PROJECT REPORT

A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

Submitted By

PNT2022TMID37259

Felicia Grace	311619104017
Harshitha K	311619104025
Keerthana S	311619104033
Madhumitha N	311619104038

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Project Overview

The handwritten digit recognition is the ability of computers to recognize human handwritten digits. It is a hard task for the machine because handwritten digits are not perfect and can be made with many different flavors. Handwritten digit recognition using MNIST dataset is a major project made with the help of Neural Network. It basically detects the scanned images of handwritten digits. The MNIST data set is widely used for this recognition process and it has 70000 handwritten digits. We use Artificial neural networks to train these images and build a deep learning model. Web application is created where the user can upload an image of a handwritten digit. This image is analyzed by the model and the detected result is returned to the UI.

1.2 Purpose

Handwritten character recognition is one of the practically important issues in pattern recognition applications. The applications of digit recognition includes in postal mail sorting, bank check processing, form data entry, etc. . As handwritten digits are not the same size, thickness, position and direction, in this case by the way, various difficulties should be considered to find the handwritten digital recognition problem.The main problem lies within the ability on developing an efficient algorithm that can recognize hand written digits, which is submitted by users by the way of a scanner, tablet, and other digital devices.

2. LITERATURE SURVEY

2.1 Existing problem

The purpose of this paper is to recognize and predict the handwritten digits from Zero to Nine using Artificial Intelligence and Neural network concepts. Handwritten digits might not be easy to decipher since every person's handwriting differs with time to time. Real-time application for Handwritten Digit Recognition System is when certain digits written in the cheque paper or an account number is written by the customer or user, the individual who checks the written digits might not be able to recognize the digits. Handwritten digits are identified and analyzed by a model, which is trained and tested with a dataset containing over 7000 different ways of handwritten digits.

These days, a growing number of people are using images to transfer data. In addition it is the main distribution to separate the important data from the images. Image Recognition is an important research area for your most used apps. In general, in the field of pattern recognition, one of the most difficult tasks is the precise computerization of human handwriting. Without a doubt, this is a very difficult subject because there are so many variations of handwriting from person to person. Despite the fact that this difference does not cause problems for humans, however, it is becoming increasingly difficult to instruct computers to interpret common handwriting. In the case of image recognition, for example, classification by hand, it is important to know how the information is displayed in the pictures.

2.2 References

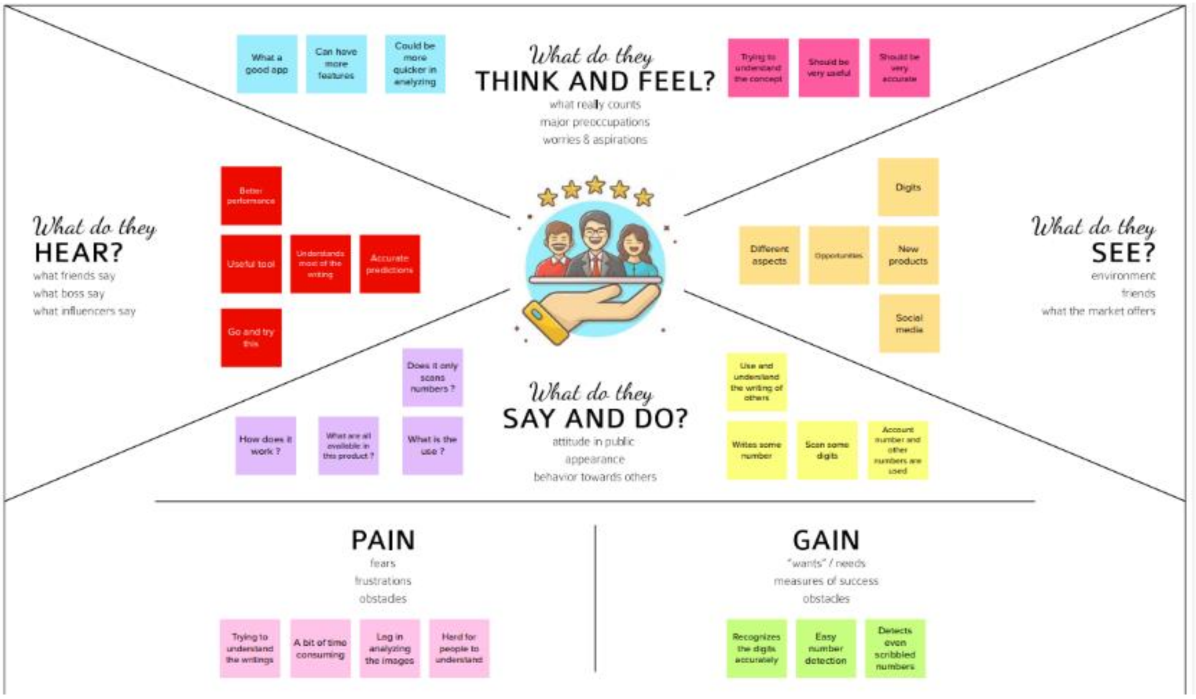
Handwritten Digit Classification Using the MNIST Dataset by M. Wu and Z. Zhang, 2010. Handwritten digit recognition with decoding, A. Dutta and A. Dutta Handwritten digital recognition is utilized in this article. Comprehensive learning techniques have been established. The same data was used to train and evaluate a number of popular machine learning methods, including KNN, SVM, RFC, and CNN, to uncover comparisons across divisions. The more deeply you learn using these techniques, the more accurate you will be. In contrast to previous research techniques, this technique focuses on which category helps construct models with separation accuracy of greater than 99%. When CNN is used as the backend and Tensorflow as the software, it can deliver accuracy of roughly 98.72%. CNN provides 98.72% accuracy in the first test.

2.3 Problem Statement Definition

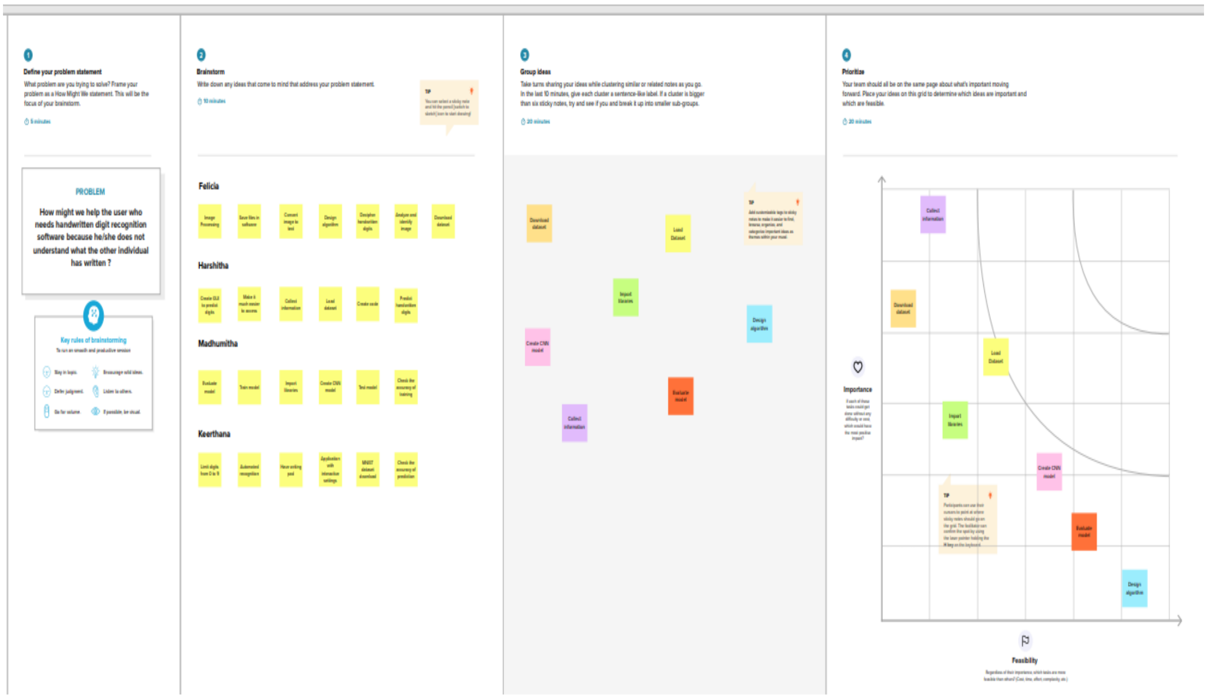
A Novel Method for Handwritten Digit Recognition System How might we help the user who needs handwritten digit recognition software because he/she does not understand what the other individual has written? This software will help them to understand the digits with ease. Every individual will be able to understand the digits that were written, scribbled or even photographed, without any issues by using this software. Additionally, there would be issues with identifying the numbers because of similarities between numerals like 1 and 7, 5 and 6, 3 and 8, 2 and 5, 2 and 7, etc. Finally, the individuality and variation of each individual's handwriting influence the structure and appearance of the digits. Therefore, this project aims to address these problems and provide a novel handwritten digit recognition system.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming



1 **Define your problem statement**

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

5 minutes

PROBLEM

How might we help the user who needs handwritten digit recognition software because he/she does not understand what the other individual has written ?

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

TIP: You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

Felicia	Harshitha	Madhumitha	Keerthana
Image Processing	Create GUI to predict digits	Evaluate model	Limit digits from 0 to 9
Save files in software	Make it much easier to access	Train model	Automated recognition
Convert image to text	Collect information	Import libraries	Have writing pad
Design algorithm	Load dataset	Create CNN model	Application with interactive settings
Decipher handwritten digits	Create code	Test model	MNIST dataset download
Analyze and identify image	Predict handwritten digits	Check the accuracy of training	Check the accuracy of prediction
Download dataset			

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

TIP: Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

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

TIP: Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the H key on the keyboard.

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

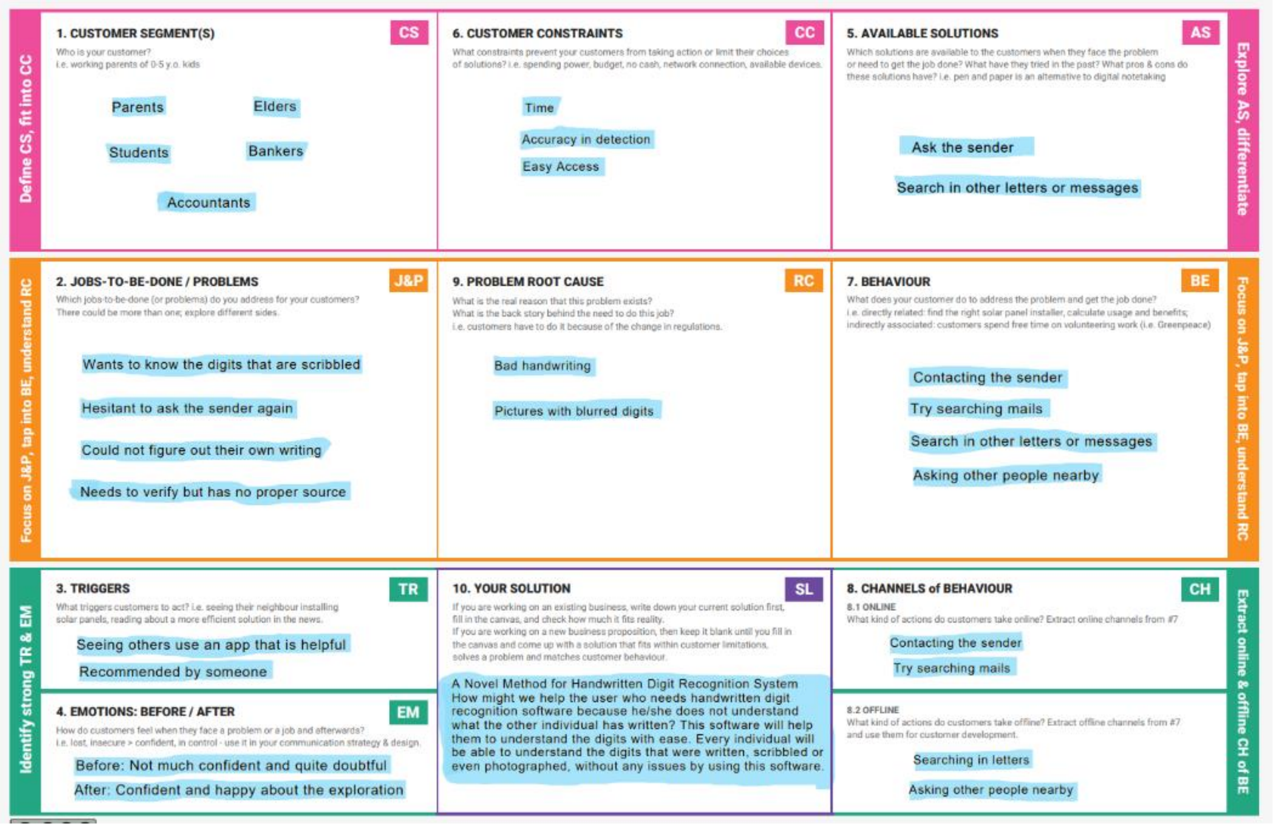
Feasibility: Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	How might we help the user who needs handwritten digit recognition software because he/she does not understand what the other individual has written? This software will help them to understand the digits with ease.
2.	Idea / Solution description	To recognize and predict the handwritten digits from Zero to Nine using Artificial Intelligence and Neural network concepts. Handwritten digits are identified and analyzed by a model, which is trained and tested with a dataset containing over 7000 different ways of handwritten digits.
3.	Novelty / Uniqueness	Create GUI with writing pad to predict the digits and save them as files.

4.	Social Impact / Customer Satisfaction	Recognizes the digits accurately, Easy number detection, Detects even scribbled numbers.
5.	Business Model (Revenue Model)	Activities: • Download Database • Train model • Test model • Create GUI Key resources: • Python • Anaconda • Flask • Cloud
6.	Scalability of the Solution	Less maintenance cost since it has less software requirements. It would be very helpful in the industry if it is developed and put into action.

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

Parents

Elders

Students

Bankers

Accountants

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.

Time

Accuracy in detection

Easy Access

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

Ask the sender

Search in other letters or messages

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.

Wants to know the digits that are scribbled

Hesitant to ask the sender again

Could not figure out their own writing

Needs to verify but has no proper source

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.

Bad handwriting

Pictures with blurred digits

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)

Contacting the sender

Try searching mails

Search in other letters or messages

Asking other people nearby

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.

Seeing others use an app that is helpful

Recommended by someone

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.

Before: Not much confident and quite doubtful

After: Confident and happy about the exploration

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.

A Novel Method for Handwritten Digit Recognition System
How might we help the user who needs handwritten digit recognition software because he/she does not understand what the other individual has written? This software will help them to understand the digits with ease. Every individual will be able to understand the digits that were written, scribbled or even photographed, without any issues by using this software.

8. CHANNELS of BEHAVIOUR CH

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

Contacting the sender

Try searching mails

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

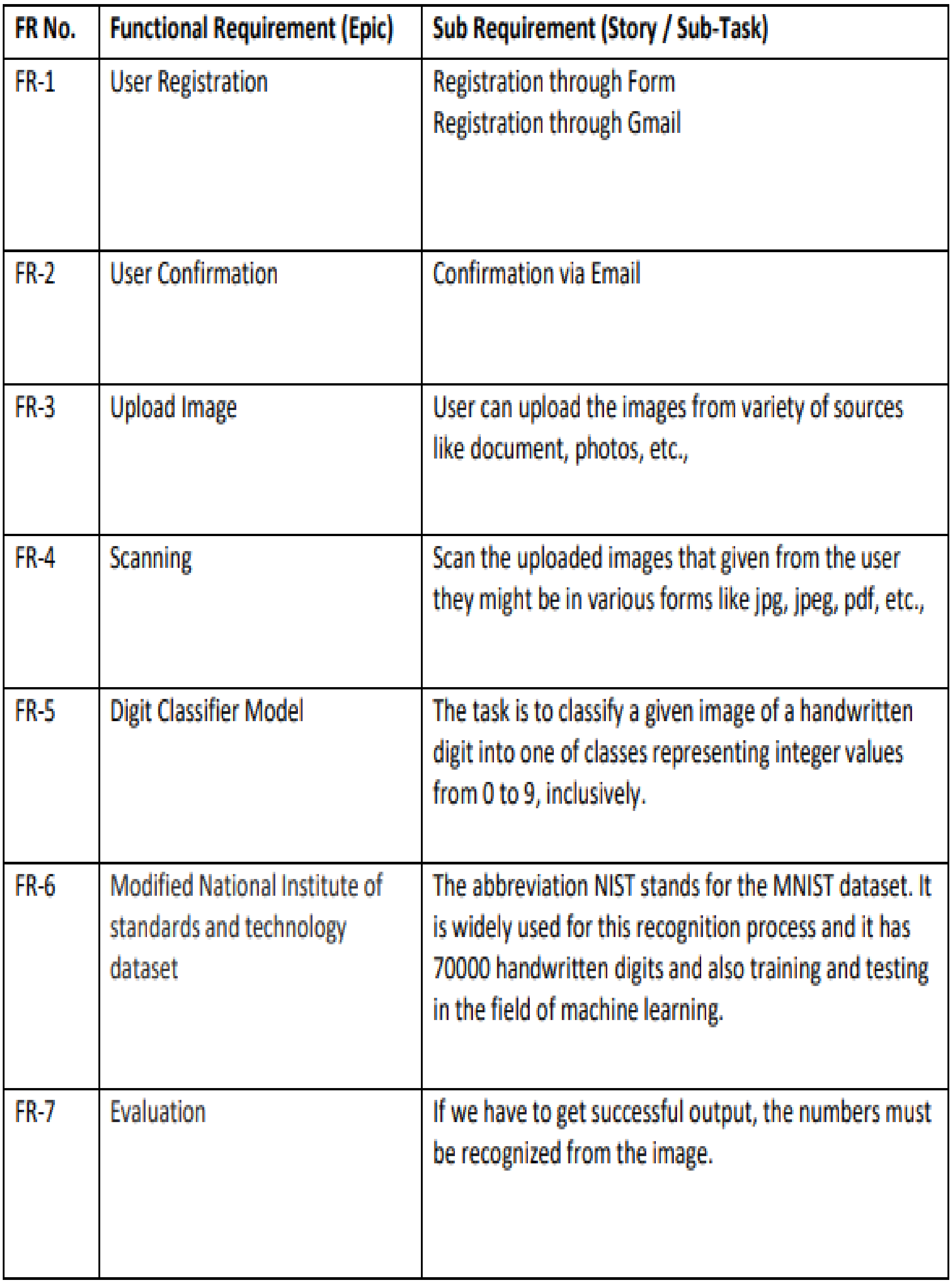
Searching in letters

Asking other people nearby

Extract online & offline CH of BE

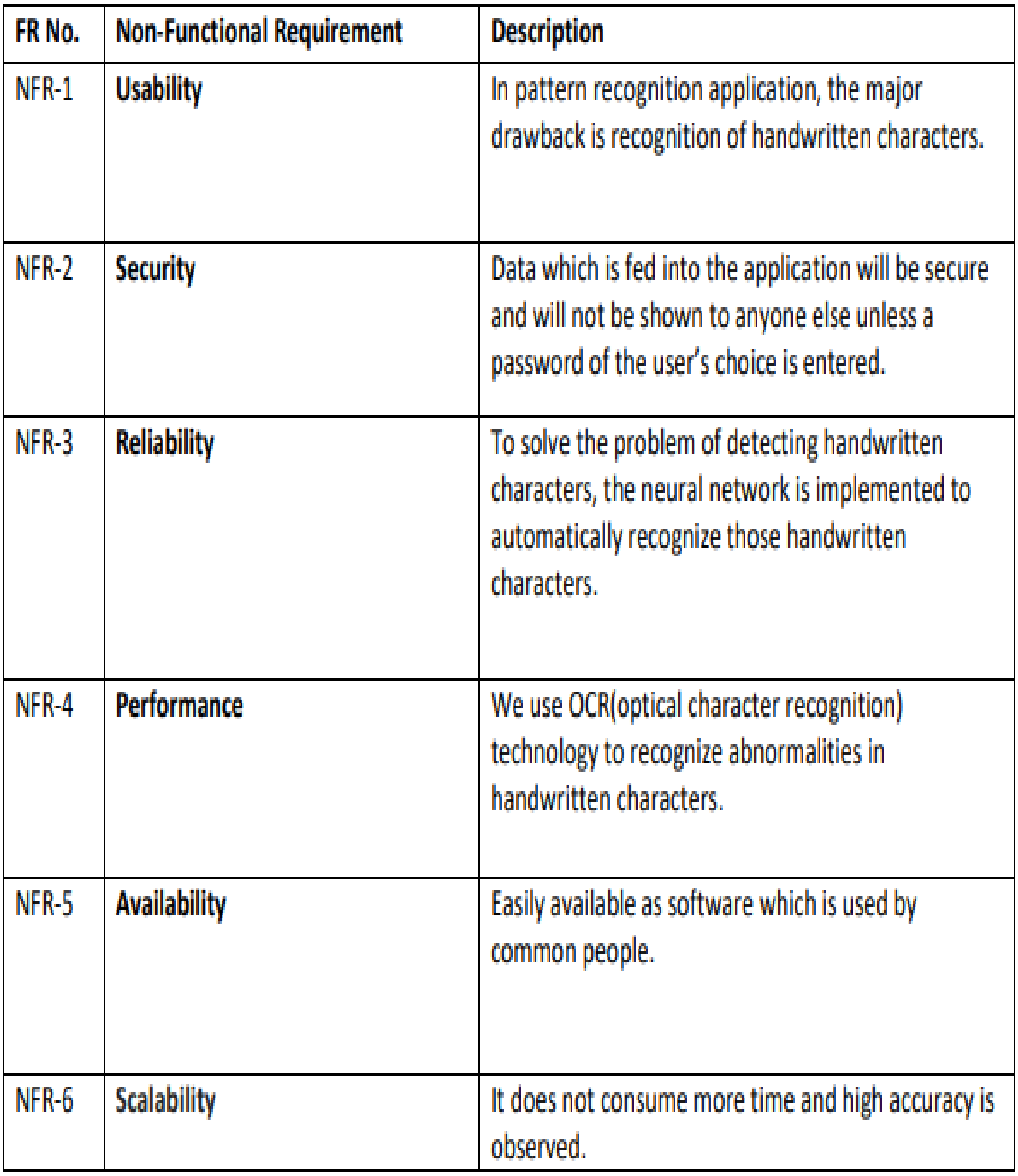
4. REQUIREMENT ANALYSIS

4.1 Functional Requirements



FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail
FR-2	User Confirmation	Confirmation via Email
FR-3	Upload Image	User can upload the images from variety of sources like document, photos, etc.,
FR-4	Scanning	Scan the uploaded images that given from the user they might be in various forms like jpg, jpeg, pdf, etc.,
FR-5	Digit Classifier Model	The task is to classify a given image of a handwritten digit into one of classes representing integer values from 0 to 9, inclusively.
FR-6	Modified National Institute of standards and technology dataset	The abbreviation NIST stands for the MNIST dataset. It is widely used for this recognition process and it has 70000 handwritten digits and also training and testing in the field of machine learning.
FR-7	Evaluation	If we have to get successful output, the numbers must be recognized from the image.

4.2 Non Functional Requirements

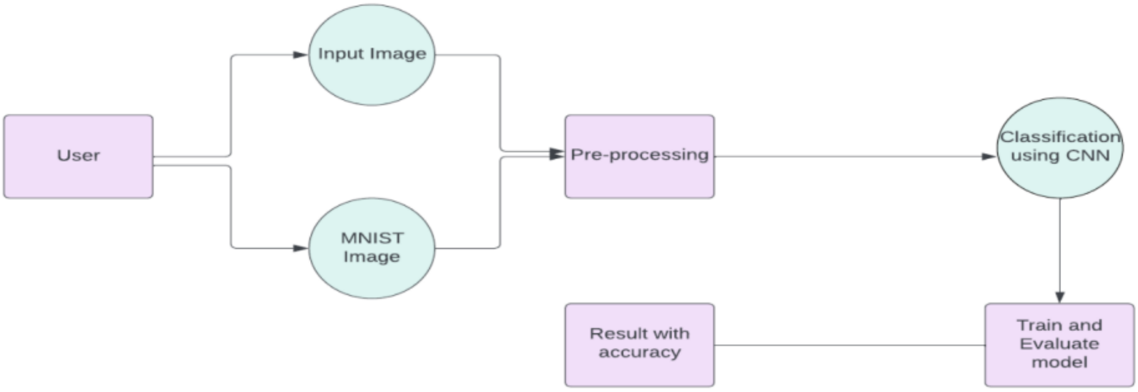


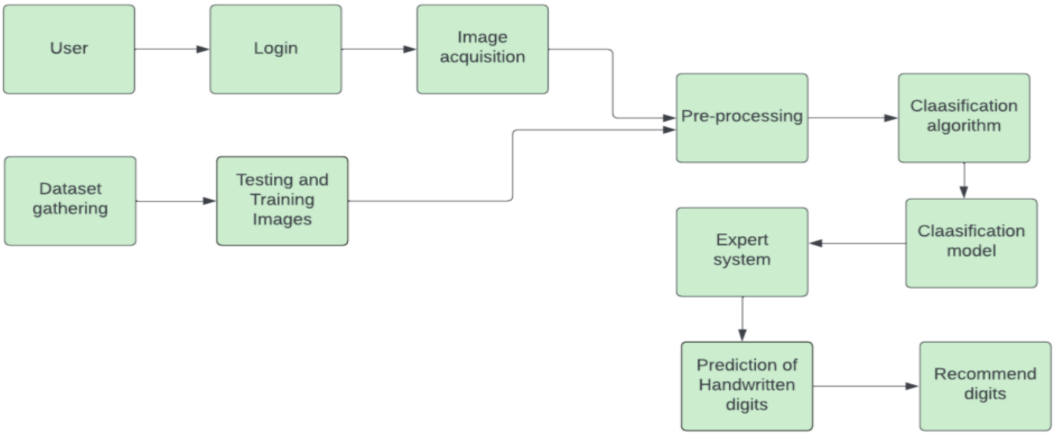
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	In pattern recognition application, the major drawback is recognition of handwritten characters.
NFR-2	**Security**	Data which is fed into the application will be secure and will not be shown to anyone else unless a password of the user's choice is entered.
NFR-3	**Reliability**	To solve the problem of detecting handwritten characters, the neural network is implemented to automatically recognize those handwritten characters.
NFR-4	**Performance**	We use OCR(optical character recognition) technology to recognize abnormalities in handwritten characters.
NFR-5	**Availability**	Easily available as software which is used by common people.
NFR-6	**Scalability**	It does not consume more time and high accuracy is observed.

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.





5.2 Solution & Technical Architecture

MNIST Dataset Description

Because everyone in the world has a unique writing style, handwriting identification is one of the fascinating research projects now being conducted. It is the ability of a computer to automatically recognise and comprehend handwritten numbers or letters. Every aspect of life is being digitized to lessen the need for human labor as a result of advancements in science and technology. Thus, handwritten digit recognition is required in many real-time applications. The MNIST data collection, which contains 70000 handwritten digits, is frequently utilized for this recognition method. In order to train these photos and create a deep learning model, we use artificial neural networks. A web application is developed that allows users to upload pictures of handwritten numbers. The model examines this picture.

The 60,000 training and 10,000 testing labeled handwritten digit images in the MNIST Handwritten Digit Recognition Dataset.

Each image has a total of 784 (2828) pixels, or 28 pixels in height and 28 pixels in width. There is just one pixel value assigned to each pixel. It displays the brightness or darkness of that pixel (larger numbers indicate darker pixel). The integer for this pixel value ranges from 0 to 255.

Label: 5
Label: 0
Label: 4
Label: 1
Label: 9
Label: 2
Label: 1
Label: 3
Label: 1
Label: 4
Label: 3
Label: 5
Label: 3
Label: 6
Label: 1
Label: 7
Label: 2
Label: 8
Label: 6
Label: 9

PROCEDURE

- Install the TensorFlow library.
- Prepare the model's dataset.
- For the purpose of classifying the handwritten digits, create a single layer perceptron model.
- Plot the accuracy change over time.
- Analyze the test data to evaluate the model.
- Speculate on the model summary.
- To make the model a multi-layer perceptron, add a hidden layer.
- To avoid overfitting and assess its impact on accuracy, include Dropout.
- Increase the number of hidden layer neurons and assess how accuracy is affected.
- Test the impact of various optimizers on accuracy.
- Increase the hidden layers and assess the accuracy impact.
- Change the batch size and epochs, then assess the impact on accuracy.

For the recognition of handwritten digits, the MNIST dataset is frequently used. 10,000 test photos make up the dataset, which includes 60,000 training images. The discipline of image processing relies heavily on artificial neural networks because they most closely resemble the human brain.

A significant project done with the use of neural networks is the recognition of handwritten digits using the MNIST dataset. In essence, it recognizes digits that were scribbled and scanned.

Our handwritten digit identification system goes a step further in that it can now recognize handwritten numbers typed directly on the screen with the aid of an integrated GUI in addition to detecting them in scanned photos.

APPROACH

This project will be approached utilizing a three-layered neural network.

- The input layer: The input layer transfers the information from our example systems to the following layer so that the latter can compute its activations.
- The hidden layer: The network's nonlinear ties are provided by hidden units termed activations that make up the hidden layer. Depending on our needs, there can be a variety of concealed layers.
- The output layer: The nodes in this stratum are referred to as output units. It gives us access to the neural network's final prediction, which may be used to make final predictions.

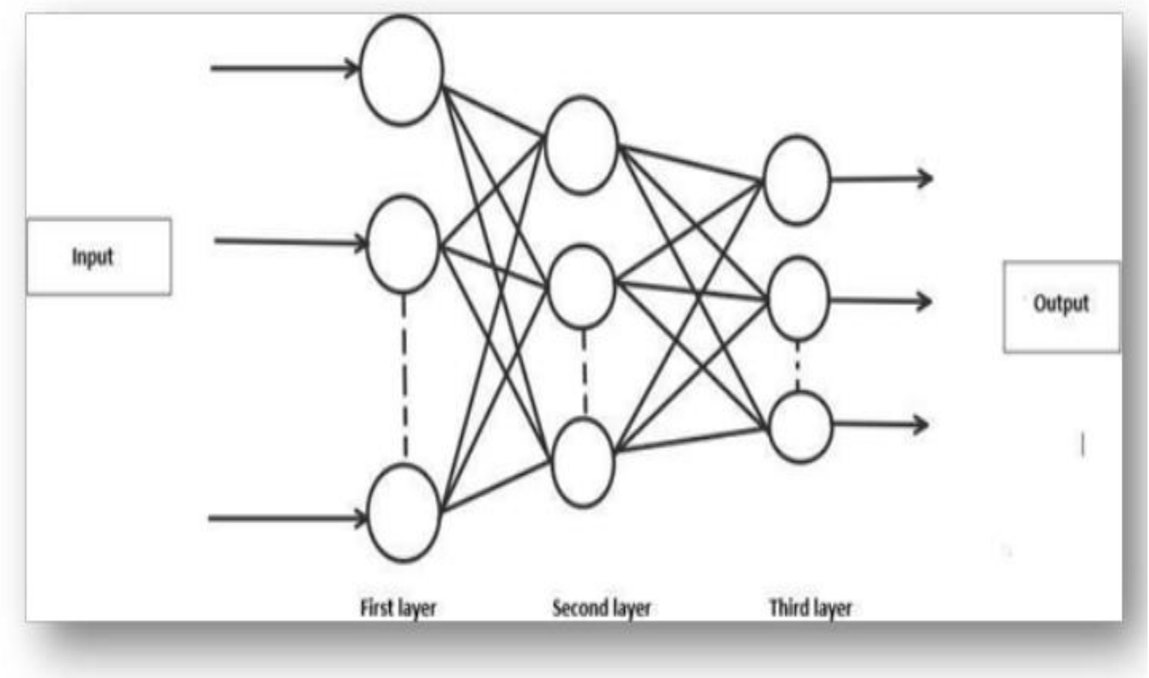
A neural network is a model of the brain's operations. It is made up of numerous layers with a variety of activations; these activations mimic the neurons in our brain. An attempt is made by a neural network to learn a set of parameters from a set of data that might aid in understanding the underlying relationships. Since neural networks are capable of adapting to changing input, the network can produce the best outcome without having to change the output criterion.

METHODOLOGY

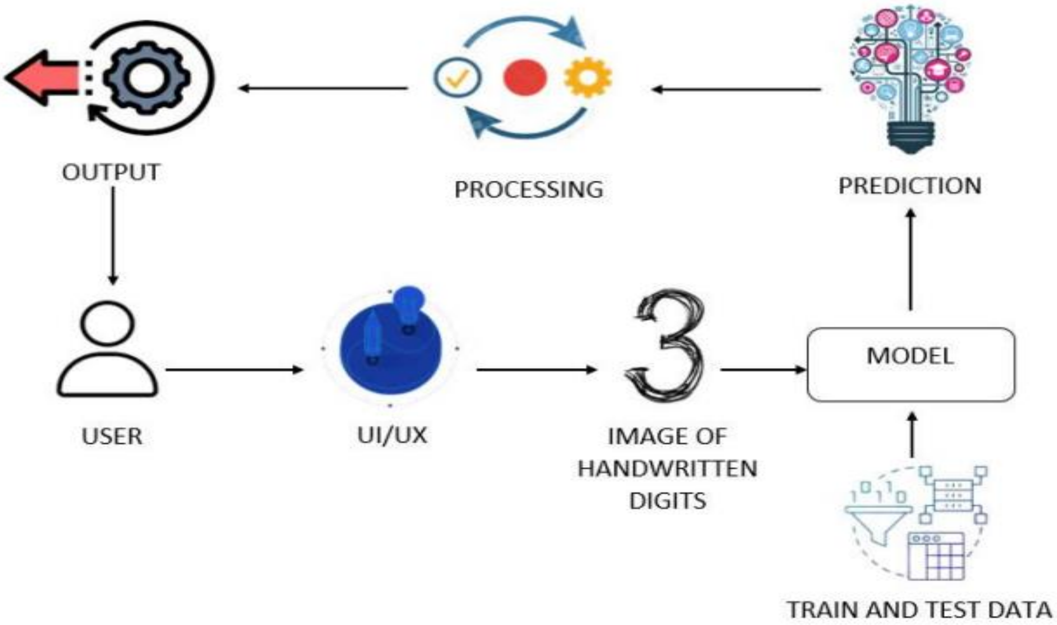
A neural network with one hidden layer and 100 activation units has been put into practice (excluding bias units). The features (X) and labels (Y) were retrieved after the data was loaded from a.mat file.

To prevent overflow during computation, features are then scaled into a range of [0,1] by dividing by 255. 10,000 testing cases and 60,000 training examples make up the data. With the training data, feedforward is used to calculate the hypothesis, and backpropagation is then used to lower the error between the layers.

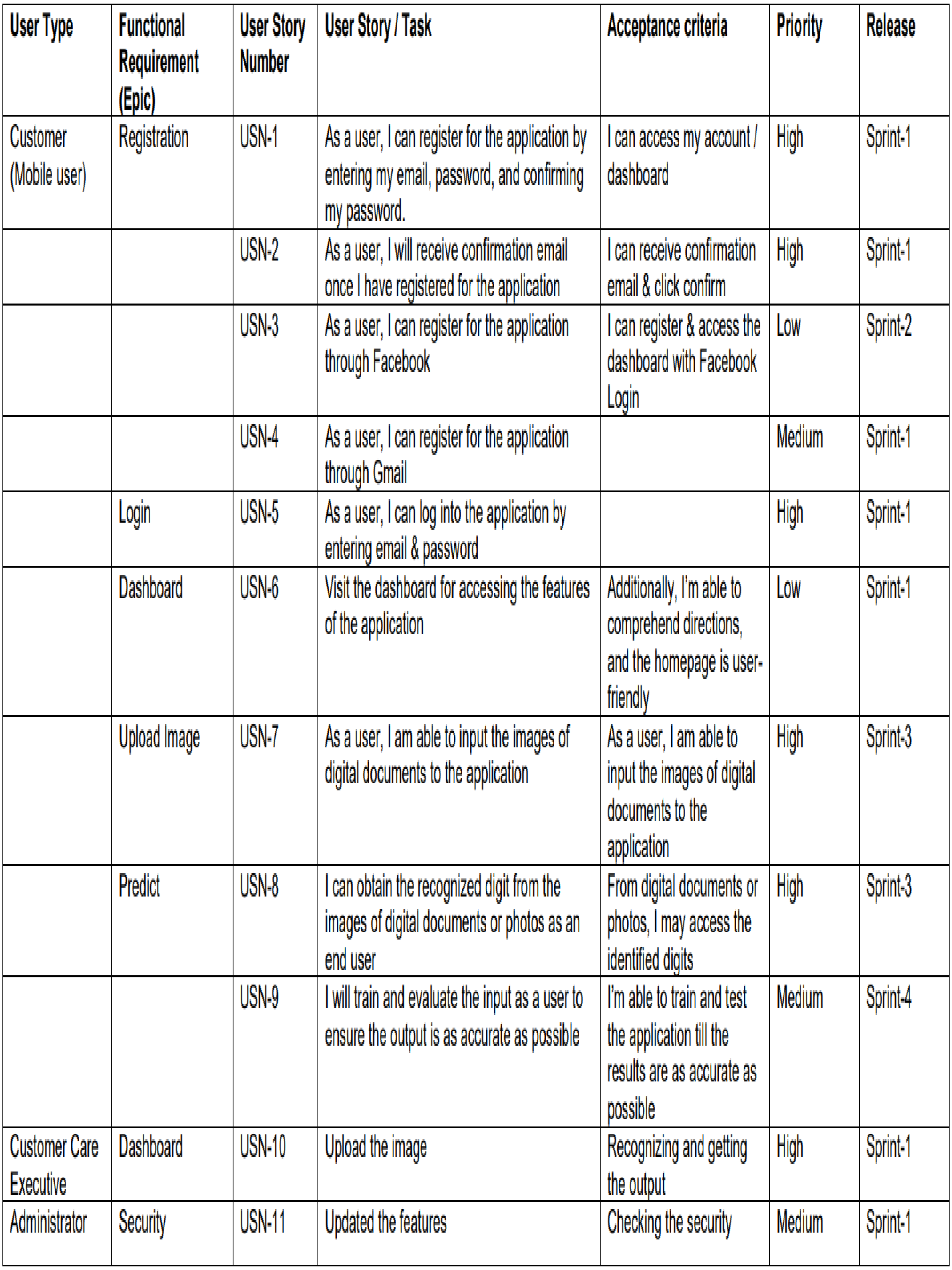
To combat overfitting, the regularization parameter lambda is set to 0.1. To identify the model that fits the situation the optimizer runs for 70 times.



TECHNICALARCHITECTURE



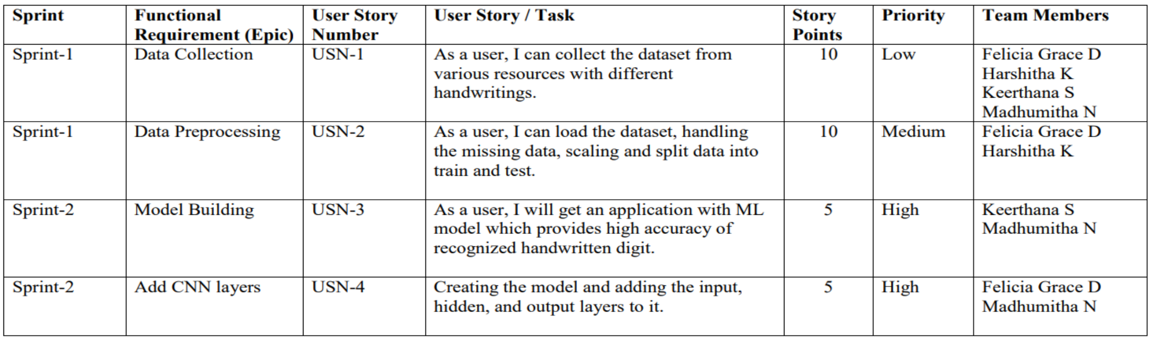
5.3User Stories



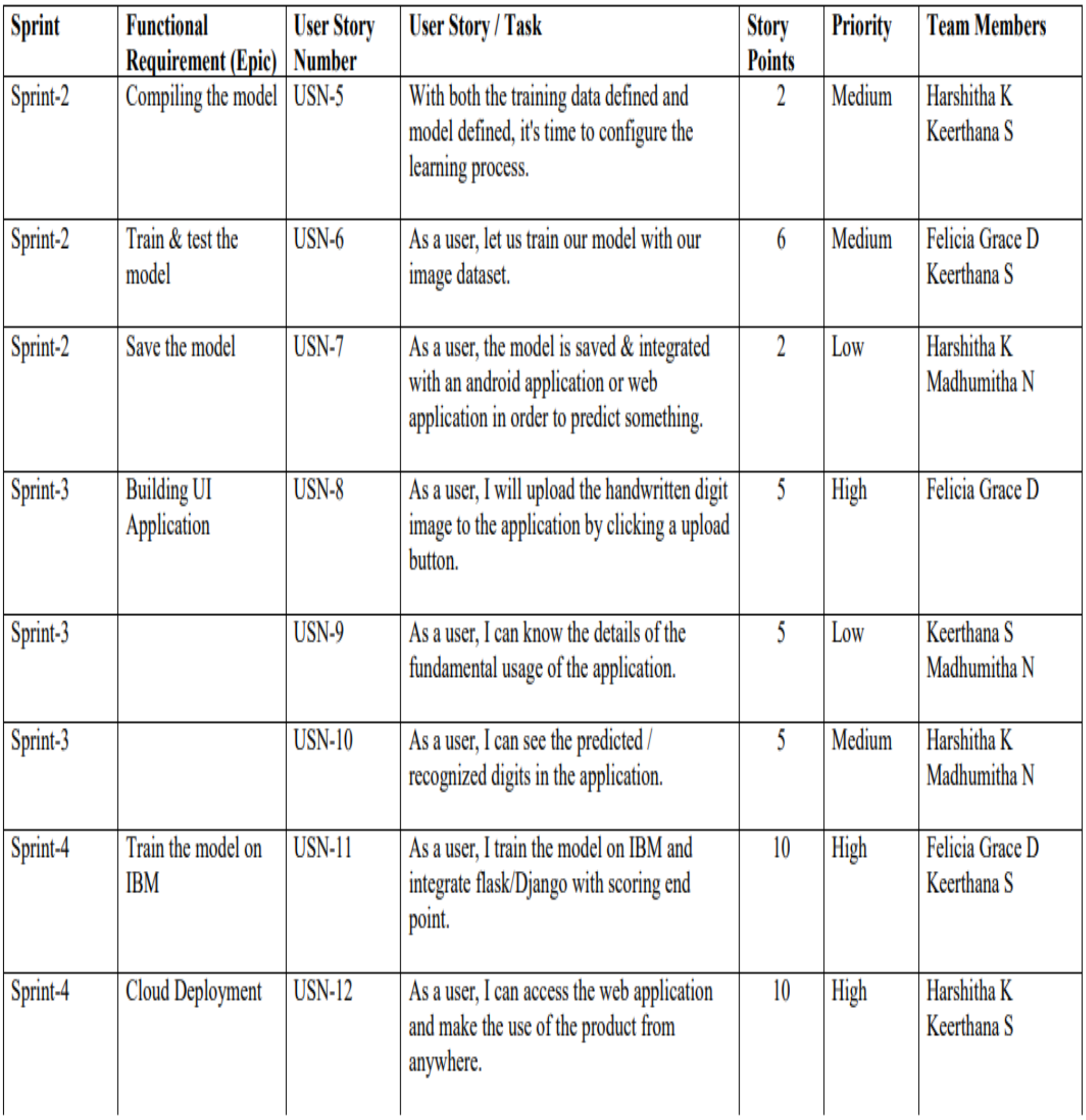
User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	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		Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password		High	Sprint-1
	Dashboard	USN-6	Visit the dashboard for accessing the features of the application	Additionally, I'm able to comprehend directions, and the homepage is user-friendly	Low	Sprint-1
	Upload Image	USN-7	As a user, I am able to input the images of digital documents to the application	As a user, I am able to input the images of digital documents to the application	High	Sprint-3
	Predict	USN-8	I can obtain the recognized digit from the images of digital documents or photos as an end user	From digital documents or photos, I may access the identified digits	High	Sprint-3
		USN-9	I will train and evaluate the input as a user to ensure the output is as accurate as possible	I'm able to train and test the application till the results are as accurate as possible	Medium	Sprint-4
Customer Care Executive	Dashboard	USN-10	Upload the image	Recognizing and getting the output	High	Sprint-1
Administrator	Security	USN-11	Updated the features	Checking the security	Medium	Sprint-1

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint planning and estimation

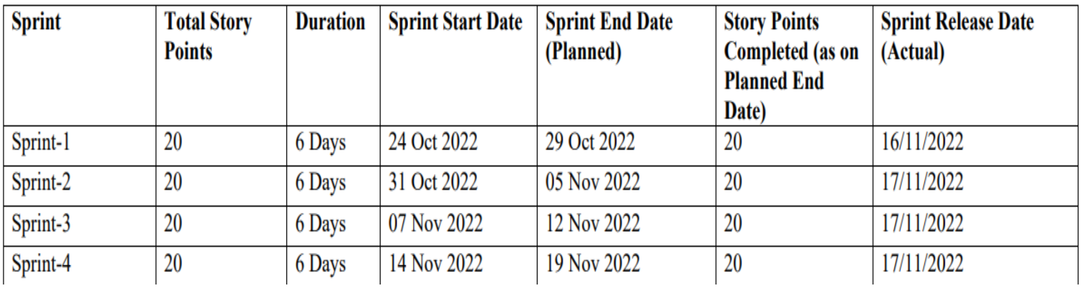


Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data Collection	USN-1	As a user, I can collect the dataset from various resources with different handwritings.	10	Low	Felicia Grace D Harshitha K Keerthana S Madhumitha N
Sprint-1	Data 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	Felicia Grace D Harshitha K
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	Keerthana S Madhumitha N
Sprint-2	Add CNN layers	USN-4	Creating the model and adding the input, hidden, and output layers to it.	5	High	Felicia Grace D Madhumitha N



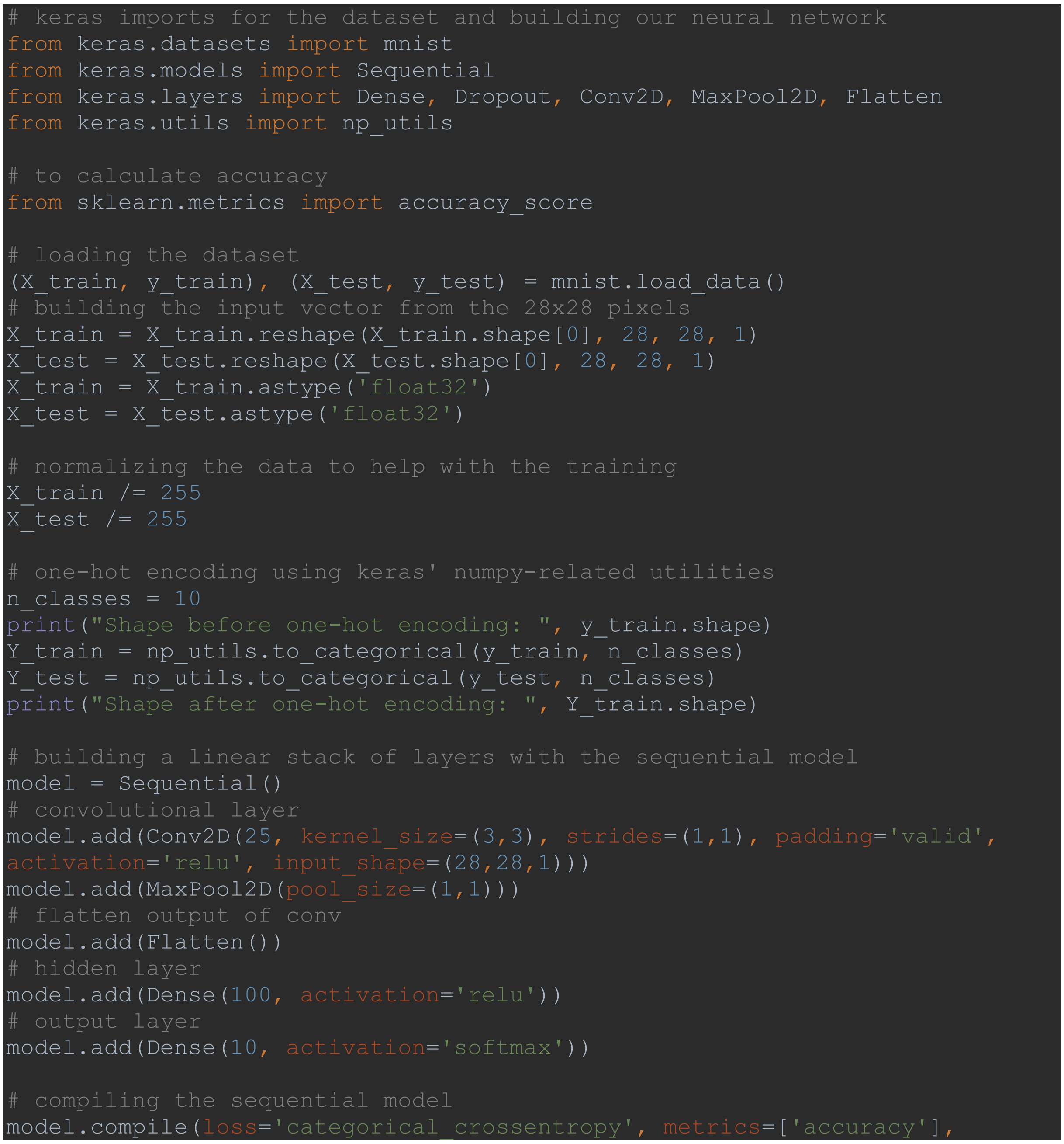
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-2	Compiling the model	USN-5	With both the training data defined and model defined, it's time to configure the learning process.	2	Medium	Harshitha K Keerthana S
Sprint-2	Train & test the model	USN-6	As a user, let us train our model with our image dataset.	6	Medium	Felicia Grace D Keerthana S
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	Harshitha K Madhumitha N
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.	5	High	Felicia Grace D
Sprint-3		USN-9	As a user, I can know the details of the fundamental usage of the application.	5	Low	Keerthana S Madhumitha N
Sprint-3		USN-10	As a user, I can see the predicted / recognized digits in the application.	5	Medium	Harshitha K Madhumitha N
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	Felicia Grace D Keerthana S
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	Harshitha K Keerthana 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	16/11/2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	17/11/2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	17/11/2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	17/11/2022

7. Coding and Solutioning:



```
# keras imports for the dataset and building our neural network
from keras.datasets import mnist
from keras.models import Sequential
from keras.layers import Dense, Dropout, Conv2D, MaxPool2D, Flatten
from keras.utils import np_utils

# to calculate accuracy
from sklearn.metrics import accuracy_score

# loading the dataset
(X_train, y_train), (X_test, y_test) = mnist.load_data()
# building the input vector from the 28x28 pixels
X_train = X_train.reshape(X_train.shape[0], 28, 28, 1)
X_test = X_test.reshape(X_test.shape[0], 28, 28, 1)
X_train = X_train.astype('float32')
X_test = X_test.astype('float32')

# normalizing the data to help with the training
X_train /= 255
X_test /= 255

# one-hot encoding using keras' numpy-related utilities
n_classes = 10
print("Shape before one-hot encoding: ", y_train.shape)
Y_train = np_utils.to_categorical(y_train, n_classes)
Y_test = np_utils.to_categorical(y_test, n_classes)
print("Shape after one-hot encoding: ", Y_train.shape)

# building a linear stack of layers with the sequential model
model = Sequential()
# convolutional layer
model.add(Conv2D(25, kernel_size=(3,3), strides=(1,1), padding='valid',
activation='relu', input_shape=(28,28,1)))
model.add(MaxPool2D(pool_size=(1,1)))
# flatten output of conv
model.add(Flatten())
# hidden layer
model.add(Dense(100, activation='relu'))
# output layer
model.add(Dense(10, activation='softmax'))

# compiling the sequential model
model.compile(loss='categorical_crossentropy', metrics=['accuracy'],
optimizer='adam')

# training the model for 10 epochs
model.fit(X_train, Y_train, batch_size=128, epochs=5,
validation_data=(X_test, Y_test))

# Save the model
model.save('mnistCNN.h5')
```

8. ADVANTAGES & DISADVANTAGES

Advantages

1) 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.

2) The generative models can perform recognition driven segmentation.

3) Better readability of digits from different handwriting.

Disadvantages

1) It requires much more computation than more standard OCR techniques.

2) It is not done in real time as a person writes and therefore not appropriate for immediate text input.

9. CONCLUSION

There are numerous uses for handwritten digit recognition in the fields of medicine, banking, student administration, taxation, etc. To extract the digit from the handwritten image, a variety of classifiers including KNN, SVM, and CNN are employed. According to the evaluation, CNN performs better than the competition. This study discusses the stages of HDR using a CNN classifier. The MNIST dataset is a common dataset used to evaluate the performance of classifiers. It consists of handwritten numbers from 0 to 9. Three separate stages make up HDR. The first step is preprocessing, which involves converting the dataset into binary format and applying image processing on it. Segmentation, the second stage, involves dividing the image into several pieces. The third stage is feature extraction, during which image features are found. CNN is utilised in the classification stage, which comes last. The CNN classifier greatly enhances the outcomes of HDR, but it is still possible to further enhance the complexity, execution time, and accuracy of the results by combining classifiers or utilising other algorithms in addition to CNN.

10. FUTURE SCOPE

There is still much work to be done on this project, and it may use a lot of improvement. The following are a few ways this project could be improved:

- Add the ability to save the results of multiple image detection from digits.
- Adding capability to recognise multiple digits

- To detect numbers from complicated images, improve the model.

- Adding support for additional languages will benefit users worldwide. This undertaking has limitless potential and may constantly be improved. By putting this idea into practise in the real world, numerous sectors will gain, many workers' workloads will be reduced, and overall work efficiency will increase.

11. APPENDIX

SOURCE CODE

1. app.py

```
import numpy as np
import os
from PIL import Image
from flask import Flask, request, render_template, url_for
from werkzeug.utils import secure_filename, redirect
from gevent.pywsgi import WSGIServer
from keras.models import load_model
from keras.preprocessing import image
from flask import send_from_directory

UPLOAD_FOLDER = 'C:/Users/messysaw/Desktop/handwritten-digit-recognition/uploads'


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

model = load_model("models/mnistCNN.h5")


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


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

        upload_img = os.path.join(UPLOAD_FOLDER, filepath)
        img = Image.open(upload_img).convert("L")  # convert image to monochrome
        img = img.resize((28, 28))  # resizing of input image
        im2arr = np.array(img)  # converting to image
        im2arr = im2arr.reshape(1, 28, 28, 1)  # reshaping according to our requirement

        pred = model.predict(im2arr)

        num = np.argmax(pred, axis=1)  # printing our Labels

        return render_template('predict.html', num=str(num[0]))


if __name__ == '__main__':
    app.run(debug=True, threaded=False)
```

2. index.html

```
<html>

<head>
  <title>Digit Recognition Website</title>

    <meta charset="UTF-8">
    <title>Upload image</title>

    <meta name="viewport" content="width=device-width">

</head>

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

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

</script>

<style>

     #clear_button{
  margin-left: 15px;
  font-weight: bold;
  color: green;
}
```

```
#confidence{
  font-family: 'Josefin Sans', sans-serif;
  margin-top: 7.5%;
}

#content{
  margin: 0 auto;
  padding: 2% 15%;
  padding-bottom: 0;
}

#predict_button{
  margin-right: 15px;
  color: green;
  font-weight: bold;
}

#prediction_heading{
  font-family: 'Josefin Sans', sans-serif;
  margin-top: 7.5%;
}

#result{
  font-size: 5rem;
}

.btn {
    font-size: 15px;
    padding: 10px;
    -webkit-appearance: none;
    background: #eee;
    border: 1px solid #888;
    margin-top: 20px;
    margin-bottom: 20px;
}

.buttons_div{
  margin-bottom: 30px;
  margin-right: 80px;
}

.leftside{
  text-align: center;
  margin: 0 auto;
  margin-top: 2%;
  /* padding-left: 10%; */
}
```

```
#frame{
  margin-right: 10%;
}

.predicted_answer{
  text-align: center;
  margin: 0 auto;
  padding: 3% 5%;
  padding-top: 0;
  /* padding-left: 10%; */
}

@media (min-width: 720px) {
  .leftside{
    padding-left: 10%;
  }
}

    .welcome{
   text-align: center;
   position: relative;
   color: #33d6ff;
   background-color: #ffe6ff;
   padding-top: 1%;
   padding-bottom: 1%;
   font-weight: bold;
   font-family: sans-serif;
}

#team_id{
   text-align: right;
   font-size: 25px;
   padding-right: 3%;
}

p{
  font-family: 'Source Code Pro', monospace,sans-serif;
  font-weight: bold;
  color: #00ffcc;
  margin-top: 1%;
}

.heading{
  font-family: 'Varela Round', sans-serif;
  font-style: italic;
  color: #ff4d4d;
  font-weight: 700;
```

```
 font-size: 2rem;
 display: inline;
}

#title{
 padding: 1.5% 15%;
 margin: 0 auto;
 text-align: center;
}

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

<body>

 <h1 class="welcome">IBM PROJECT
 <div id="team_id">TEAM ID : PNT2022TMID37259</div>
 </h1>
 <section id="title">
   <h4 class="heading">HANDWRITTEN DIGIT RECOGNITION</h4>
   <br><br>
    <p>
      The website is designed to predict the handwritten digit.
    </p>
   <p>
      A Novel Method for Handwritten Digit Recognition System How might we help the user who needs
      handwritten digit recognition software because he/she does not understand what the other
      individual has written? This software will help them to understand the digits with ease. Every
      individual will be able to understand the digits that were written, scribbled or even
      photographed, without any issues by using this software.</p>

    <br>

 </section>

 <section id="content">

      <div class="leftside">
      <form action="/predict" method="POST" enctype="multipart/form-data">
      <label >Select an image:</label>
    <input id="image" type="file" name="image" accept="image/png, image/jpeg"
onchange="preview()"><br><br>
     <img id="frame" src="" width="100px" height="100px"/>
     <div class="buttons_div">
      <button type="submit" class="btn btn-dark" id="predict_button">Predict</button>
      <button type="button" class="btn btn-dark" id="clear_button">&nbsp Clear
&nbsp</button>
     </div>
    </form>
    </div>
  </section>

</body>

</html>
```

3. predict.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Prediction</title>
</head>

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

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

  #ans{
 text-align: center;
 font-size: 40px;
 margin: 0 auto;
 padding: 3% 5%;
 font-family: Arial;
 padding-top: 7%;
 color: white;
 }

#num{
    font-size: 50px;
  }

</style>
<body>
  <div id="rectangle">
    <h1 id="ans">Predicted Number <br><center id="num">{{num}}</center></h1>
  </div>
</body>
</html>
```

Github and Project demo link

Github link: https://github.com/IBM-EPBL/IBM-Project-3929-1658672553

Project demo link:

https://drive.google.com/file/d/1kQk3FiAKiFXau_dFjuVCL70FtRwVTeDu/view?usp=sharing