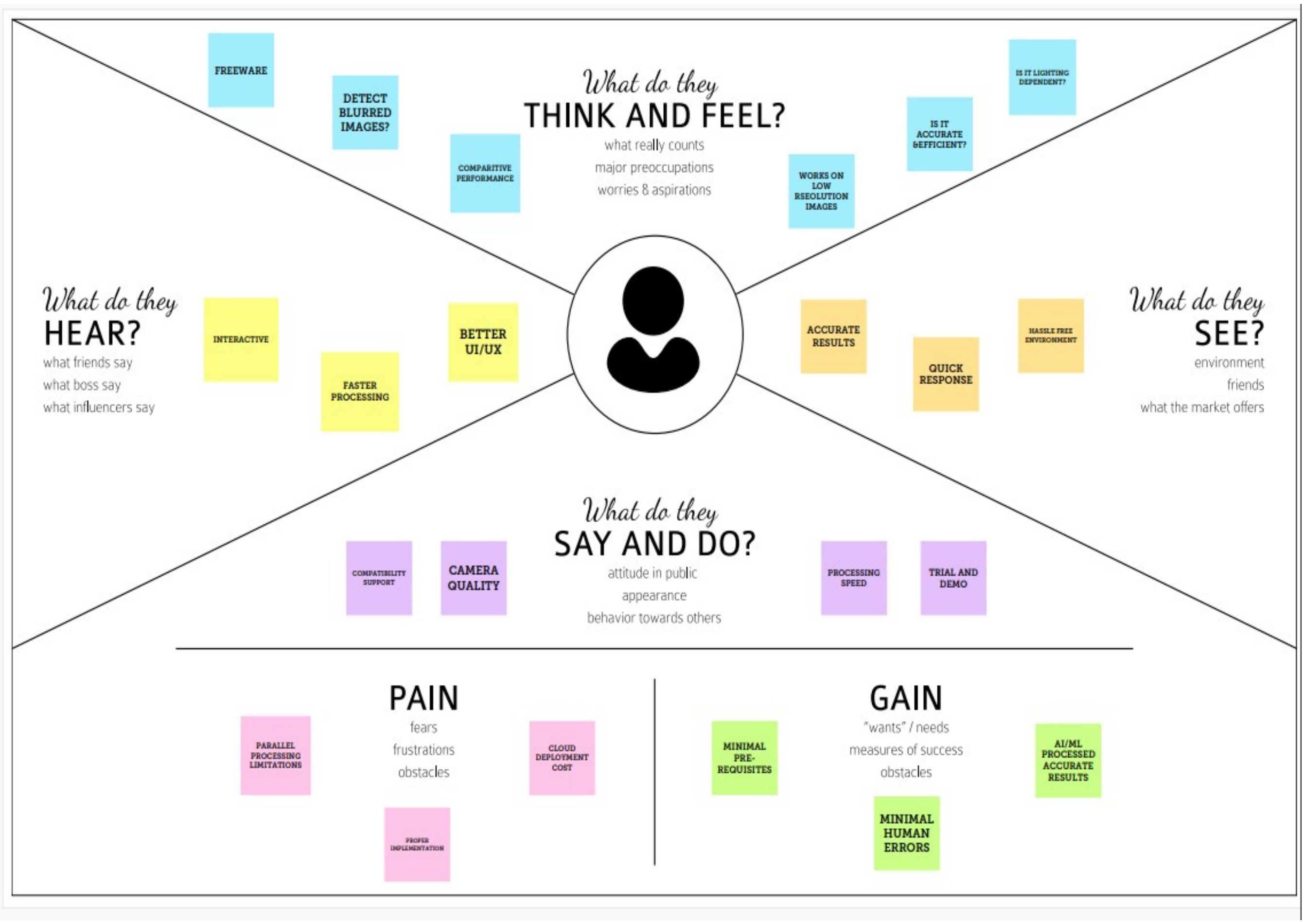
1. Because handwriting varies from person to person, handwritten numbers are not always the same size, width, and margins.

2. The likeness of digits like 1 and 7, which are justified and oriented to 5 and 6, 3 and 8, 2 and 7, etc. Therefore, it is a significant challenge for computers to classify between these values.

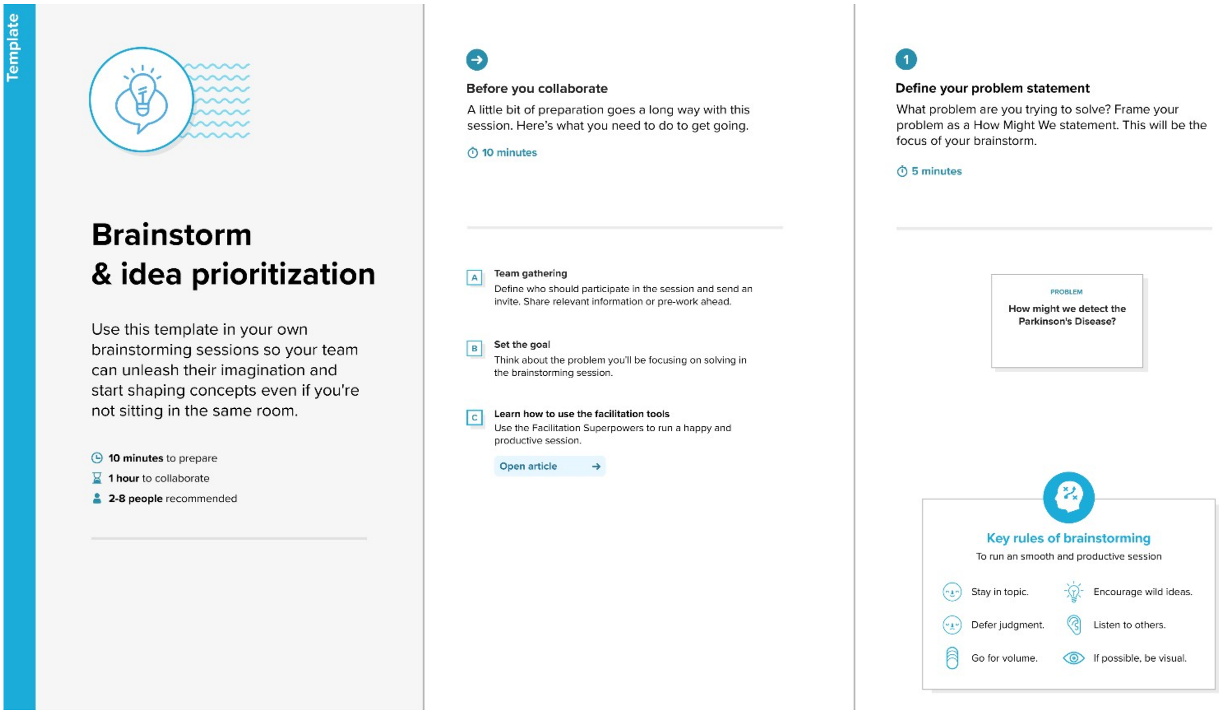
3. The uniqueness and variety of each person's handwriting also has an impact on the way the numerals form and appear.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming



Template

Brainstorm
& idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

10 minutes to prepare
1 hour to collaborate
2-8 people recommended

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

A **Team gathering**
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B **Set the goal**
Think about the problem you'll be focusing on solving in the brainstorming session.

C **Learn how to use the facilitation tools**
Use the Facilitation Superpowers to run a happy and productive session.

Open article →

1 **Define your problem statement**

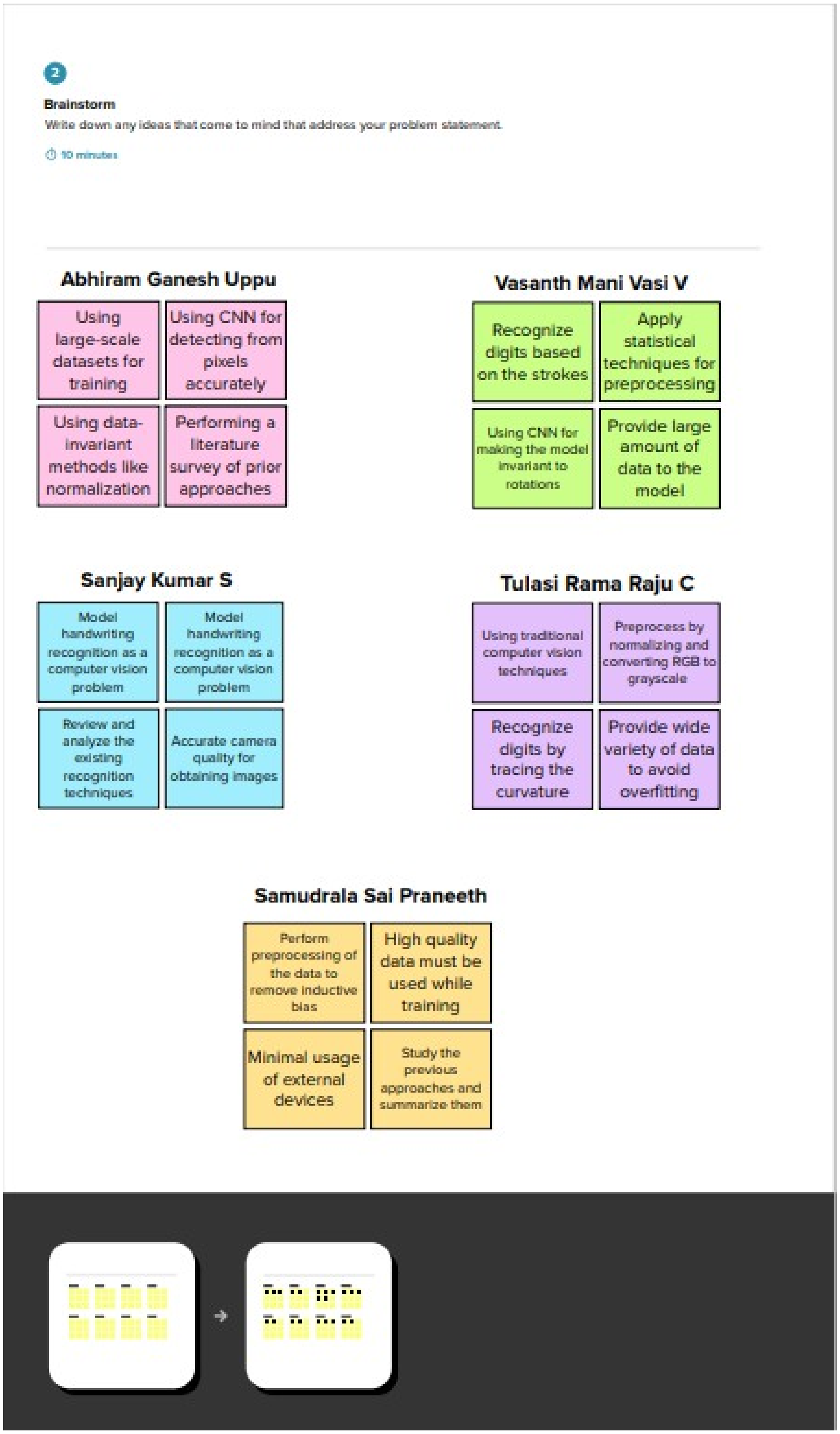
What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

5 minutes

PROBLEM
How might we detect the Parkinson's Disease?

Key rules of brainstorming
To run an smooth and productive session

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



2

Brainstorm

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

10 minutes

Abhiram Ganesh Uppu

- Using large-scale datasets for training
- Using CNN for detecting from pixels accurately
- Using data-invariant methods like normalization
- Performing a literature survey of prior approaches

Vasanth Mani Vasi V

- Recognize digits based on the strokes
- Apply statistical techniques for preprocessing
- Using CNN for making the model invariant to rotations
- Provide large amount of data to the model

Sanjay Kumar S

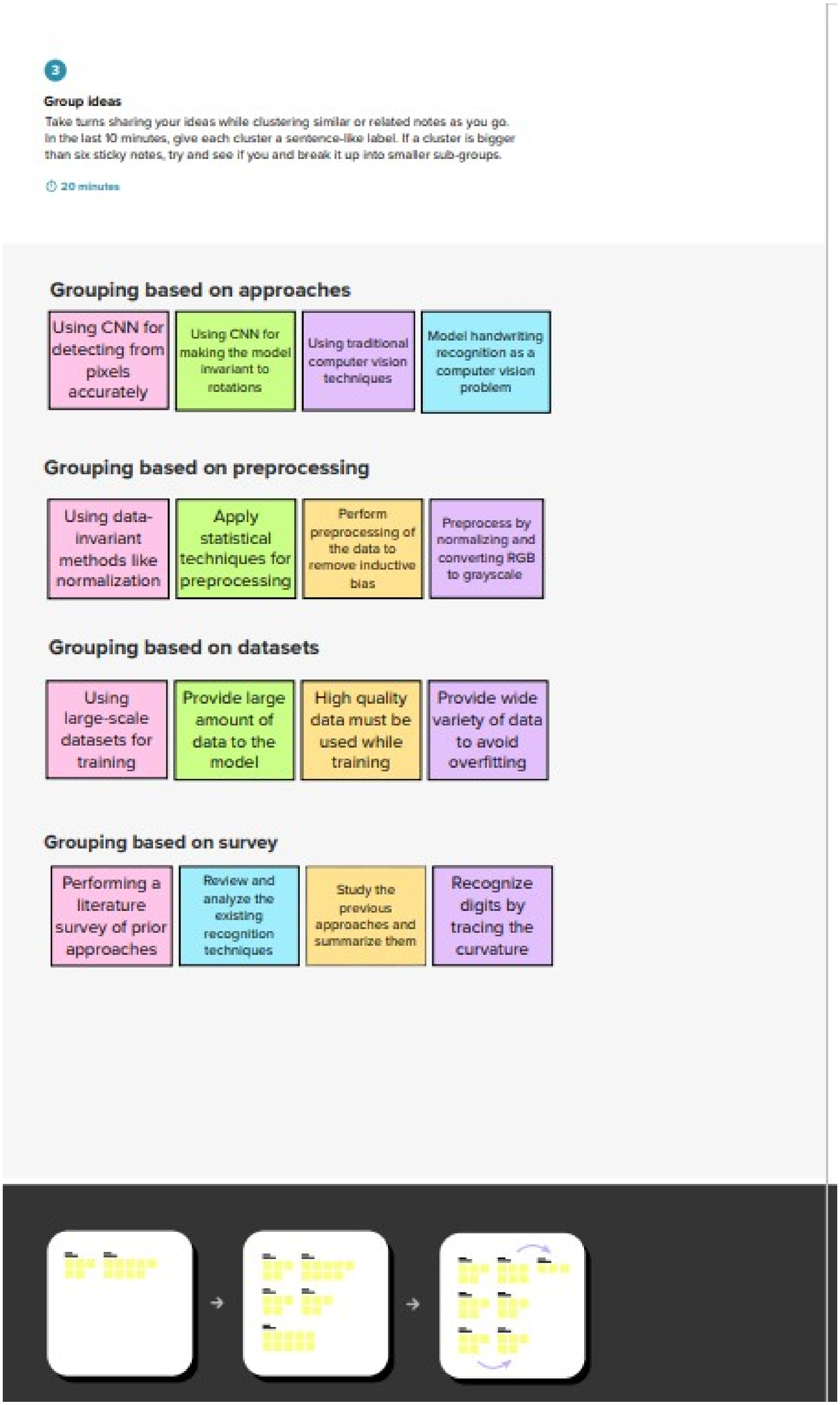
- Model handwriting recognition as a computer vision problem
- Model handwriting recognition as a computer vision problem
- Review and analyze the existing recognition techniques
- Accurate camera quality for obtaining images

Tulasi Rama Raju C

- Using traditional computer vision techniques
- Preprocess by normalizing and converting RGB to grayscale
- Recognize digits by tracing the curvature
- Provide wide variety of data to avoid overfitting

Samudrala Sai Praneeth

- Perform preprocessing of the data to remove inductive bias
- High quality data must be used while training
- Minimal usage of external devices
- Study the previous approaches and summarize them



3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10 minutes, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

Grouping based on approaches

- Using CNN for detecting from pixels accurately
- Using CNN for making the model invariant to rotations
- Using traditional computer vision techniques
- Model handwriting recognition as a computer vision problem

Grouping based on preprocessing

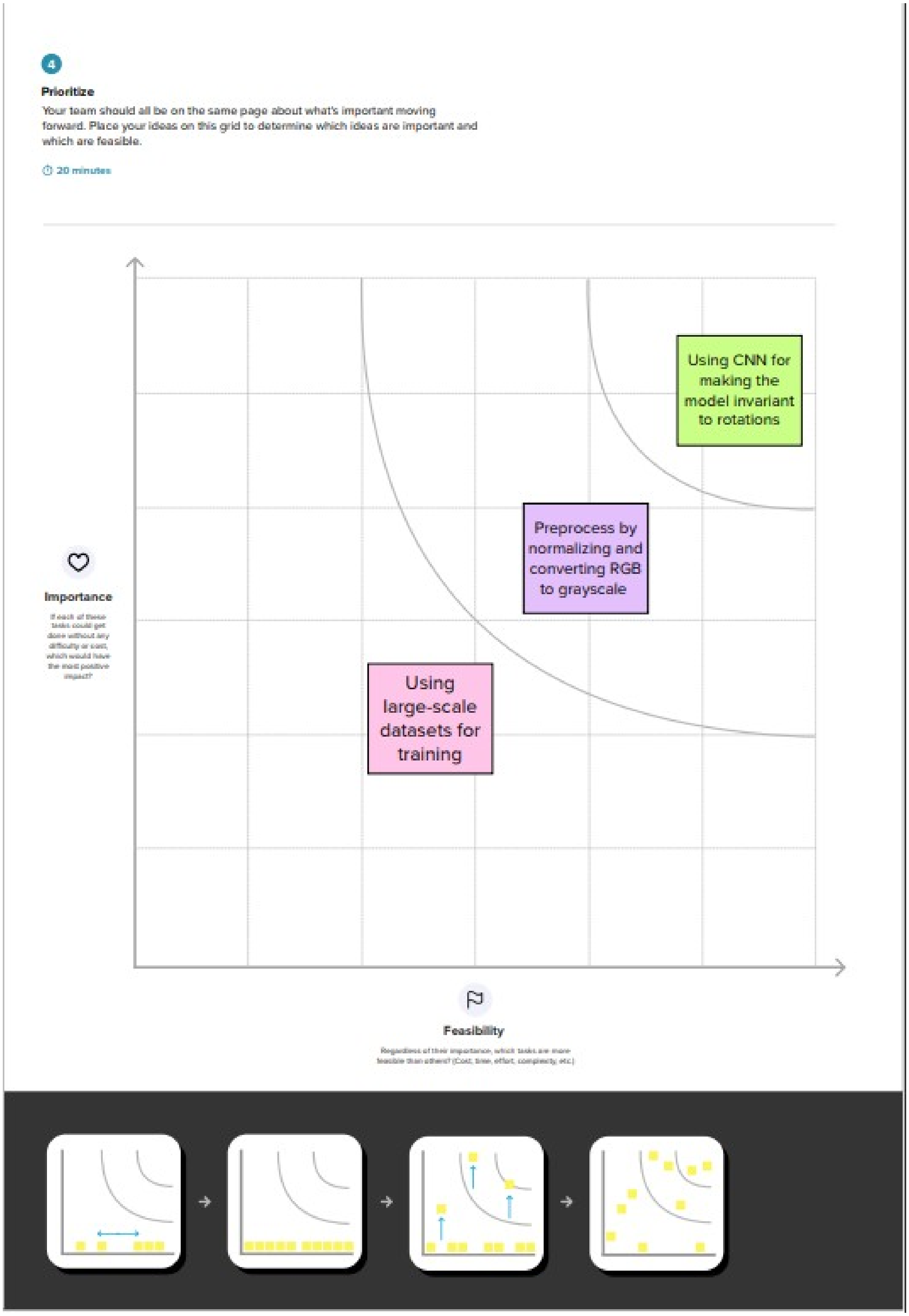
- Using data-invariant methods like normalization
- Apply statistical techniques for preprocessing
- Perform preprocessing of the data to remove inductive bias
- Preprocess by normalizing and converting RGB to grayscale

Grouping based on datasets

- Using large-scale datasets for training
- Provide large amount of data to the model
- High quality data must be used while training
- Provide wide variety of data to avoid overfitting

Grouping based on survey

- Performing a literature survey of prior approaches
- Review and analyze the existing recognition techniques
- Study the previous approaches and summarize them
- Recognize digits by tracing the curvature



4

Prioritize

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

20 minutes

Using CNN for making the model invariant to rotations

Preprocess by normalizing and converting RGB to grayscale

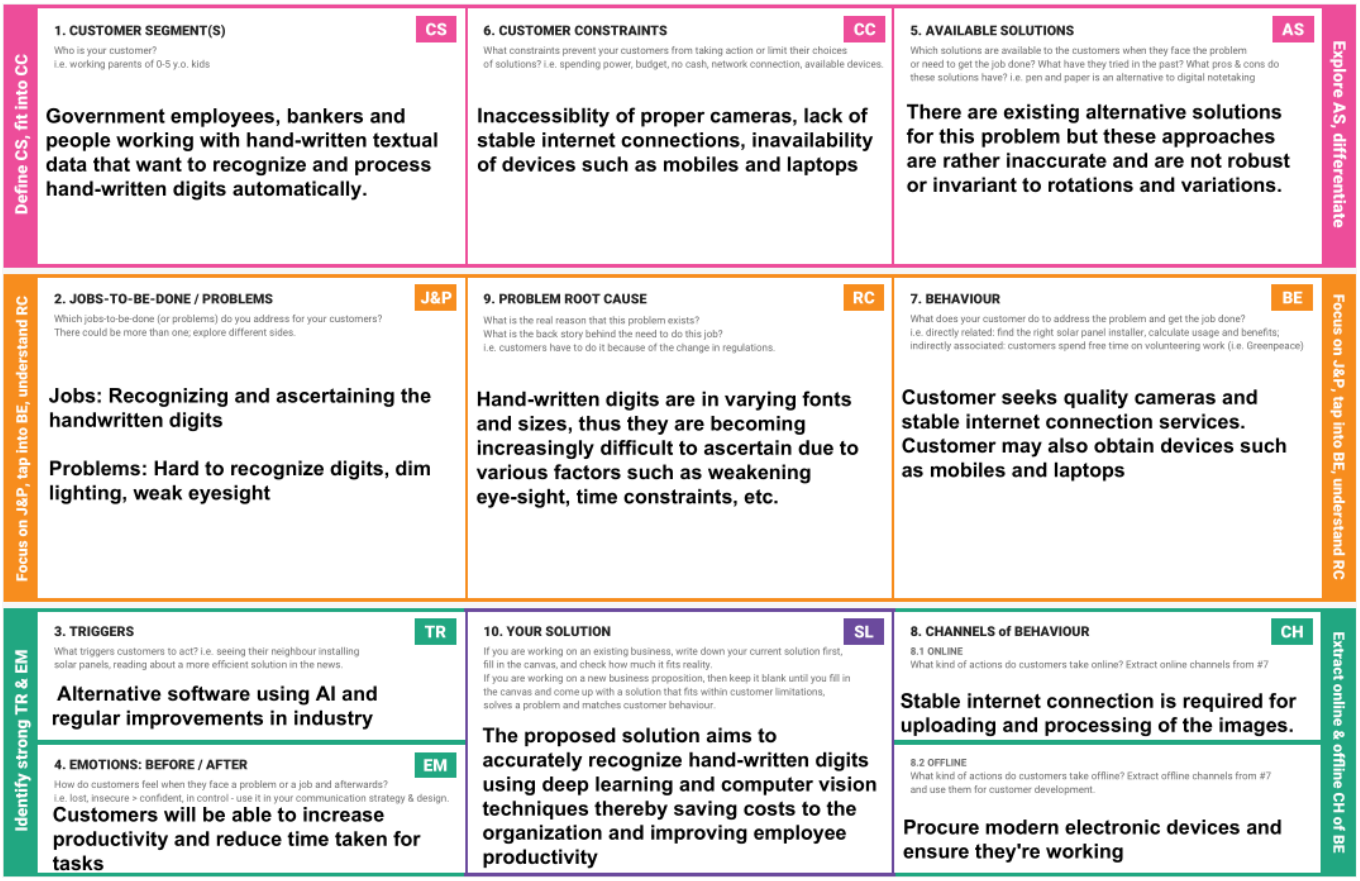
Using large-scale datasets for training

Importance
If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?
Feasibility
Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

3.3 Proposed Solution

S. No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	A machine learning system for automatic Handwritten Digit Recognition
2	Idea / Solution description	The handwritten digits are recognised by machine learning algorithms which use deep convolutional neural networks (CNNs) trained on a large-scale dataset.
3	Novelty / Uniqueness	The approach to detect handwritten digits from pixels of images is performed using state of the art machine learning and computer vision techniques. The previous approaches have various constraints
4	Social Impact / Customer Satisfaction	Handwritten digits can be recognised easily without any strenuous efforts. This reduces time and improves productivity for people
5	Business Model (Revenue Model)	It is a software as a service business model where the highly scalable service can be used for detecting handwritten digits efficiently at low cost
6	Scalability of the Solution	The solution is deployed as a RESTful machine learning service which is highly containerized. Thus, the solution can scale to higher processing loads.

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

Government employees, bankers and people working with hand-written textual data that want to recognize and process hand-written digits automatically.

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.

Inaccessiblity of proper cameras, lack of stable internet connections, inavailability of devices such as mobiles and laptops

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 existing alternative solutions for this problem but these approaches are rather inaccurate and are not robust or invariant to rotations and variations.

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.

Jobs: Recognizing and ascertaining the handwritten digits

Problems: Hard to recognize digits, dim lighting, weak eyesight

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.

Hand-written digits are in varying fonts and sizes, thus they are becoming increasingly difficult to ascertain due to various factors such as weakening eye-sight, time constraints, etc.

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)

Customer seeks quality cameras and stable internet connection services. Customer may also obtain devices such as mobiles and laptops

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 neighbour installing solar panels, reading about a more efficient solution in the news.

Alternative software using AI and regular improvements in industry

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.

Customers will be able to increase productivity and reduce time taken for tasks

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 behaviour.

The proposed solution aims to accurately recognize hand-written digits using deep learning and computer vision techniques thereby saving costs to the organization and improving employee productivity

8. CHANNELS of BEHAVIOUR CH

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

Stable internet connection is required for uploading and processing of the images.

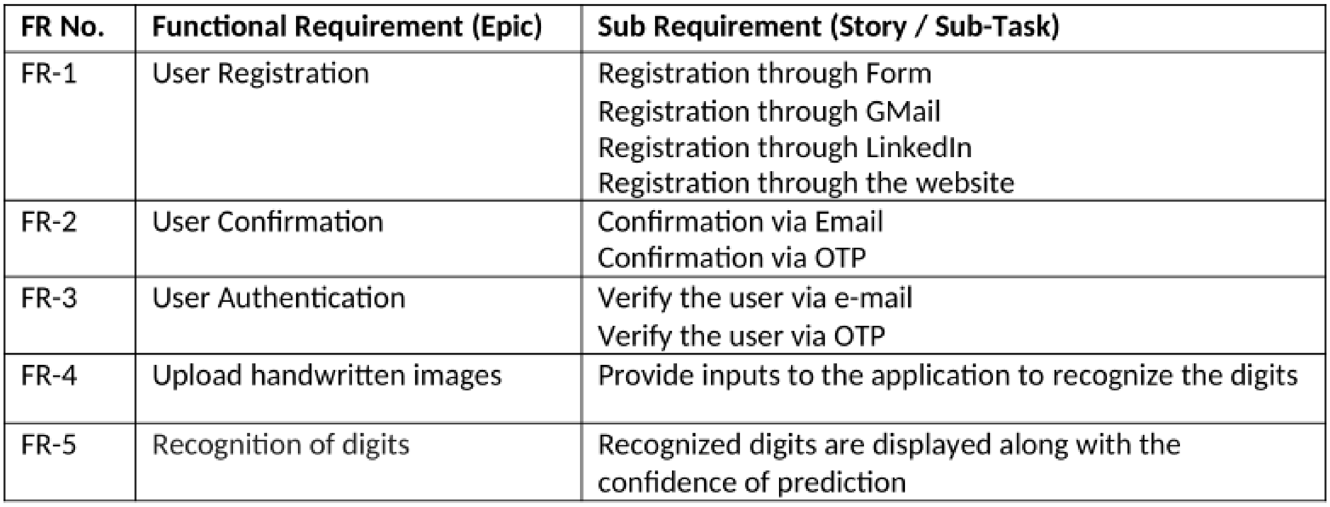
8.2 OFFLINE
What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.

Procure modern electronic devices and ensure they're working

Extract online & offline CH of BE

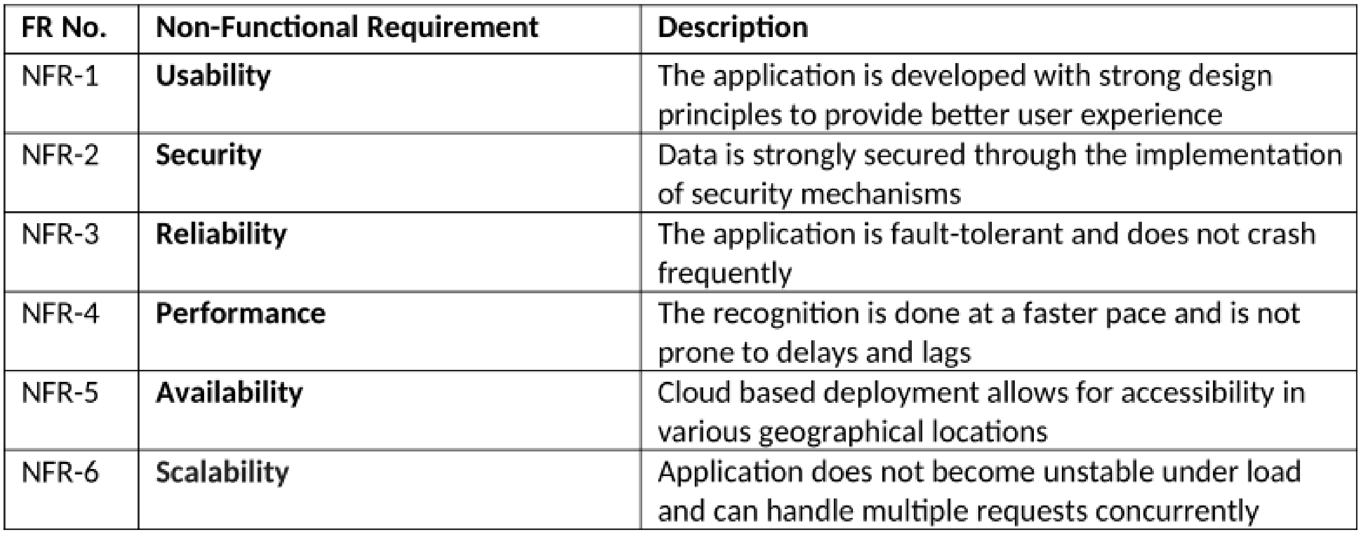
4. REQUIREMENT ANALYSIS

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 Registration through the website
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Authentication	Verify the user via e-mail Verify the user via OTP
FR-4	Upload handwritten images	Provide inputs to the application to recognize the digits
FR-5	Recognition of digits	Recognized digits are displayed along with the confidence of prediction

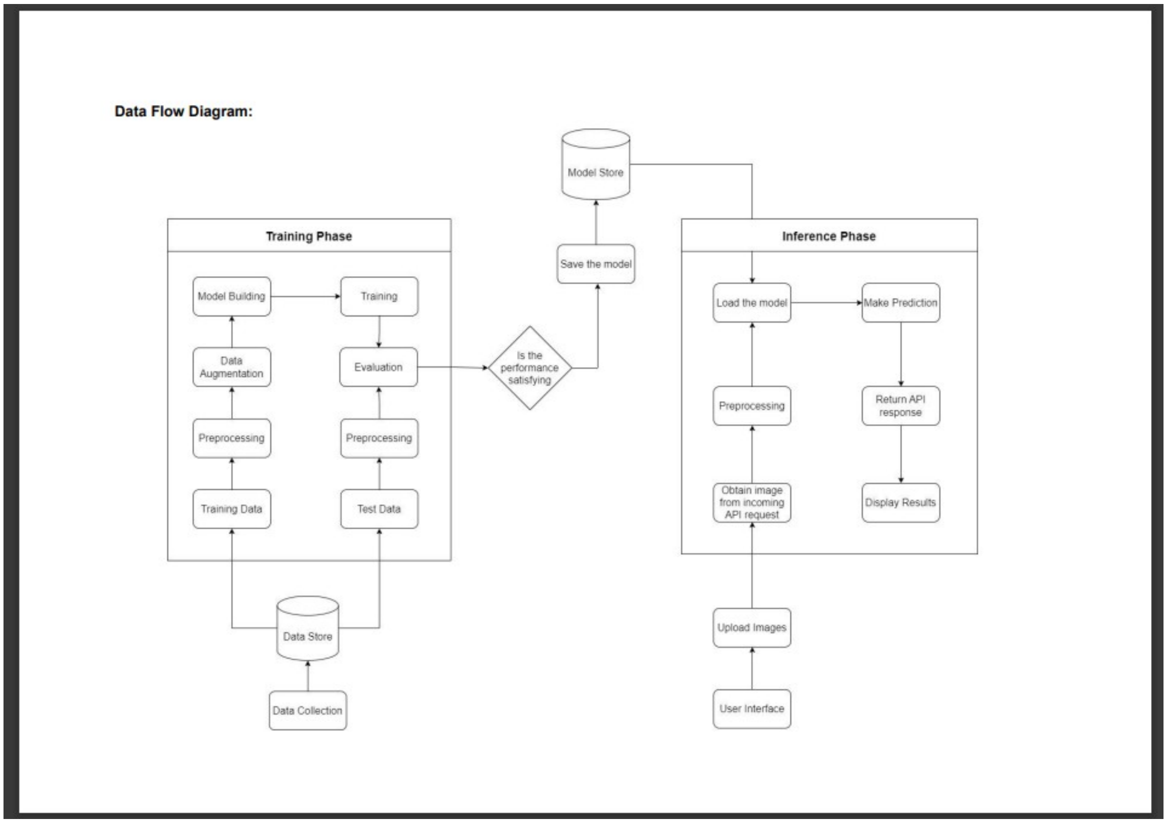
4.2 Non-Functional requirements



FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The application is developed with strong design principles to provide better user experience
NFR-2	**Security**	Data is strongly secured through the implementation of security mechanisms
NFR-3	**Reliability**	The application is fault-tolerant and does not crash frequently
NFR-4	**Performance**	The recognition is done at a faster pace and is not prone to delays and lags
NFR-5	**Availability**	Cloud based deployment allows for accessibility in various geographical locations
NFR-6	**Scalability**	Application does not become unstable under load and can handle multiple requests concurrently

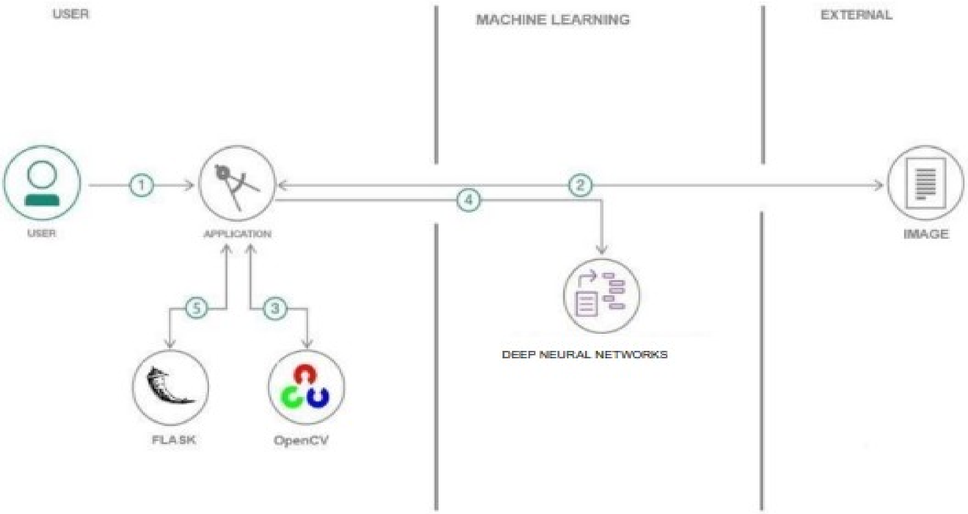
5. PROJECT DESIGN

5.1 Data Flow Diagrams

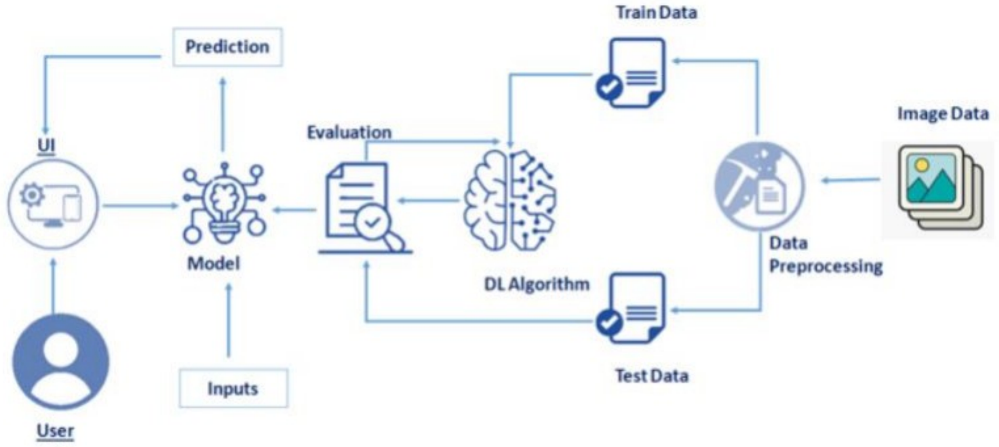


5.2 Solution & Technical Architecture

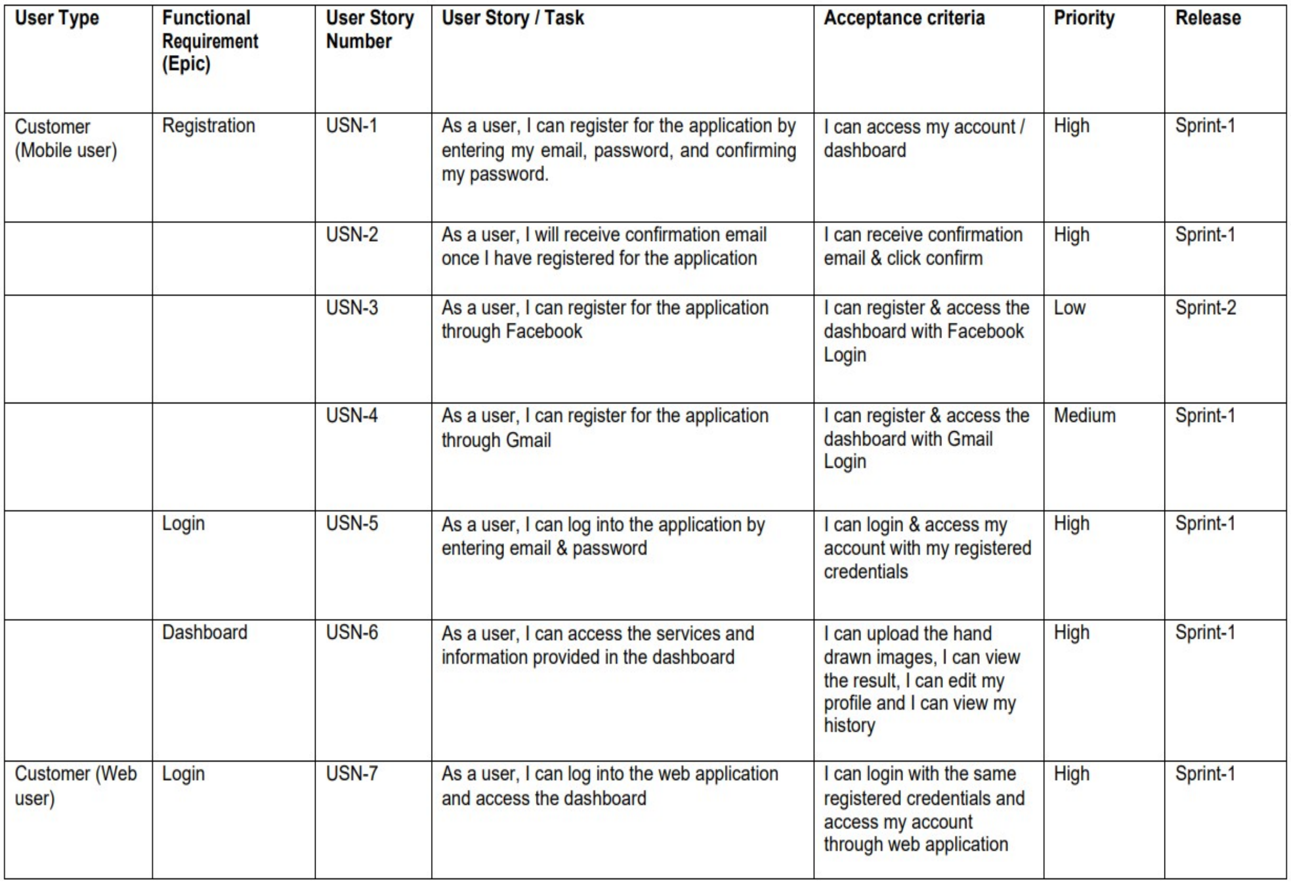
5.2.1 Solution Architecture



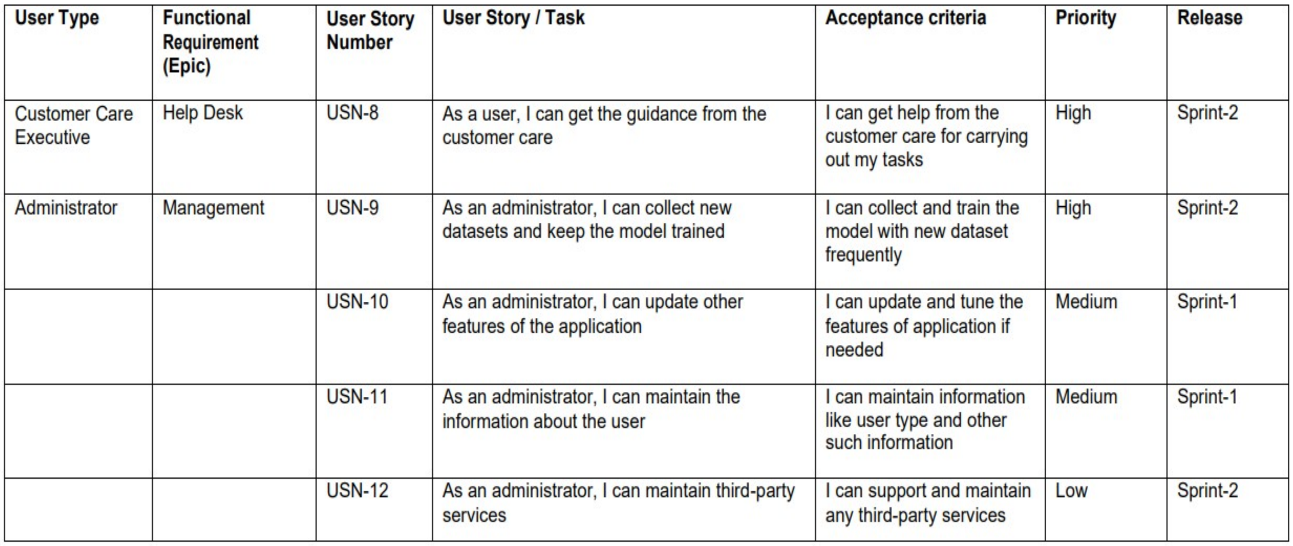
5.2.2 Technical Architecture



5.3 User Stories



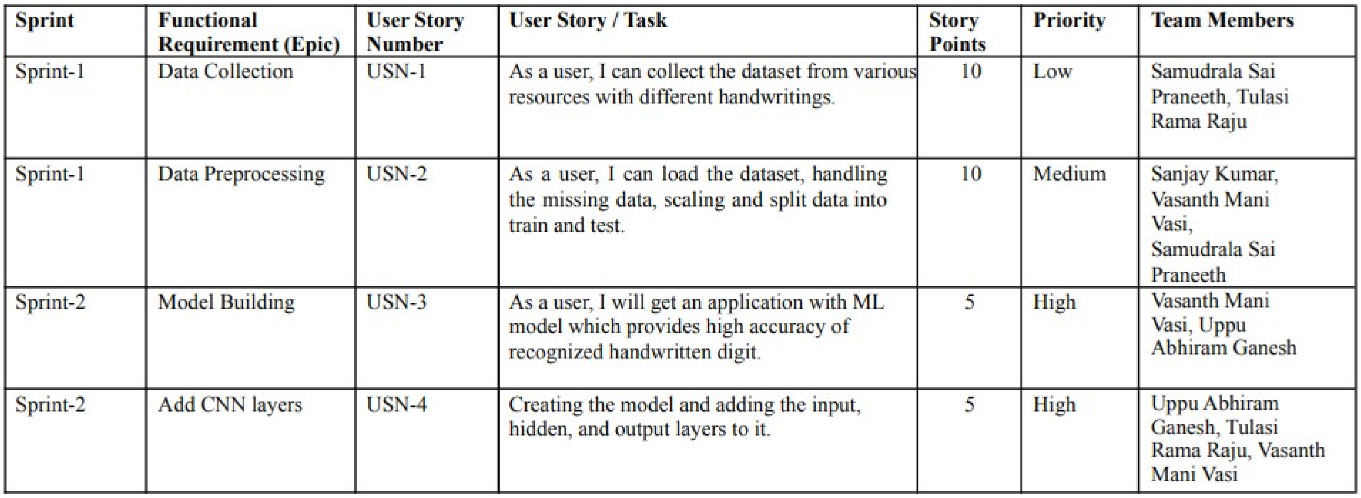
User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail	I can register & access the dashboard with Gmail Login	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	I can login & access my account with my registered credentials	High	Sprint-1
	Dashboard	USN-6	As a user, I can access the services and information provided in the dashboard	I can upload the hand drawn images, I can view the result, I can edit my profile and I can view my history	High	Sprint-1
Customer (Web user)	Login	USN-7	As a user, I can log into the web application and access the dashboard	I can login with the same registered credentials and access my account through web application	High	Sprint-1

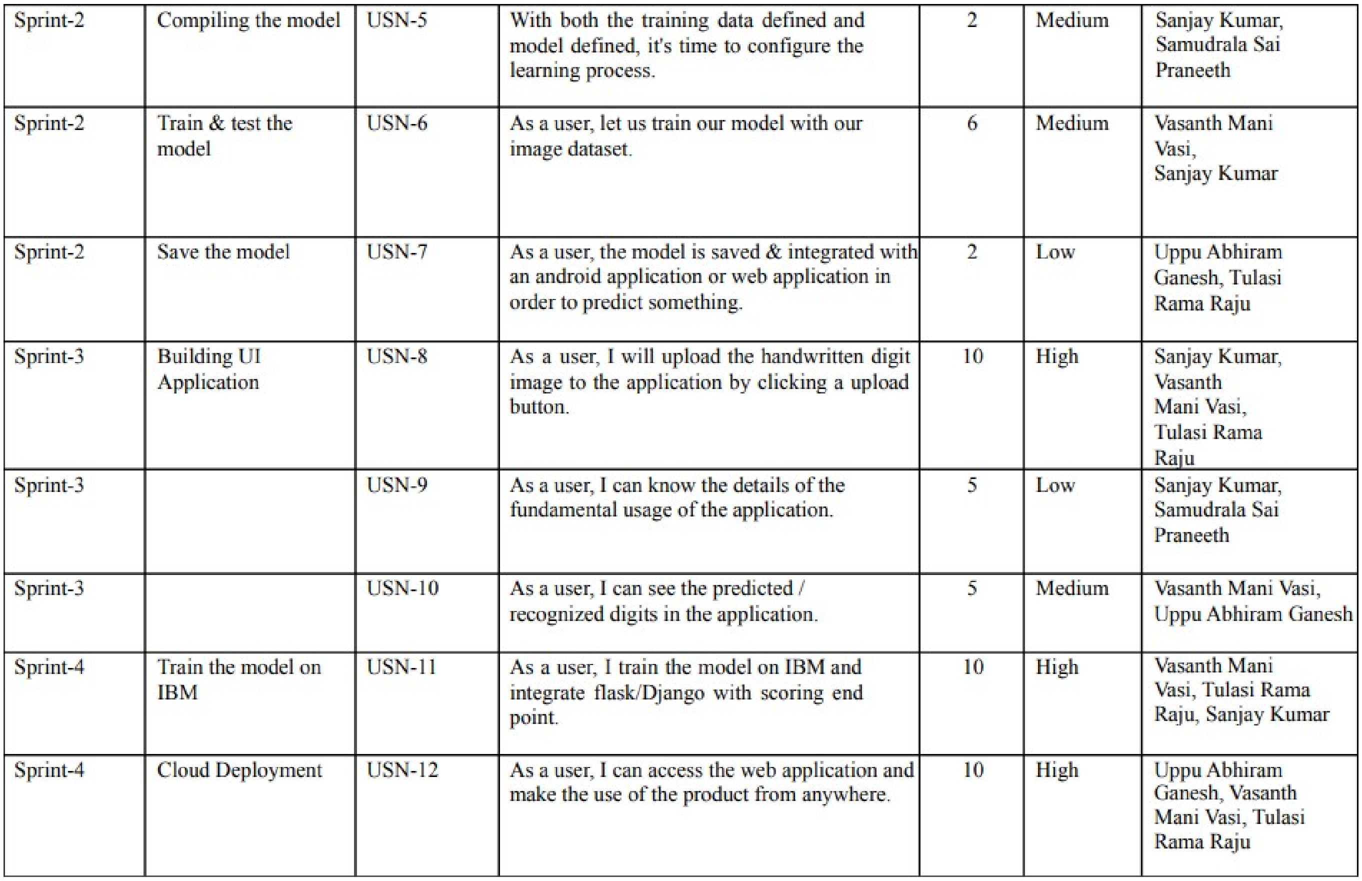


User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer Care Executive	Help Desk	USN-8	As a user, I can get the guidance from the customer care	I can get help from the customer care for carrying out my tasks	High	Sprint-2
Administrator	Management	USN-9	As an administrator, I can collect new datasets and keep the model trained	I can collect and train the model with new dataset frequently	High	Sprint-2
		USN-10	As an administrator, I can update other features of the application	I can update and tune the features of application if needed	Medium	Sprint-1
		USN-11	As an administrator, I can maintain the information about the user	I can maintain information like user type and other such information	Medium	Sprint-1
		USN-12	As an administrator, I can maintain third-party services	I can support and maintain any third-party services	Low	Sprint-2

6. PROJECT PLANNING & SCHEDULING

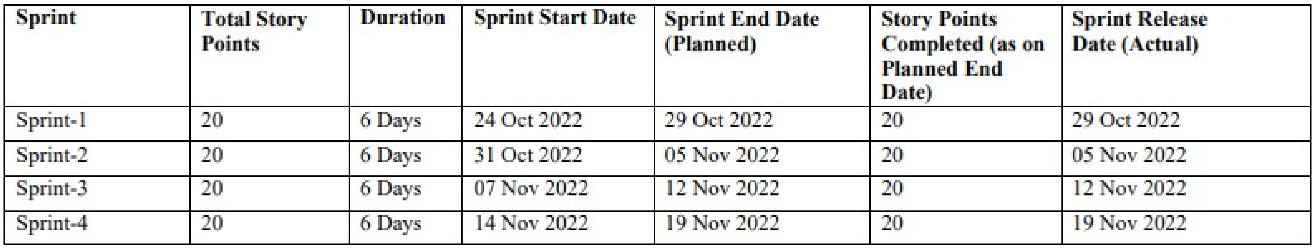
6.1 Sprint Planning & Estimation





Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data Collection	USN-1	As a user, I can collect the dataset from various resources with different handwritings.	10	Low	Samudrala Sai Praneeth, Tulasi Rama Raju
Sprint-1	Data Preprocessing	USN-2	As a user, I can load the dataset, handling the missing data, scaling and split data into train and test.	10	Medium	Sanjay Kumar, Vasanth Mani Vasi, Samudrala Sai Praneeth
Sprint-2	Model Building	USN-3	As a user, I will get an application with ML model which provides high accuracy of recognized handwritten digit.	5	High	Vasanth Mani Vasi, Uppu Abhiram Ganesh
Sprint-2	Add CNN layers	USN-4	Creating the model and adding the input, hidden, and output layers to it.	5	High	Uppu Abhiram Ganesh, Tulasi Rama Raju, Vasanth Mani Vasi
Sprint-2	Compiling the model	USN-5	With both the training data defined and model defined, it's time to configure the learning process.	2	Medium	Sanjay Kumar, Samudrala Sai Praneeth
Sprint-2	Train & test the model	USN-6	As a user, let us train our model with our image dataset.	6	Medium	Vasanth Mani Vasi, Sanjay Kumar
Sprint-2	Save the model	USN-7	As a user, the model is saved & integrated with an android application or web application in order to predict something.	2	Low	Uppu Abhiram Ganesh, Tulasi Rama Raju
Sprint-3	Building UI Application	USN-8	As a user, I will upload the handwritten digit image to the application by clicking a upload button.	10	High	Sanjay Kumar, Vasanth Mani Vasi, Tulasi Rama Raju
Sprint-3		USN-9	As a user, I can know the details of the fundamental usage of the application.	5	Low	Sanjay Kumar, Samudrala Sai Praneeth
Sprint-3		USN-10	As a user, I can see the predicted / recognized digits in the application.	5	Medium	Vasanth Mani Vasi, Uppu Abhiram Ganesh
Sprint-4	Train the model on IBM	USN-11	As a user, I train the model on IBM and integrate flask/Django with scoring end point.	10	High	Vasanth Mani Vasi, Tulasi Rama Raju, Sanjay Kumar
Sprint-4	Cloud Deployment	USN-12	As a user, I can access the web application and make the use of the product from anywhere.	10	High	Uppu Abhiram Ganesh, Vasanth Mani Vasi, Tulasi Rama Raju

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 & SOLUTION

7.1 Feature1

Homepage:

Hand written digit website is a responsive web application which detects the number of the uploaded drawn image. The home page contains information about the application. It also contains the recognize tab to upload the hand drawn image.

index.html

```
<html>
   <head>
      <title>Handwritten Digit Recognition System</title>
      <link rel="stylesheet" href="../static/style.css">
      <script>
         window.onload = function(){
            document.getElementById("loader").style.display="none";
            document.getElementById("content").style.visibility="visible";
         }
      </script>
   </head>
<body class="style1">
   <span id="loader">
      <img src="../static/loading.gif">
   </span>
   <span id="content">
   <div class="header">
      <p><a href="/">Home</a><a href="/upload">Recognize</a></p>
   </div>
   <h1>Handwritten Recognition System</h1>
   <p class="introduction">Handwritten Text Recognition is a technology that is much needed in this world
as of today.
      The digit recognition system is used to recognize the digits from different sources like email, bank
cheque, papers, images, etc.
      Before proper implementation of this technology we have relied on writting texts with our own hands
```

```
which can result in errors.
     It's diffcult to store and access physical data with effiecieny.
     The project presents recognizing the handwritten digits (0 to 9) from the famous MNIST dataset.
     Here we will be using artificial neural networks convolution neural networks.
  </p>
  </span>
</body>
</html>
```

7.2 Feature 2

Prediction page:

The user will add the hand-written image in the home page and press the recognize button. Then the page will redirect to a new page and provide information about the recognized digit.

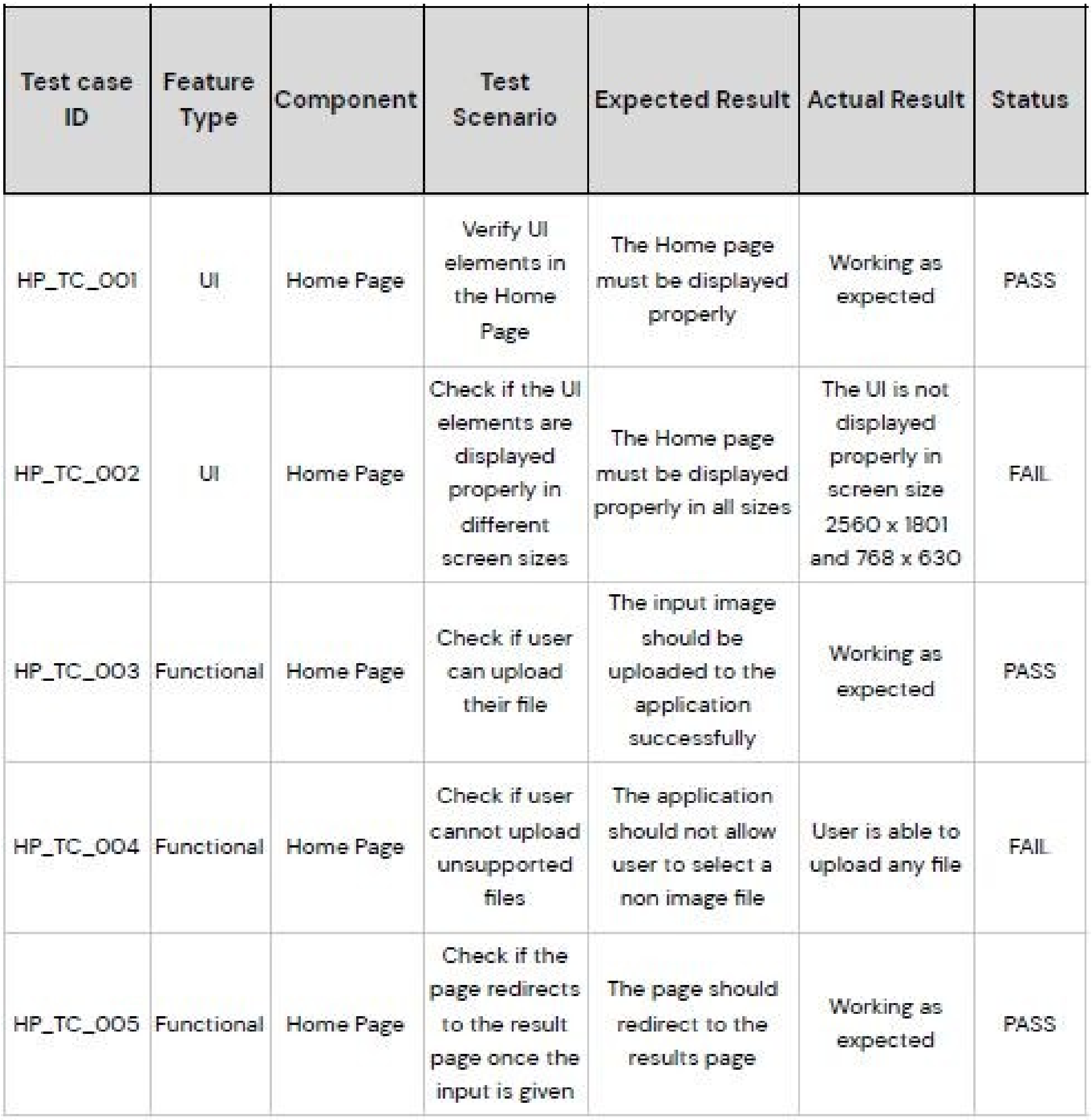
app.py

```
import os
import pybase64
import requests
import numpy as np
from PIL import Image
from io import BytesIO
import tensorflow as tf

from tensorflow import keras
from keras.models import load_model
from flask import Flask, render_template, request

app = Flask(__name__)

API_KEY = "" # Removed API key from code before pushing to github
API_ENDPOINT = "https://us-south.ml.cloud.ibm.com/ml/v4/deployments/680e8d3b-1e5b-40e9-
b639-89cf48bcdf6e/predictions?version=2022-11-26"

token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey":
API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})
mltoken = token_response.json()["access_token"]
header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}

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

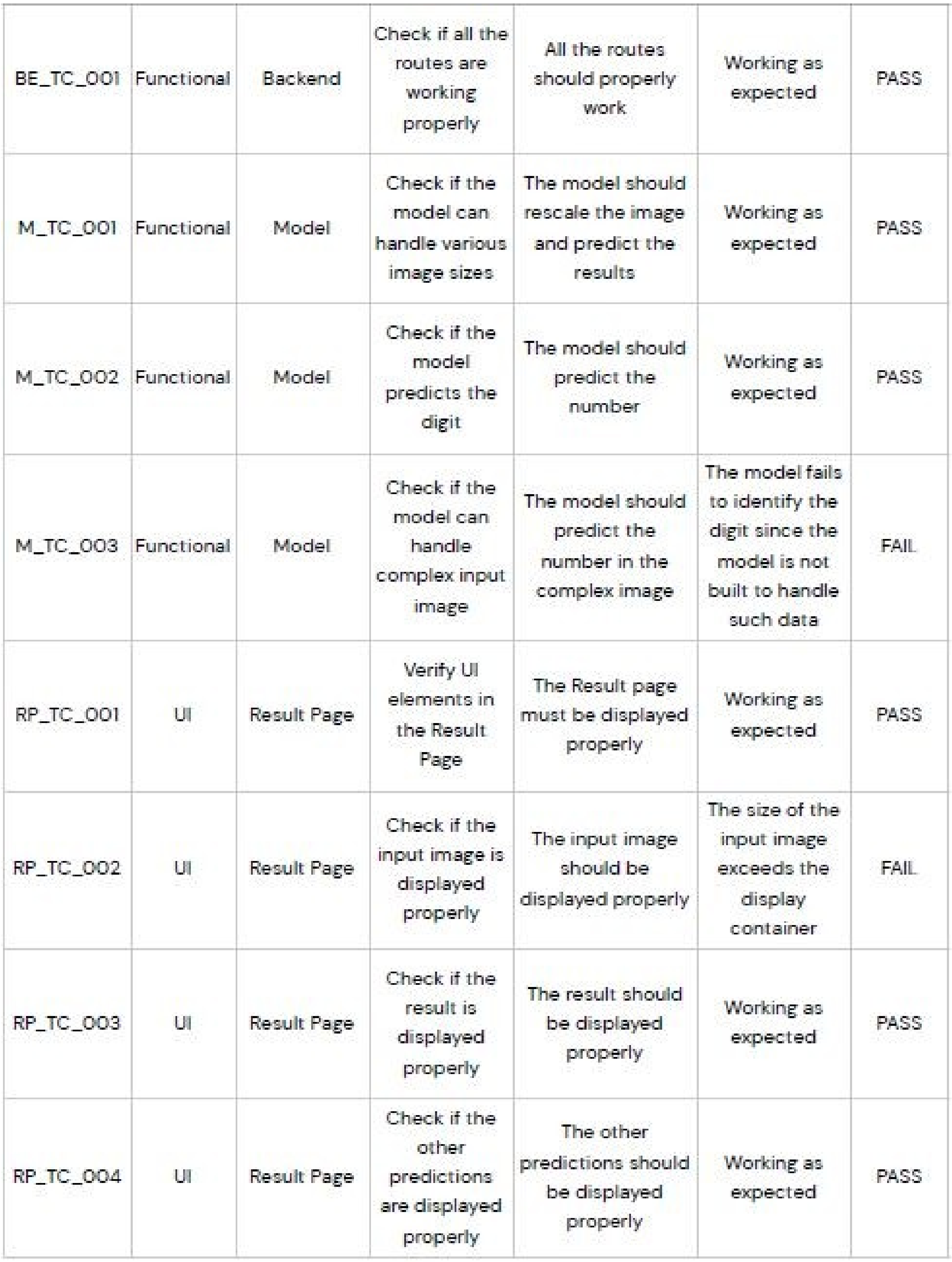
```
@app.route('/upload')
def upload_file2():
    return render_template('predict.html')


@app.route('/predict', methods=['POST'])
def upload_image_file():
    if request.method == 'POST':
        file = request.files['file']
        img = Image.open(request.files['file'].stream).convert('L')
        img = img.resize((28, 28))
        im2arr = np.array(img)
        im2arr = im2arr.reshape(28, 28, 1)

        payload_scoring = {"input_data": [{"fields": [], "values": [im2arr.tolist()]}]}
        response_scoring = requests.post(
            API_ENDPOINT,
            json=payload_scoring,
            headers={'Authorization': 'Bearer ' + mltoken}
        )

        response = response_scoring.json()
        print(response)
        prediction = response['predictions'][0]['values'][0]
        y_pred = response['predictions'][0]['values'][0][1]

        prediction_percentage = str(round(max(prediction[0])*100, 2))+"%"
        filename = file.filename
        path = os.path.join("static/images", filename)
        img = Image.open(file.stream)
        file.save(path)

        if filename.endswith('jpg') or filename.endswith('jpeg'):
            with BytesIO() as buf:
                img.save(buf, 'jpeg')
                image_bytes = buf.getvalue()
            encoded_string = pybase64.b64encode(image_bytes).decode()
            encoded_string = "data:image/jpeg;base64,"+encoded_string
        if filename.endswith('png'):
            with BytesIO() as buf:
                img.save(buf, 'png')
                image_bytes = buf.getvalue()
            encoded_string = pybase64.b64encode(image_bytes).decode()
            encoded_string = "data:image/png;base64,"+encoded_string
        os.remove(path)
        if 0 <= y_pred <= 9:
            return render_template("result.html", digit=y_pred, user_image=encoded_string,
percentage=prediction_percentage, showcase=str(y_pred))
        else:
            return render_template("result.html", digit="No digit found.", user_image=encoded_string,
percentage=prediction_percentage)
    return render_template('predict.html')

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=8000, debug=True)
```

8. TESTING

8.1 Test Case



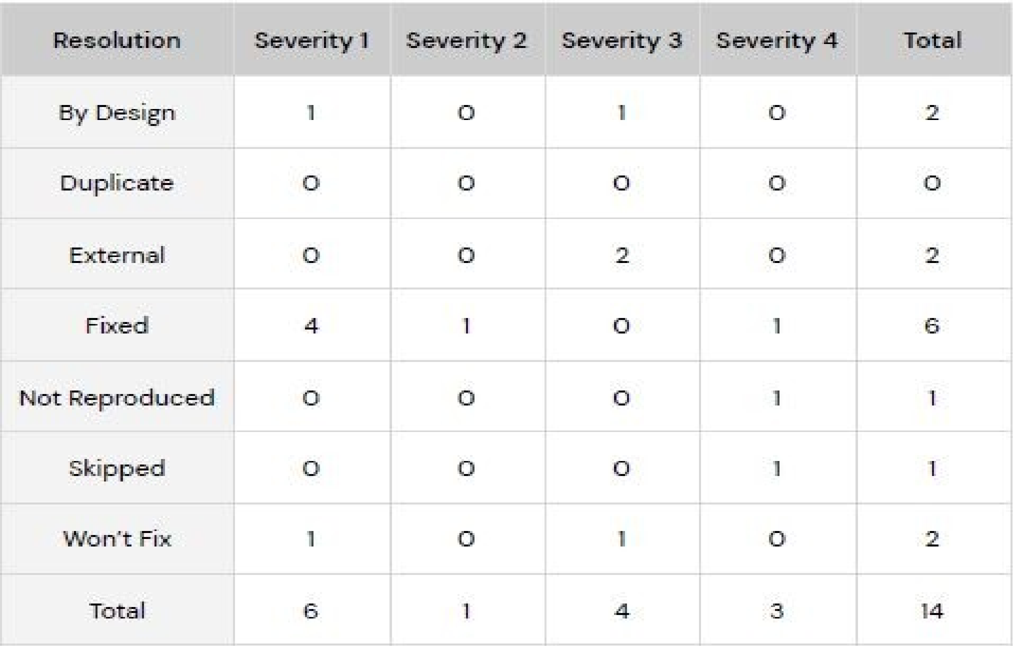
Test case ID	Feature Type	Component	Test Scenario	Expected Result	Actual Result	Status
HP_TC_001	UI	Home Page	Verify UI elements in the Home Page	The Home page must be displayed properly	Working as expected	PASS
HP_TC_002	UI	Home Page	Check if the UI elements are displayed properly in different screen sizes	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
HP_TC_003	Functional	Home Page	Check if user can upload their file	The input image should be uploaded to the application successfully	Working as expected	PASS
HP_TC_004	Functional	Home Page	Check if user cannot upload unsupported files	The application should not allow user to select a non image file	User is able to upload any file	FAIL
HP_TC_005	Functional	Home Page	Check if the page redirects to the result page once the input is given	The page should redirect to the results page	Working as expected	PASS



BE_TC_001	Functional	Backend	Check if all the routes are working properly	All the routes should properly work	Working as expected	PASS
M_TC_001	Functional	Model	Check if the model can handle various image sizes	The model should rescale the image and predict the results	Working as expected	PASS
M_TC_002	Functional	Model	Check if the model predicts the digit	The model should predict the number	Working as expected	PASS
M_TC_003	Functional	Model	Check if the model can handle complex input image	The model should predict the number in the complex image	The model fails to identify the digit since the model is not built to handle such data	FAIL
RP_TC_001	UI	Result Page	Verify UI elements in the Result Page	The Result page must be displayed properly	Working as expected	PASS
RP_TC_002	UI	Result Page	Check if the input image is displayed properly	The input image should be displayed properly	The size of the input image exceeds the display container	FAIL
RP_TC_003	UI	Result Page	Check if the result is displayed properly	The result should be displayed properly	Working as expected	PASS
RP_TC_004	UI	Result Page	Check if the other predictions are displayed properly	The other predictions should be displayed properly	Working as expected	PASS

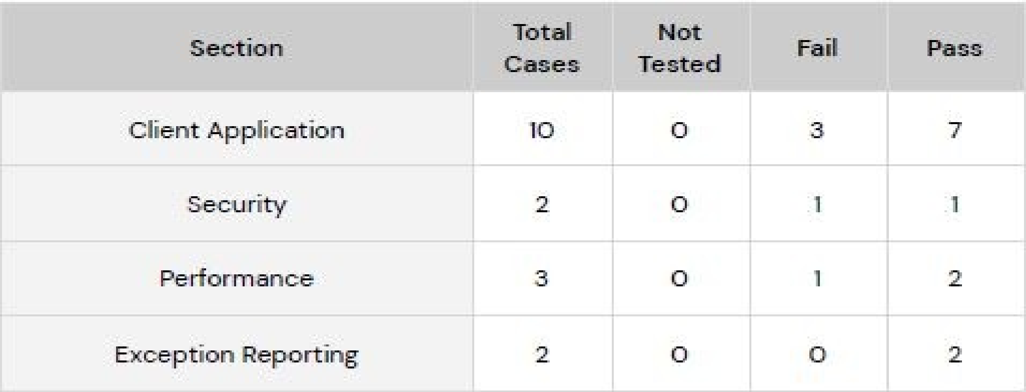
8.2 User Acceptance Testing

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

Test Case Analysis

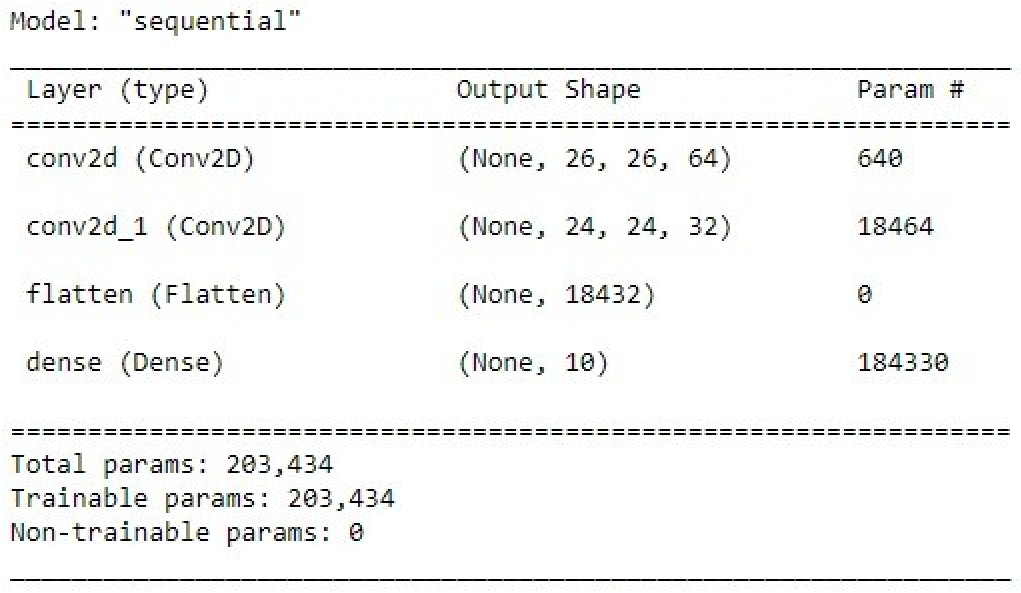


Section	Total Cases	Not Tested	Fail	Pass
Client Application	10	0	3	7
Security	2	0	1	1
Performance	3	0	1	2
Exception Reporting	2	0	0	2

9. RESULTS

9.1 Performance Metrics

9.1.1 Model Summary

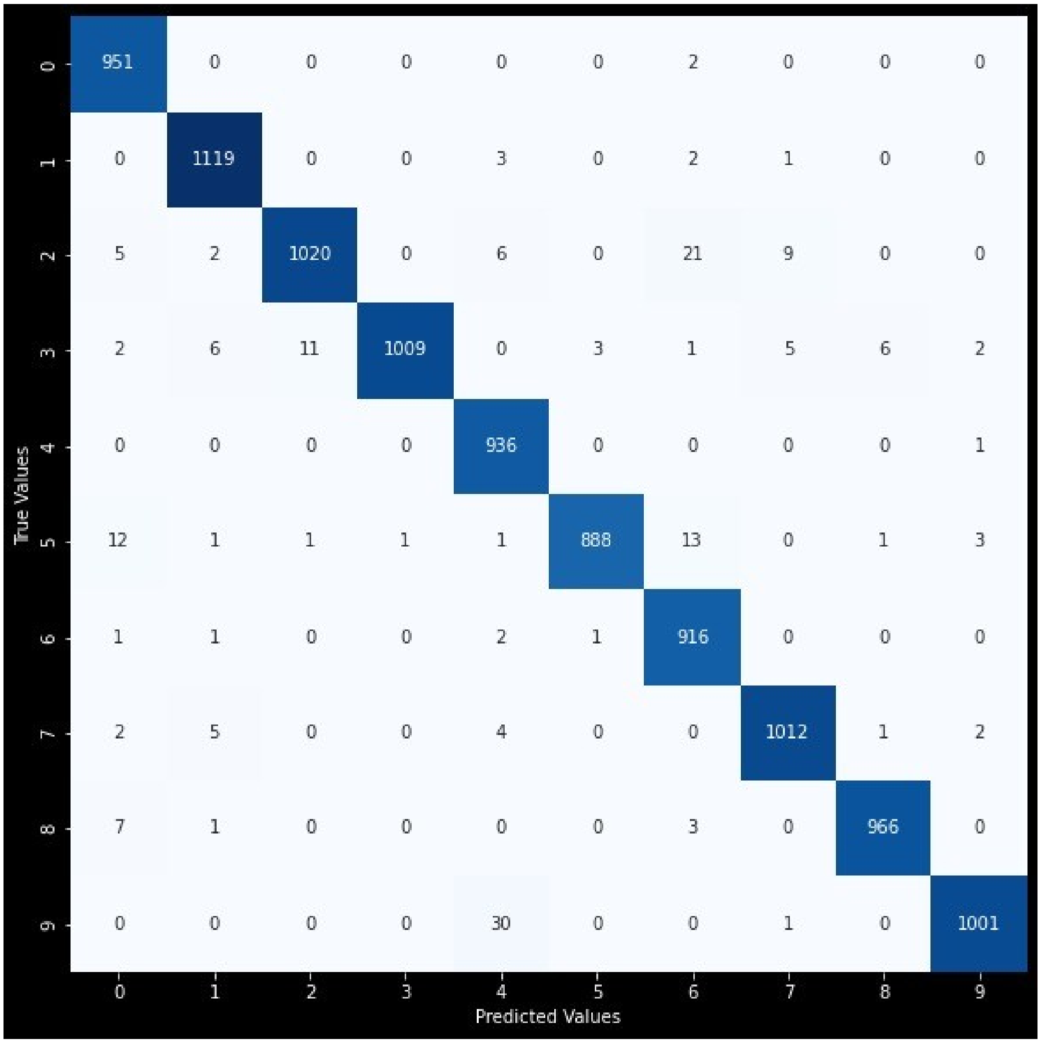


```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 26, 26, 64)        640

 conv2d_1 (Conv2D)           (None, 24, 24, 32)        18464

 flatten (Flatten)           (None, 18432)             0

 dense (Dense)               (None, 10)                184330

=================================================================
Total params: 203,434
Trainable params: 203,434
Non-trainable params: 0
_________________________________________________________________
```

9.1.2 Accuracy

Content	Value
Training Accuracy	**96.98%**
Training Loss	**0.0982**
Validation Accuracy	**97.39%**
Validation Loss	**0.0828**

9.1.3 Confusion Matrix

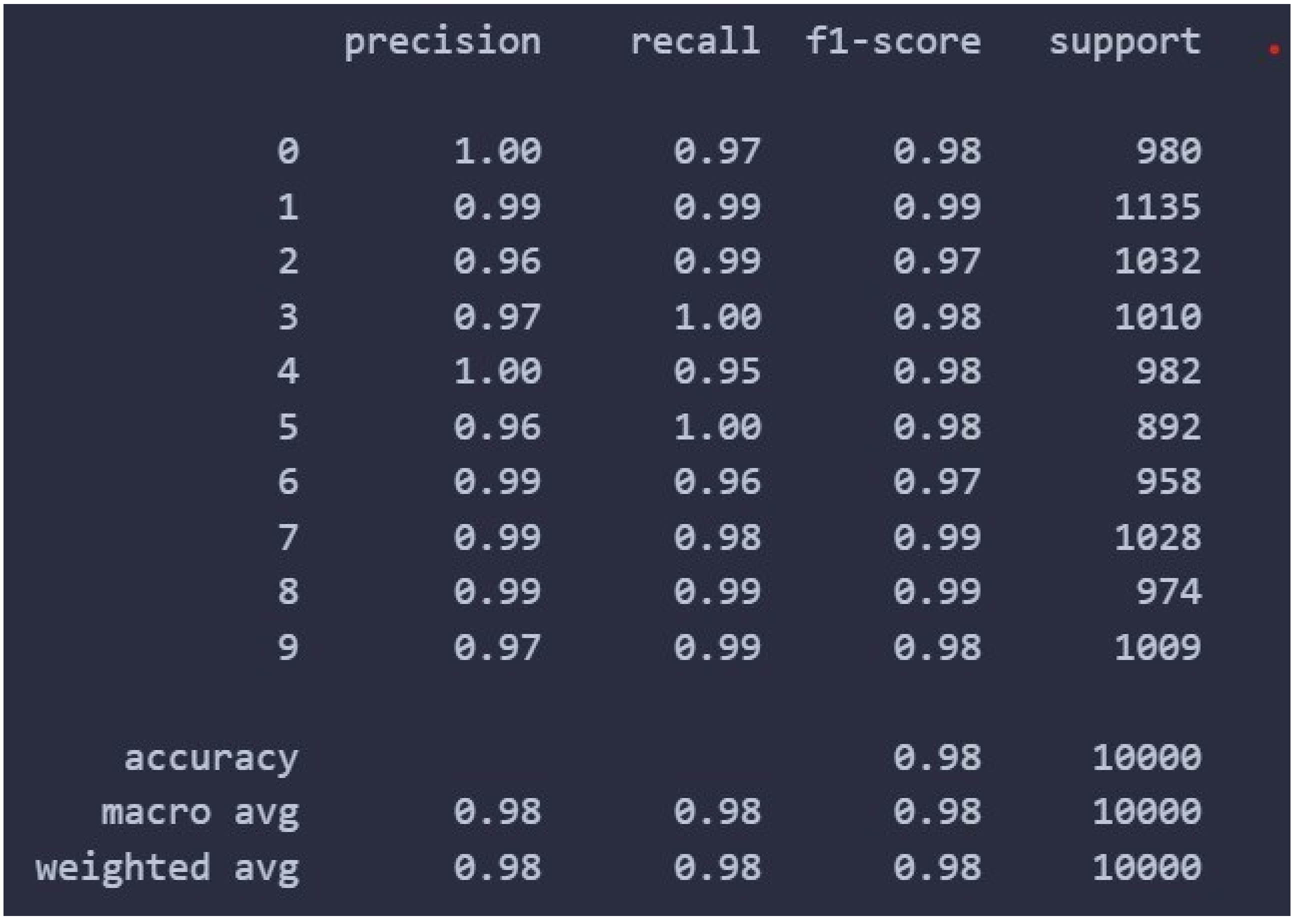


True Values

	0	1	2	3	4	5	6	7	8	9
0	951	0	0	0	0	0	2	0	0	0
1	0	1119	0	0	3	0	2	1	0	0
2	5	2	1020	0	6	0	21	9	0	0
3	2	6	11	1009	0	3	1	5	6	2
4	0	0	0	0	936	0	0	0	0	1
5	12	1	1	1	1	888	13	0	1	3
6	1	1	0	0	2	1	916	0	0	0
7	2	5	0	0	4	0	0	1012	1	2
8	7	1	0	0	0	0	3	0	966	0
9	0	0	0	0	30	0	0	1	0	1001

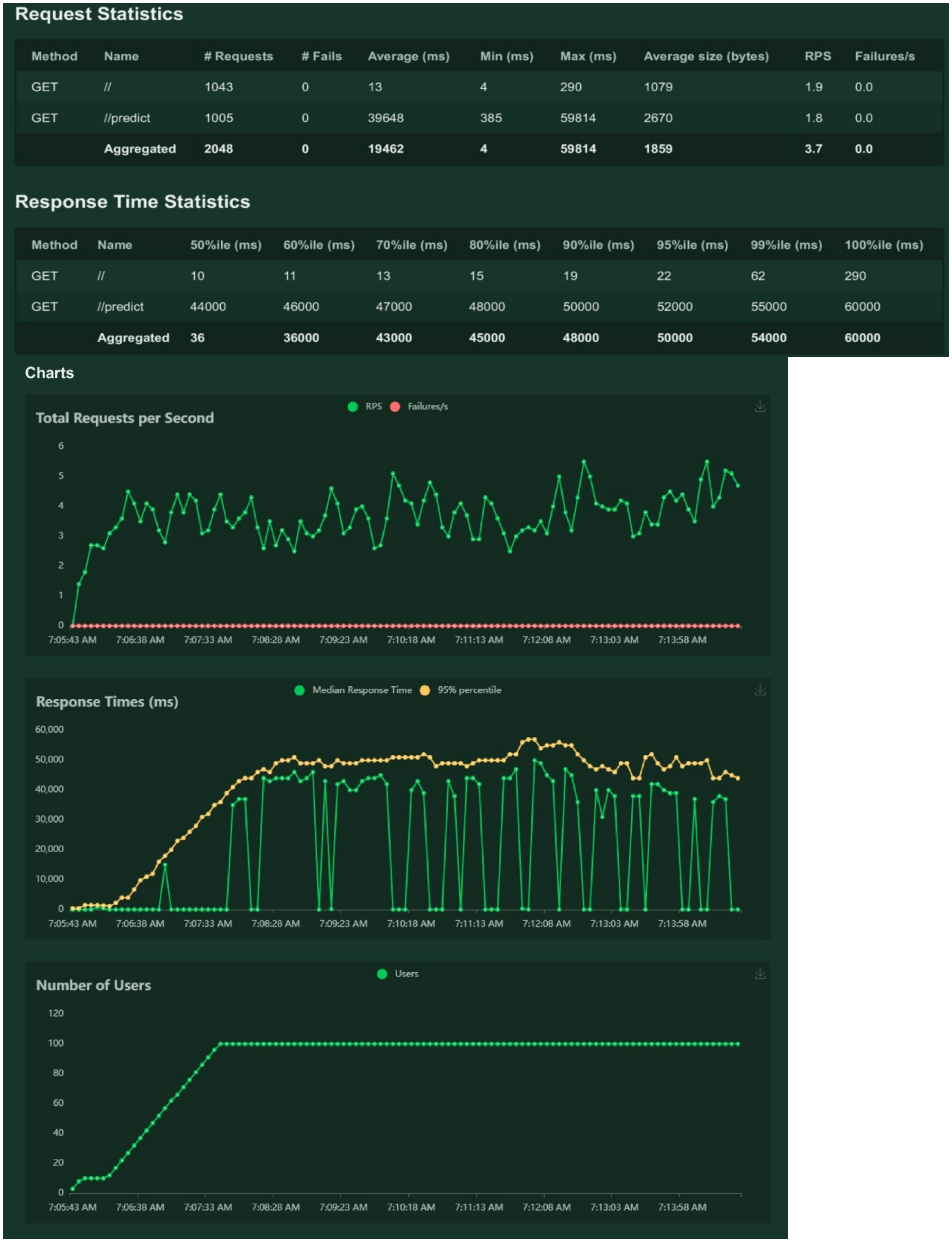
Predicted Values

9.1.4 Classification Report



```
              precision    recall  f1-score   support

           0       1.00      0.97      0.98       980
           1       0.99      0.99      0.99      1135
           2       0.96      0.99      0.97      1032
           3       0.97      1.00      0.98      1010
           4       1.00      0.95      0.98       982
           5       0.96      1.00      0.98       892
           6       0.99      0.96      0.97       958
           7       0.99      0.98      0.99      1028
           8       0.99      0.99      0.99       974
           9       0.97      0.99      0.98      1009

    accuracy                           0.98     10000
   macro avg       0.98      0.98      0.98     10000
weighted avg       0.98      0.98      0.98     10000
```

9.1.5 Application Test Report



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

10. ADVANTAGES & DISADVANTAGES

Advantages:

- Reduces the manual work.
- More accurate compared to an average human.
- Capable of handling a lot of data.
- Can be used from anywhere from any device.

Disadvantages:

- Cannot handle complex data.
- All the data must be in digital format.
- Requires a high performing server.
- Prone to occasional errors.

11. CONCLUSION

This project demonstrated a web application that uses machine learning to recognize handwritten numbers. Flask, HTML, CSS, JavaScript, and a few other technologies were used to create this project. The model predicts the handwritten digit using a CNN network. During testing, the model achieved a 98.91% 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

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

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

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

13. APPENDIX

Source Code

MODEL CREATION

```
import numpy as np
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 Convolution2D,MaxPooling2D,Flatten,Dense

(X_train, Y_train), (X_test, Y_test) = mnist.load_data()
X_train = X_train.reshape(60000,28,28,1).astype('float32')
X_test = X_test.reshape(10000,28,28,1).astype('float32')
number_of_classes = 10
Y_train = np_utils.to_categorical(Y_train, number_of_classes)
Y_test = np_utils.to_categorical(Y_test,number_of_classes)

model = Sequential()
model.add(Convolution2D(64, (3,3), input_shape=(28,28,1),activation='relu'))
model.add(Convolution2D(32,(3,3),activation='relu'))
model.add(Flatten())
model.add(Dense(number_of_classes, activation='softmax'))
model.compile(loss='categorical_crossentropy',optimizer='Adam',metrics=['accuracy'])

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

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

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

FLASK APP

```
import os
import pybase64
import requests
import numpy as np
from PIL import Image
from io import BytesIO
import tensorflow as tf
```

```
from tensorflow import keras
from keras.models import load_model
from flask import Flask, render_template, request

app = Flask(__name__)

API_KEY = "" # Remove API Key before pushing to github
API_ENDPOINT = "https://us-south.ml.cloud.ibm.com/ml/v4/deployments/680e8d3b-1e5b-40e9-
b639-89cf48bcdf6e/predictions?version=2022-11-26"

token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey":
API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})
mltoken = token_response.json()["access_token"]
header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}

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


@app.route('/upload')
def upload_file2():
    return render_template('predict.html')


@app.route('/predict', methods=['POST'])
def upload_image_file():
    if request.method == 'POST':
        file = request.files['file']
        img = Image.open(request.files['file'].stream).convert('L')
        img = img.resize((28, 28))
        im2arr = np.array(img)
        im2arr = im2arr.reshape(28, 28, 1)

        payload_scoring = {"input_data": [{"fields": [], "values": [im2arr.tolist()]}]}
        response_scoring = requests.post(
            API_ENDPOINT,
            json=payload_scoring,
            headers={'Authorization': 'Bearer ' + mltoken}
        )

        response = response_scoring.json()
        print(response)
        prediction = response['predictions'][0]['values'][0]
        y_pred = response['predictions'][0]['values'][0][1]

        prediction_percentage = str(round(max(prediction[0])*100, 2))+"%"
        filename = file.filename
        path = os.path.join("static/images", filename)
        img = Image.open(file.stream)
        file.save(path)

        if filename.endswith('jpg') or filename.endswith('jpeg'):
            with BytesIO() as buf:
                img.save(buf, 'jpeg')
                image_bytes = buf.getvalue()
            encoded_string = pybase64.b64encode(image_bytes).decode()
            encoded_string = "data:image/jpeg;base64,"+encoded_string
        if filename.endswith('png'):
            with BytesIO() as buf:
                img.save(buf, 'png')
                image_bytes = buf.getvalue()
            encoded_string = pybase64.b64encode(image_bytes).decode()
            encoded_string = "data:image/png;base64,"+encoded_string
        os.remove(path)

        if 0 <= y_pred <= 9:
            return render_template("result.html", digit=y_pred, user_image=encoded_string,
percentage=prediction_percentage, showcase=str(y_pred))
        else:
            return render_template("result.html", digit="No digit found.", user_image=encoded_string,
percentage=prediction_percentage)
    return render_template('predict.html')

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=8000, debug=True)
```

HOME PAGE

```
<html>
  <head>
    <title>Handwritten Digit Recognition System</title>
    <link rel="stylesheet" href="../static/style.css">
    <script>
      window.onload = function(){
        document.getElementById("loader").style.display="none";
        document.getElementById("content").style.visibility="visible";
      }
    </script>
  </head>
<body class="style1">
  <span id="loader">
    <img src="../static/loading.gif">
  </span>
  <span id="content">
```

```
    <div class="header">
        <p><a href="/">Home</a><a href="/upload">Recognize</a></p>
    </div>
    <h1>Handwritten Recognition System</h1>
    <p class="introduction">Handwritten Text Recognition is a technology that is much needed in this world as of
today.
        The digit recognition system is used to recognize the digits from different sources like email, bank cheque,
papers, images, etc.
        Before proper implementation of this technology we have relied on writting texts with our own hands which
can result in errors.
        It's diffcult to store and access physical data with effiecieny.
        The project presents recognizing the handwritten digits (0 to 9) from the famous MNIST dataset.
        Here we will be using artificial neural networks convolution neural networks.
    </p>
    </span>
</body>
</html>
```

PREDICT PAGE

```
<html>
<head>
    <title>Handwritten Digit Recognition System</title>
    <link rel="stylesheet" href="../static/style.css">
    <script src="../static/script.js"></script>
</head>
<body class="style2">
    <span id="loader">
        <img src="../static/loading.gif">
    </span>
    <span id="content">
    <div class="header">
        <p><a href="/">Home</a><a href="/upload">Recognize</a></p>
    </div>
    <h2><strong>Digit Recognition</strong></h2>
    <div class="container">
    <form class="form" action="/predict" method="POST" enctype="multipart/form-data">
    <input type="file" name="file" id="upload" style="display:none"  accept="image/x-png,image/jpeg" required>
    <label for="upload" class="upload">Choose</label>
    <input type='submit' class="recognize" value="Recognize" id="submit">
    <p id="status"></p>
    </form>
    </div></span>
</body>
</html>
```

RESULT PAGE

```
<html>
<head>
    <title>Handwritten Digit Recognition System</title>
    <link rel="stylesheet" href="../static/style.css">
```

```
  <script src="../static/script.js"></script>
</head>
<body class="style2">
  <span id="loader">
    <img src="../static/loading.gif">
  </span>
  <span id="content">
  <div class="header">
    <p><a href="/">Home</a><a href="/upload">Recognize</a></p>
  </div>
  <h2><strong>Digit Recognition</strong></h2>
  <div class="container">
  <span>
  <form class="form" action="/predict" method="POST" enctype="multipart/form-data">
  <input type="file" name="file" id="upload" style="display:none"  accept="image/x-png,image/jpeg" required>
  <label for="upload" class="upload">Choose</label>
  <input type='submit' class="recognize" value="Recognize" id="submit">
  <p id="status"></p>
  </form>
  </span>
  <div><h3>Recognized digit: {{digit}}</h3>
    <h4>Probability: {{percentage}}</h4>
    <img src="{{ user_image }}" alt="User Image" class="user-image"></div>
  </div>
  </span>
</body>
</html>
```

STYLES.CSS

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

.header a{
   color:black;
   font-size: 15pt;
   padding:10px;
   text-decoration:none;
}

.header a:hover{
   border-bottom: 2px black solid;
}

h1, h2, .introduction, .form{
   text-align:center;
}

h1, .introduction{
   color: white;
}

h2{
   color:red;
}

.introduction{
   font-size:15pt;
}

.style1 {
        background: url(../static/images/bg_one.jpeg) no-repeat center center fixed;
        -webkit-background-size: cover;
        -moz-background-size: cover;
        -o-background-size: cover;
        background-size: cover;
}

.style2 {
        background: url(../static/images/bg_two.png) no-repeat center center fixed;
        -webkit-background-size: cover;
        -moz-background-size: cover;
        -o-background-size: cover;
        background-size: cover;
}

.upload{
   background-color:rgb(55,210,180);
```

```
    border-radius:5px;
    cursor:pointer;
    -webkit-border-radius:5px;
    -moz-border-radius:5px;
    -ms-border-radius:5px;
    -o-border-radius:5px;
}

.upload:hover{
    background-color:rgb(0, 136, 109);
}

.recognize{
    background-color:rgb(3,120,255);
    border:none;
    cursor:pointer;
}

.recognize:hover{
    background-color:rgb(0, 72, 155);
}

.upload, .recognize{
    padding:10px;
    color:white;
    border-radius:5px;
    -webkit-border-radius:5px;
    -moz-border-radius:5px;
    -ms-border-radius:5px;
    -o-border-radius:5px;
}

.container{
    display:flex;
    width:100%;
    justify-content:space-around;
}

.user-image{
    width:150px;
    height:150px;
}

#status{
    color:red;
    visibility:hidden;
}
```

SCRIPT.JS

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

GitHub & Project DemoLink

GitHub: https://www.github.com/IBM-EPBL/IBM-Project-11068-1659258981

Project Demo: (Uploaded in GitHub)