PROJECT REPORT

A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION

submitted by

PNT2022TMID53246

Rajeev Krishna	-	SSNCE195002086
Priyadharshan R	-	SSNCE195002085
Maanas Karthikeyan	-	SSNCE195002069
Sai Prakash. R	-	SSNCE195002098

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

A novel method for handwritten digit recognition with neural networks

Abstract:

In today's society, character recognition is becoming increasingly vital. It facilitates human work and aids with the resolution of more difficult issues. One illustration ishandwritten character recognition, which is extensively used worldwide. This technique was created to recognise zip codes or postal codes for use in mail sorting. This can aid people in the difficult-to-read postal code mail sorting process. Researchers have been working on handwriting recognition for more than thirty years. The number of firms participating in handwriting recognition research has steadily expanded over the last several years. Handwriting processing has advanced due to a mix of factors such as improved recognition rates and the usageof complicated systems.

We can enter our handwriting into some handwriting recognition systems. Eitherusing amouse or a third-party drawing tablet, you can accomplish this. We have the option of typing the input or leaving it as an "ink object" with our own handwriting. Additionally, we can manually type the content into any Microsoft Office software file that we want the system to identify. Typing 1s and 0s will allow us to do this. As a Boolean variable, this operates.

CHAPTER 2

LITERATURE SURVEY

Paper 1: Novel Deep Neural Network Model for Handwritten Digit Classification and Recognition

Year: 2021
Authors: Ayush Kumar Agrawal and Vineet Kumar Awasthi

An artificial neural network has one hidden layer between the input and output layers, whereas a deep neural network has numerous hidden layers with input and output layers. Deep neural networks use several hidden layers to increase model performance and achieve higher accuracy compared to accuracy of machine learning models. Most researchers do their research in the area of pattern recognition. In the field of pattern recognition, there are many patterns that can be used, including handwritten numbers, characters, pictures, faces, sounds, and speech. This study focuses on the classification and recognition of handwritten digits.1000 were utilized as test samples and 1000 were training samples.10000 picture samples make up the USPS dataset, of which 7291 serve as training samples and 2007 serve as testing samples. We've used the proposed deep neural network technique in this paper to classify and identify data from the ARDIS and USPS datasets. The suggested model consists of six layers with softmax and relu activation functions. After model implementation, accuracy for ARDIS samples reached 98.70% testing and 99.76% training, which is greater than accuracy from prior research. Additionally, using the USPS samples dataset, 98.22% training accuracy and 93.01% testing accuracy were attained. When compared to earlier methodologies, the data show that deep neural networks perform incredibly well.

Paper 2: A Novel Handwritten Digit Classification SystemBased on Convolutional Neural Network Approach

Year: 2021
Authors: Ali Abdullah Yahya, Jieqing Tan, Min Hu

There have been a ton of CNN classification algorithms put forth in the literature. However, these algorithms do not take into account the proper filter size selection, data preparation, dataset restrictions, or noise. As a result, few algorithms have been able to significantly increase classification accuracy. The paper makes the following contributions to solve these methods' drawbacks: First, the size of the effective receptive field (ERF)
is determined after taking domain knowledge into account. They choose a typical filter size with the aid of the ERF calculation, improving the classification accuracy of our CNN. Second, excessive data produces inaccurate results, which has a detrimental impact on classification accuracy. Before carrying out the data classification task, data preparation is conducted to ensure that the dataset is devoid of any redundant or irrelevant variables to the goal variable. Thirdly, data augmentation has been suggested as a way to reduce training and validation errors and prevent dataset limitations. Fourthly, the paper suggests adding an additive white Gaussian noise with a threshold of 0.5 to the MNIST dataset in order to imitate the natural factors that can affect image quality in the real world. With a recognition accuracy of 99.98% and 99.40% with 50% noise, our CNN algorithm achieves state-of-the-art performance in handwritten digit recognition.

Paper 3: Handwritten Character Recognition using Neural Network and TensorFlow

Year : 2019
Authors : Megha Agarwal, Shalika, Vinam Tomar, Priyanka Gupta

The offline handwritten character recognition in this study will be carried out using Tensorflow and a convolutional neural network. a process known as using SoftMax Regression, one may assign probabilities to one of the many characters in the handwritten text that offers the range of values from 0 to 1, summed to 1. The objective is to create software that is extremely accurate and that has a minimum level of spatial and temporal complexity.

It was determined that strategies for feature extraction like diagonal and direction are significantly better at producing high accuracy.

Outcomes in comparison to other conventional vertical and horizontal techniques moreover use the best
Neural network tried layers provides the benefit of a higher accurate outcome by having a high noise tolerance.

The feed forward model in neural networks is the back-propagation algorithm that was primarily used to classify the characters, recognise them, and receive training continually more.

In addition to these, normalizing along with feature extraction, the results were better and more effective. Character recognition is the outcome of accuracy.

The paper will describe the best approach to get more than 90% accuracy in the field of Handwritten Character Recognition (HCR).

Paper 4: Improved Handwritten Digit Recognition Using Convolutional Neural Networks (CNN)

Year : 2020
Authors: Savita Ahlawat , Amit Choudhary , Anand Nayyar , Saurabh Singh and Byungun Yoon

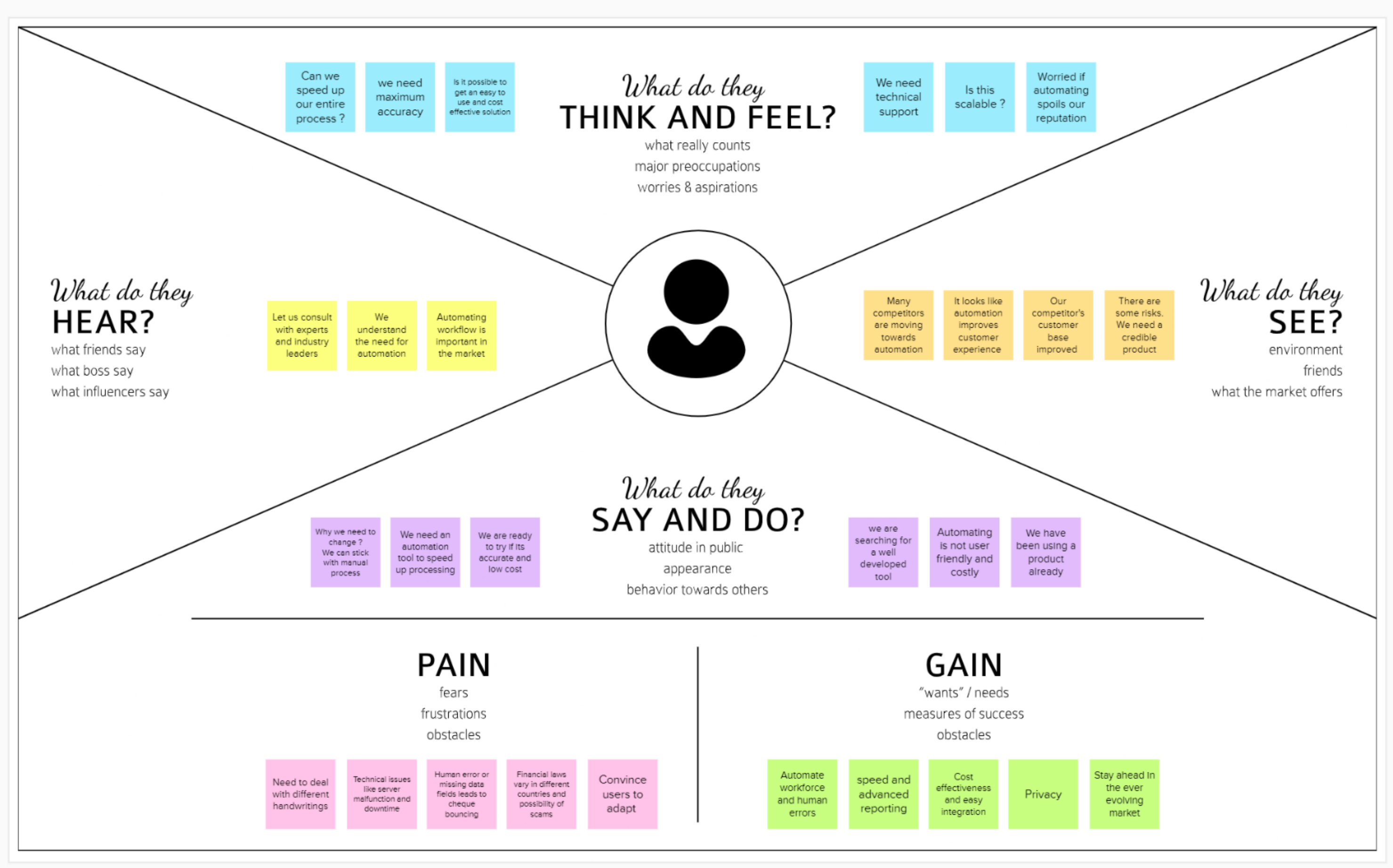
Customized features and a vast quantity of past knowledge have been used in traditional handwriting recognition systems. It is difficult to train an optical character recognition (OCR) system based on these conditions. Deep learning approaches have enabled significant performance in the field of handwriting recognition research in recent years. Nonetheless, the increasing increase in the amount of handwritten data, along with the availability of vast computing capacity, necessitates improvements in recognition accuracy and warrants additional exploration. Convolutional neural networks (CNNs) are extremely excellent in perceiving the structure of handwritten characters/words in ways that aid in the automatic extraction of distinguishing features, making CNN the best solution for solving handwriting recognition challenges.

The proposed work aims to investigate several design alternatives for CNN-based handwritten digit recognition, such as the number of layers, stride size, receptive field, kernel size, padding, and dilution. Furthermore, we intend to assess the effectiveness of several SGD optimization techniques in enhancing the performance of handwritten digit recognition. Using ensemble architecture improves the recognition accuracy of a network. In this case, we want to obtain equal accuracy by employing a pure CNN design without ensemble architecture, because ensemble structures increase computational overhead and testing complexity. As a result, a CNN design is developed in order to obtain higher accuracy than ensemble systems while reducing operational complexity and expense. Furthermore, we demonstrate an appropriate combination of learning parameters in the design of a CNN that leads us to a new absolute record in categorising MNIST handwritten digits. We conducted extensive trials and achieved 99.87% recognition accuracy for an MNIST dataset.

CHAPTER 3

IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS



What do they THINK AND FEEL?

what really counts

major preoccupations

worries & aspirations

Can we speed up our entire process ?

we need maximum accuracy

Is it possible to get an easy to use and cost effective solution

We need technical support

Is this scalable ?

Worried if automating spoils our reputation

What do they HEAR?

what friends say

what boss say

what influencers say

Let us consult with experts and industry leaders

We understand the need for automation

Automating workflow is important in the market

What do they SEE?

environment

friends

what the market offers

Many competitors are moving towards automation

It looks like automation improves customer experience

Our competitor's customer base improved

There are some risks. We need a credible product

What do they SAY AND DO?

attitude in public

appearance

behavior towards others

Why we need to change ? We can stick with manual process

We need an automation tool to speed up processing

We are ready to try if its accurate and low cost

we are searching for a well developed tool

Automating is not user friendly and costly

We have been using a product already

PAIN

fears

frustrations

obstacles

Need to deal with different handwritings

Technical issues like server malfunction and downtime

Human error or missing data fields leads to cheque bouncing

Financial laws vary in different countries and possibility of scams

Convince users to adapt

GAIN

"wants" / needs

measures of success

obstacles

Automate workforce and human errors

speed and advanced reporting

Cost effectiveness and easy integration

Privacy

Stay ahead in the ever evolving market

3.2 IDEATION & BRAINSTORMING

Rajeev Krishna:

1. Focus on one feature at a time
2. Clean and Simple UI
3. No bulk commits in Github
4. Analyze existing solutions
5. Compare with requirements more often
6. Clean customer navigations

Maanas Karthikeyan:

1. Stick together as a team - frequent discussions
2. Break into pieces - discuss, code, test and deploy
3. Each phase - working model to beta test
4. Use project management tools
5. Analyze the data source and explore other sources for future use
6. Bring drag and drop options to upload images + preview

Priyadharshan Ravichandran:

1. Discussion before every phase and writing code
2. Get feedback from other friends as a user
3. Develop while learning approach
4. Analyze user requirements and thoughts
5. Use collaborative to do apps
6. Proper project structure - develop layout and logic subsequently

Sai Prakash. R:

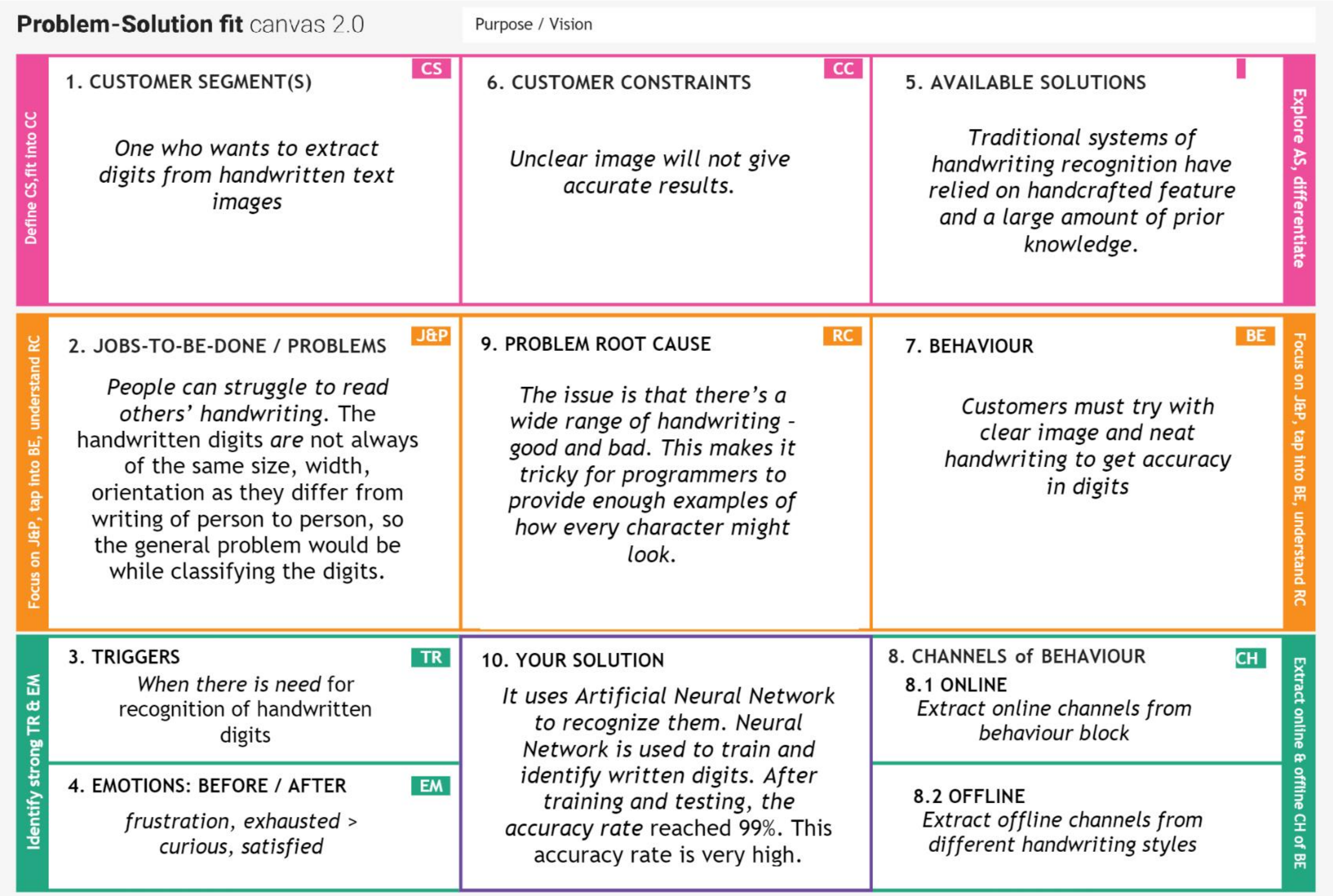
1. Think like a user - then design
2. No unwanted features - quality first
3. clean and commented code - avoid inline comments
4. Work in branches, and give pull request
5. Consider scalability and continuous development right from the start
6. Instead of developing all at once and deploying, do continuous development and deployment

3.3 PROPOSED SOLUTION

S.No	Parameter	Description
1.	Problem Statement (Problem to besolved)	Handwriting recognition is one of the compelling research works going on because every individual in this world has their own style of writing. It is the capability of the computer to identify and understand handwritten digits or characters automatically.Because of the progress in the field of science and technology, everything is being digitized toreduce human effort. Hence, there comes a need for handwritten digit recognition in manyreal-time applications. Our University is still relying on manual processing which is both time consuming and error prone. Handwriting recognition system with a reliable accuracy can make an impact in these business fields.
2.	Idea / Solution description	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. We opt to use multi-layerneural networks as deep NN. Due to the fact that data is Image, the best type of neural network satisfying our goal is **Convolutional Neural Networks**. As we have to do for most ofthe data, normalization plays an important rolein our process. Before doing any tasks, pre- processing images (our data-set) is highly recommended. Consequently better accuracy will be achieved by pre-processed data. After pre-processing and normalizing, the prepared data set could be used as input to our deep convolutional neural network. Then deep NN

		will be run and fit to our data and the result willbe produced by that.
3.	Novelty / Uniqueness	Web application is created where the user canupload an image of a handwritten digit. this image is analysed by the model and the detected result is returned on to UI. One of themajor decisions had to be made was choosing the suitable programming language satisfying our goal for extracting knowledge from our data. After some searching the suitable decision has been made by selecting Python asthe project programming language. Due to the fact that, a lot of tools and frameworks are available for Python to create powerful Artificial Neural Networks. Also IBM Watson helps to predict future outcomes, automate complex processes, and optimize user's time. And also the result accuracy will be increased from 70% which is the accuracy of the test results that the previous developed codes produced.
4.	Social Impact / Customer Satisfaction	People can struggle to read others' handwriting. The handwritten digits are not always of the same size, width, orientation as they differ from writing of person to person, sothe general problem would be while classifyingthe digits. Can automate data entry jobs and speed up evaluation process.
5.	Business Model (Revenue Model)	Can collaborate with other institutions to speedup the evaluation process which improves customer experience.
6.	Scalability of the Solution	This project will help us to detect digits more precisely. Also we can develop this model to recognize alphabets.

3.4 PROBLEM SOLUTION FIT



Problem-Solution fit canvas 2.0

Purpose / Vision

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

One who wants to extract digits from handwritten text images

6. CUSTOMER CONSTRAINTS CC

Unclear image will not give accurate results.

5. AVAILABLE SOLUTIONS

Traditional systems of handwriting recognition have relied on handcrafted feature and a large amount of prior knowledge.

Explore AS, differentiate

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

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

People can struggle to read others' handwriting. The handwritten digits *are* not always of the same size, width, orientation as they differ from writing of person to person, so the general problem would be while classifying the digits.

9. PROBLEM ROOT CAUSE RC

The issue is that there's a wide range of handwriting - good and bad. This makes it tricky for programmers to provide enough examples of how every character might look.

7. BEHAVIOUR BE

Customers must try with clear image and neat handwriting to get accuracy in digits

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

Identify strong TR & EM

3. TRIGGERS TR

When there is need for recognition of handwritten digits

4. EMOTIONS: BEFORE / AFTER EM

frustration, exhausted > curious, satisfied

10. YOUR SOLUTION

It uses Artificial Neural Network to recognize them. Neural Network is used to train and identify written digits. After training and testing, the accuracy rate reached 99%. This accuracy rate is very high.

8. CHANNELS of BEHAVIOUR CH

8.1 ONLINE

Extract online channels from behaviour block

8.2 OFFLINE

Extract offline channels from different handwriting styles

Extract online & offline CH of BE

CHAPTER 4

REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

FR No.	Functional Requirement	Sub Requirement
1	**Input**	• **Inputs** is in the form of the **images**. (JPG, PNG)
2	**Error detection**	• System shall show the error message to the user when the **input** given is **not** in the **required format**.
3	**Detect Target**	• System should **detect characters** present in theimage. • Must be able to perform classification and shouldrecognize thehandwritten input
4	**Output**	• System should **retrieve characters** present in theimage and display them to the user. • Must be able to display the accurate output in textformat.

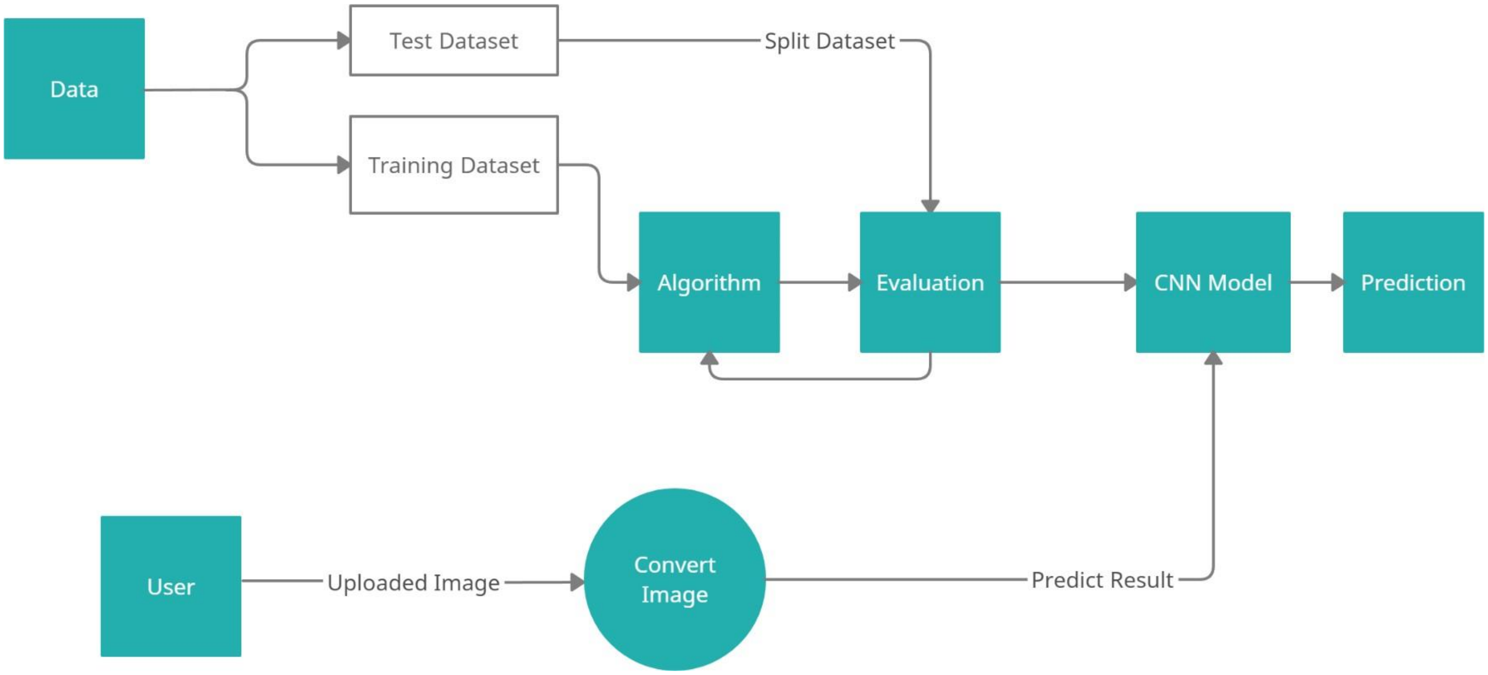
4.2 NON FUNCTIONAL REQUIREMENTS

NFR No.	Non-Functional Requirement	Descriptio n
1	**Usability**	• Application for digit recognition to map Roll numbers and the test scores obtained by students in an examination • Other applications can include filling out forms, processing bank checks, and sorting mail. • It can also used for blind-people by using soundinput.
2	**Security**	• Banking sector where it can be used to maintainthe security pin number safely.
3	**Reliability**	• This software will work reliably for low resolution image and not for graphical images • The standard implementations of neural networks achieve an accuracy of approximately
4	**Performance**	• Software will perform its intended function for alarge period of sufficient time and also it will operate in a secured environment without anyfailures.
5	**Availability**	• This system will retrieve the handwritten text regions only if the image contains written text init.
6	**Scalability**	• System can work normally under any amount ofhandwritten data that is given as input.

CHAPTER 5

PROJECT DESIGN

5.1 DATA FLOW DIAGRAM



5.2 SOLUTION & TECHNICAL ARCHITECTURE

SOLUTION ARCHITECTURE

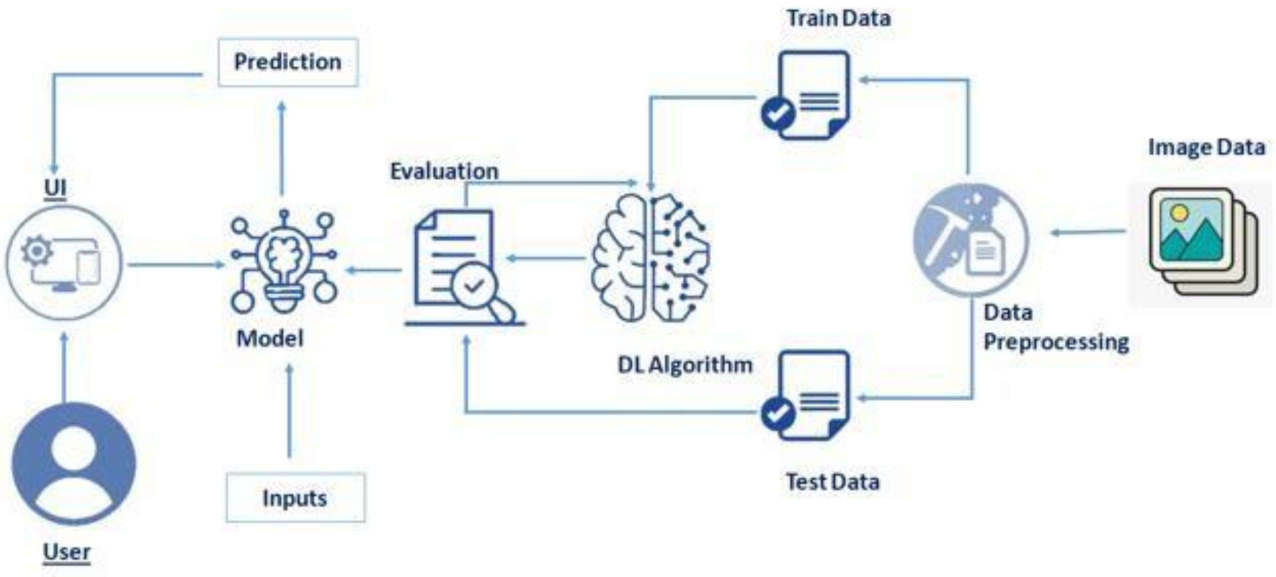
- Character recognition plays an important role in the modern world. It can solve more complex problems and makes humans' job easier. An example is handwritten character recognition. This is a system widely used in the world to recognize zip code or postal code for mail sorting.

- There are different techniques that can be used to recognize handwritten characters. Two techniques researched in this paper are Pattern Recognition and Artificial Neural Network (ANN). Both techniques are defined and different methods for each technique is also discussed.

- Bayesian Decision theory, Nearest Neighbor rule, and Linear Classification or Discrimination is types of methods for Pattern Recognition. Shape recognition, Chinese Character and Handwritten Digit recognition uses Neural Network to recognize them.

- Neural Network is used to train and identify written digits. After training and testing, the accuracy rate reached 99%. This accuracy rate is very high.

SOLUTION ARCHITECTURE DIAGRAM



TECHNICAL ARCHITECTURE

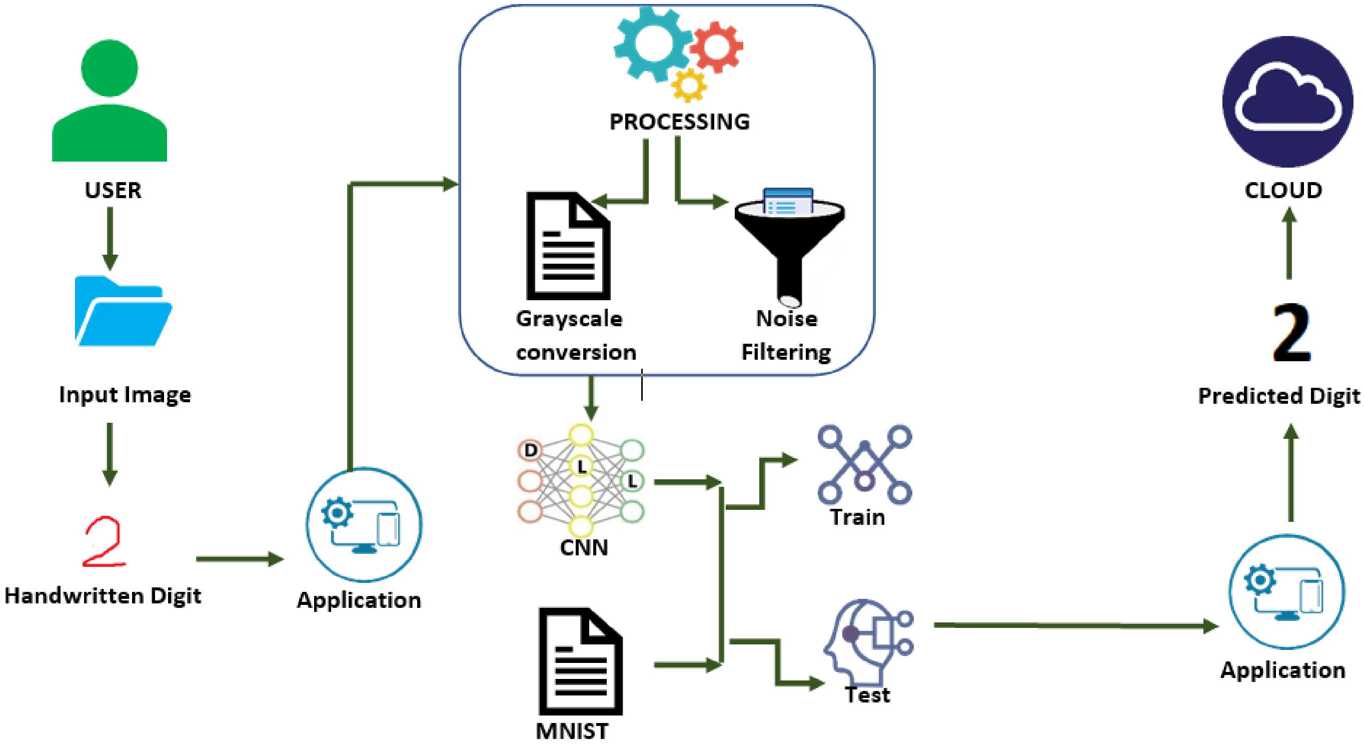


Table 1: Components & Technologies

S.No	Component	Description	Technology
1.	User Interface	Web UI, Mobile App, Desktop Application	HTML, CSS, JavaScript
2.	Application Logic-1	Make GUI Interface	HTML, CSS
3.	Application Logic-2	Importing library and Code for Digit recognition	Python, Flask
4.	Application Logic-3	Train the model on IBM	Flask
5.	Prediction	Digit prediction on the image	Keras, CNN
6.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant, etc.
7.	File Storage	File storage requirements	IBM Block Storage or Other StorageService or Local Filesystem
8.	Machine Learning Model	Purpose of Machine Learning Model is to train andtest the data and predict the user input.	Object Recognition Model, etc.
9.	Neural network	Automatically infer rules for recognizing handwritten digits	Convolutional neural network

Table 2: Application Characteristics

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Enables developers to develop complex code and webapplication quickly.	Jupyter, anaconda navigator, flaskframework.
2.	Security Implementations	After predicting the data, we don’t store any data so wecan’t manipulate it in future.	Encryption
3.	Scalable Architecture	The scalability of architecture (3 – tier, Micro-services)	Browser, Web server (Database)
4.	Availability	Speed of Digit Detection with any handwritten	Convolutional Neural Networks
5.	Performance	Neural networks achieve an accuracy of 98-99 percent incorrectly classifying the handwritten digits	Convolutional Neural Networks

5.3 USER STORIES

User Type	Functional Requirement(Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Web user)	Image upload	USN-1	User can upload image using the input component	user can upload onlyimages	High	Sprint-1
Customer (Web user)	image upload	USN-2	adding upload button to send image for prediction	button is enabled only when image is uploaded	High	Sprint-1
Customer (Web user)	results	USN-3	graph to show the prediction		High	Sprint-4
Customer (Web user)	training	USN-4	importing keras datasets and integrating it		High	Sprint-2
Customer (Web user)	training	USN-5	model training	image dataset only	High	Sprint-2
Customer (Web user)	training	USN-6	model testing	image dataset only	High	Sprint-3
Customer (Web user)	results	USN-7	result formatting		High	Sprint-4

CHAPTER 6

PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNING AND ESTIMATION

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data Collection	USN-1	As a user, I can collect the dataset fromvarious resources with different handwritings.	10	Low	Rajeev Krishna Maanas Karthikeyan
Sprint-1	Data Pre-processing	USN-2	As a user, I can load the dataset, handle the missing data, scale and split data into train and test.	10	Medium	Rajeev Krishna Priyadharshan R
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	Sai Prakash R Priyadharshan R
Sprint-2	Add CNN layers	USN-4	Creating the model and adding the input, hidden, and output layers to it.	5	High	Rajeev Krishna Priyadharshan R Sai Prakash R Maanas Karthikeyan
Sprint-2	Compiling the model	USN-5	With both the training data and model defined, it's time to configure the learningprocess.	2	Medium	Rajeev Krishna Priyadharshan R
Sprint-2	Train & test themodel	USN-6	As a user, let us train our model with our image dataset.	6	Medium	Rajeev Krishna Priyadharshan R
Sprint-2	Save the model	USN-7	As a user, the model is saved & integrated with an android application or web application in order	2	Low	Sai Prakash R Maanas Karthikeyan

			to predict something.			
Sprint-3	Building UI Application	USN-8	As a user, I will upload the handwritten digit image to the application by clicking the upload button.	5	High	Sai Prakash R Maanas Karthikeyan
Sprint-3		USN-9	As a user, I can know the details of thefundamental usage of the application.	5	Low	Rajeev KrishnaSai Prakash R
Sprint-3		USN-10	As a user, I can see the predicted / recognized digits in the application.	5	Medium	Priyadharshan R Maanas Karthikeyan
Sprint-4	Train the model on IBM	USN-11	As a user, I train the model on IBM and integrate flask/Django with scoring endpoint.	10	High	Sai Prakash R Maanas Karthikeyan Rajeev Krishna Priyadharshan R
Sprint-4	Cloud Deployment	USN-12	As a user, I can access the web application and make use of the product from anywhere.	10	High	Sai Prakash R Maanas Karthikeyan

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	29 Oct 2022	03 Nov 2022	20	03 Nov 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

CHAPTER 7

CODING & SOLUTIONING

Recognizer.py

```python
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
```

App.py

```
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(debug=True)
```

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

* {
    padding: 0;
    margin: 0;
}

body {
    background-image: url('predback.jpg');
    color: black;
    font-family: "Overpass", sans-serif;
}

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

.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;
    }
}

        </style>
    </head>
    <body>
        <div class="container">
            <div class="heading">
                <h1 class="heading__main">Handwritten Digit Recognition System</h1><br>
                <h4>
                    Handwritten Digit Recognition is a technology that is much needed in this world as of Today.This Digit Recognition System is used to recognize the digits from different sources like email, posts, cheque etc. Before proper implementation of this technology we have relied on writing text with our own hands which can result in error.It's difficult to store and access physical data with efficiency.The project presents in representing the recognization of handwritten digits (0 - 9) from the famous MNIST dataset. Here we will be using AlexNet which is an architecture of Convolutional Neural Network.
                </h2><br>
                <h2 class="heading__sub">Easily analyze and detect handwritten digits</h2><br>
            </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>
```

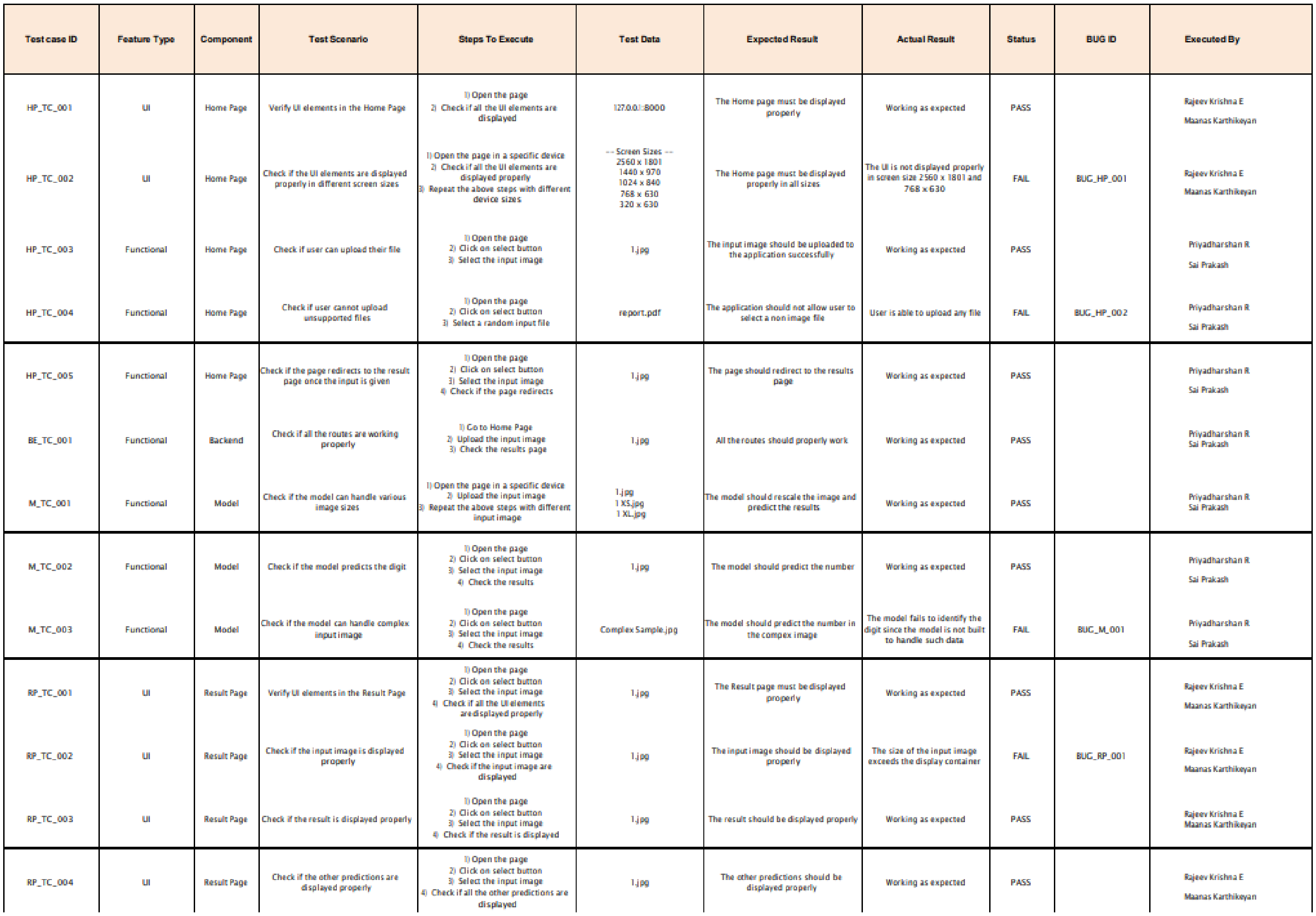
Predict.html

```
<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>
```

CHAPTER 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		Rajeev Krishna E Maanas Karthikeyan
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	Rajeev Krishna E Maanas Karthikeyan
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	1.jpg	The input image should be uploaded to the application successfully	Working as expected	PASS		Priyadharshan R Sai Prakash
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	report.pdf	The application should not allow user to select a non image file	User is able to upload any file	FAIL	BUG_HP_002	Priyadharshan R Sai Prakash
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	1.jpg	The page should redirect to the results page	Working as expected	PASS		Priyadharshan R Sai Prakash
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 results page	1.jpg	All the routes should properly work	Working as expected	PASS		Priyadharshan R Sai Prakash
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	1.jpg 1 XS.jpg 1 XL.jpg	The model should rescale the image and predict the results	Working as expected	PASS		Priyadharshan R Sai Prakash
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	1.jpg	The model should predict the number	Working as expected	PASS		Priyadharshan R Sai Prakash
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.jpg	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	Priyadharshan R Sai Prakash
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) Check if all the UI elements are displayed properly	1.jpg	The Result page must be displayed properly	Working as expected	PASS		Rajeev Krishna E Maanas Karthikeyan
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	1.jpg	The input image should be displayed properly	The size of the input image exceeds the display container	FAIL	BUG_RP_001	Rajeev Krishna E Maanas Karthikeyan
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	1.jpg	The result should be displayed properly	Working as expected	PASS		Rajeev Krishna E Maanas Karthikeyan
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	1.jpg	The other predictions should be displayed properly	Working as expected	PASS		Rajeev Krishna E Maanas Karthikeyan

8.2 USER ACCEPTANCE TESTING

8.2.1 DEFECT ANALYSIS

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Total
By Design	1	0	0	0	1
Duplicate	0	1	0	0	1
External	0	2	1	0	3
Fixed	3	0	0	1	4
Not Reproduced	0	0	1	0	1
Skipped	0	0	0	2	2
Won’t Fix	1	0	1	0	2
Total	5	3	3	3	14

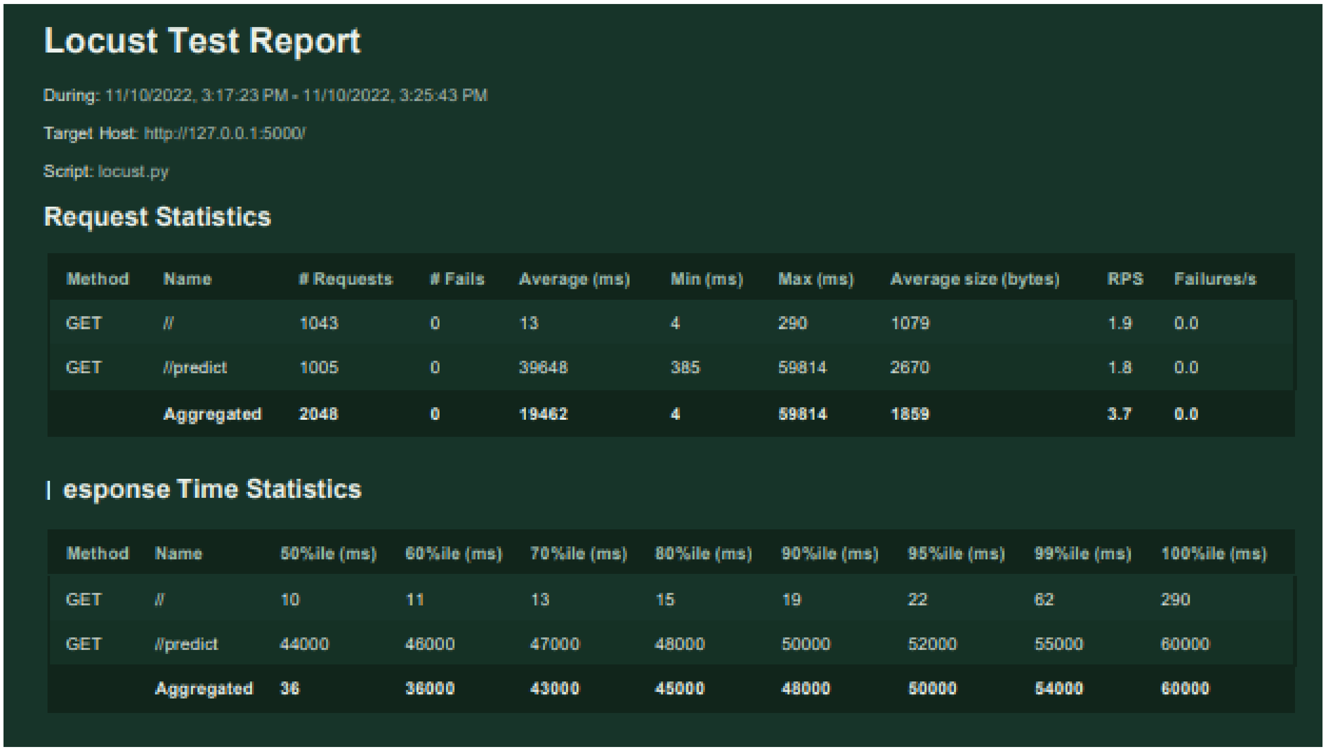
8.2.2 TEST CASE ANALYSIS

Section	Total Cases	Not Tested	Fail	Pass
Client Application	10	0	2	8
Security	2	0	1	1
Performance	3	0	1	2
Exception Reporting	2	0	0	2

CHAPTER 9

RESULTS

9.1 PERFORMANCE METRICS



Locust Test Report

During: 11/10/2022, 3:17:23 PM - 11/10/2022, 3:25:43 PM

Target Host: http://127.0.0.1:5000/

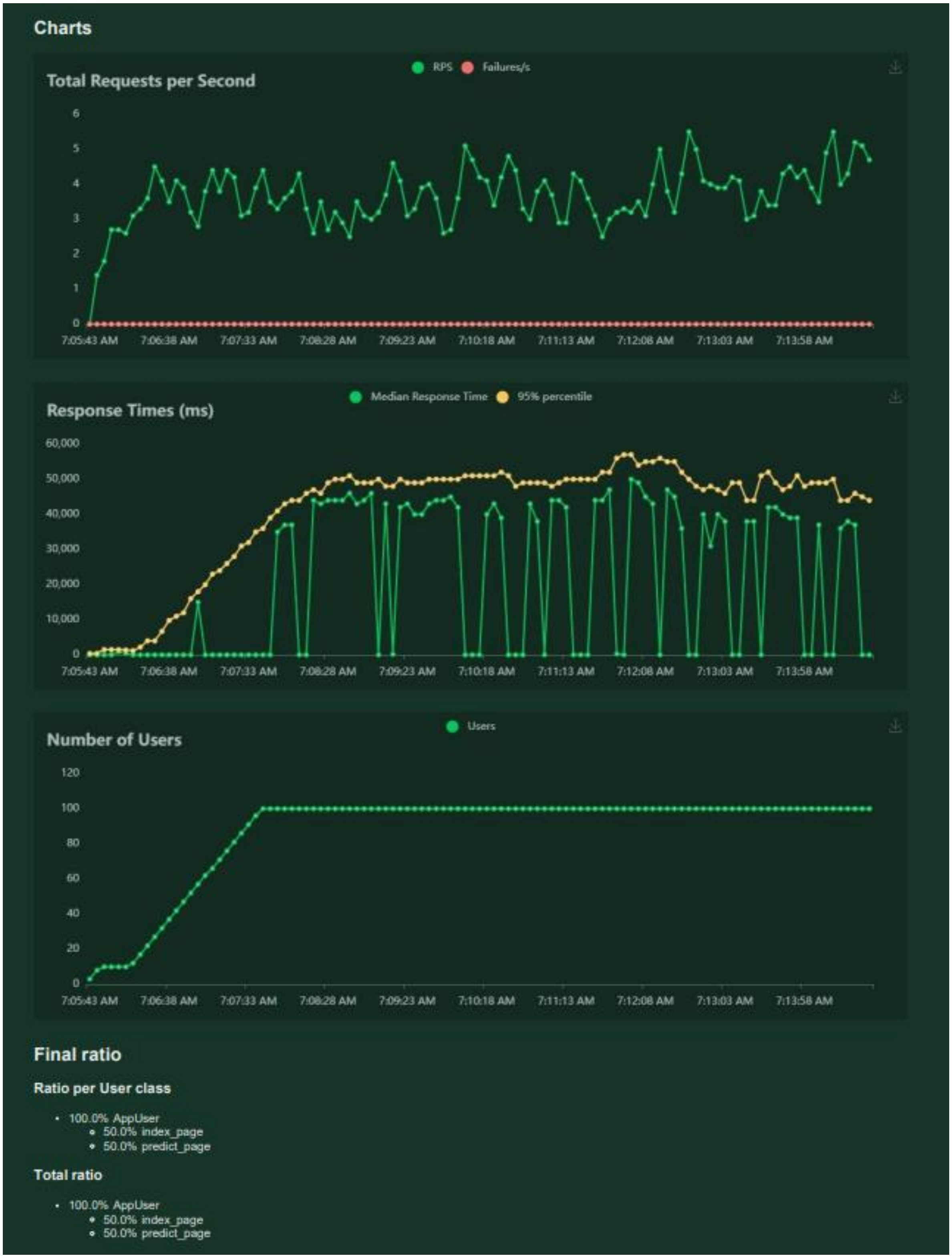
Script: locust.py

Request Statistics

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

I esponse 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

Final ratio

Ratio per User class

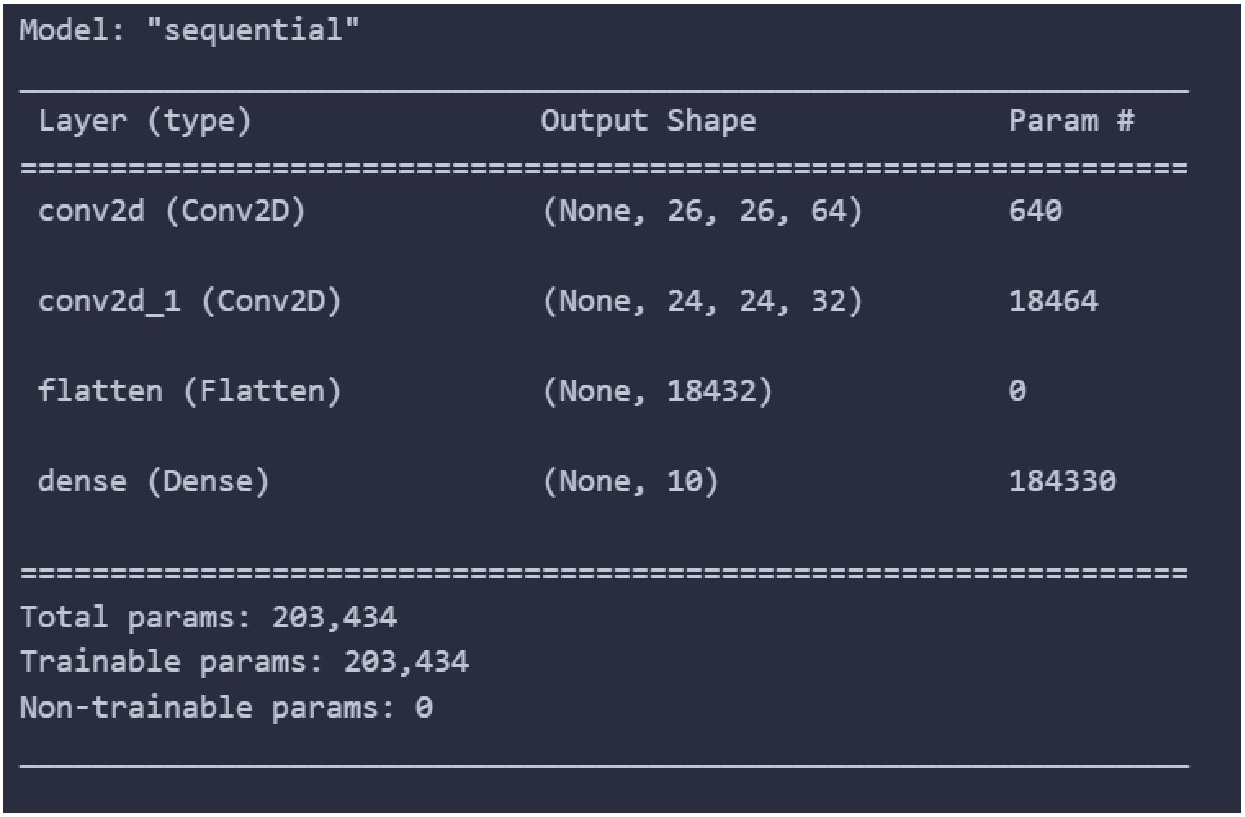
- 100.0% AppUser
  - 50.0% index_page
  - 50.0% predict_page

Total ratio

- 100.0% AppUser
  - 50.0% index_page
  - 50.0% predict_page

PERFORMANCE TESTING

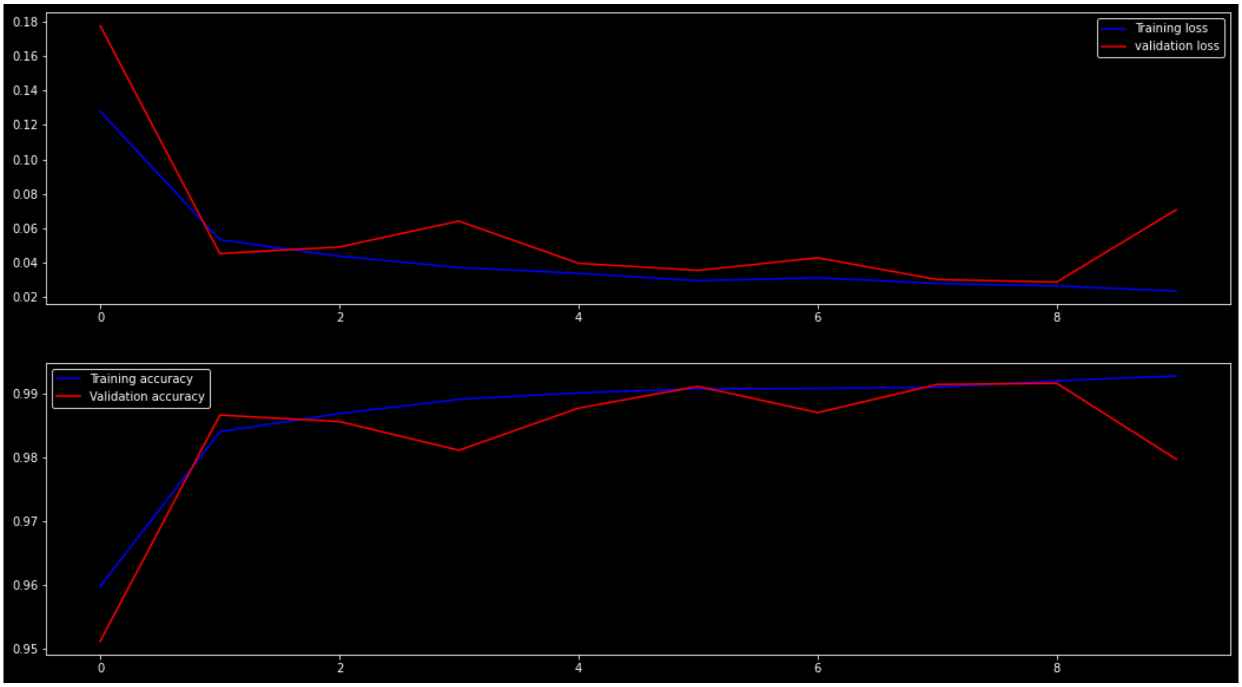
MODEL SUMMARY



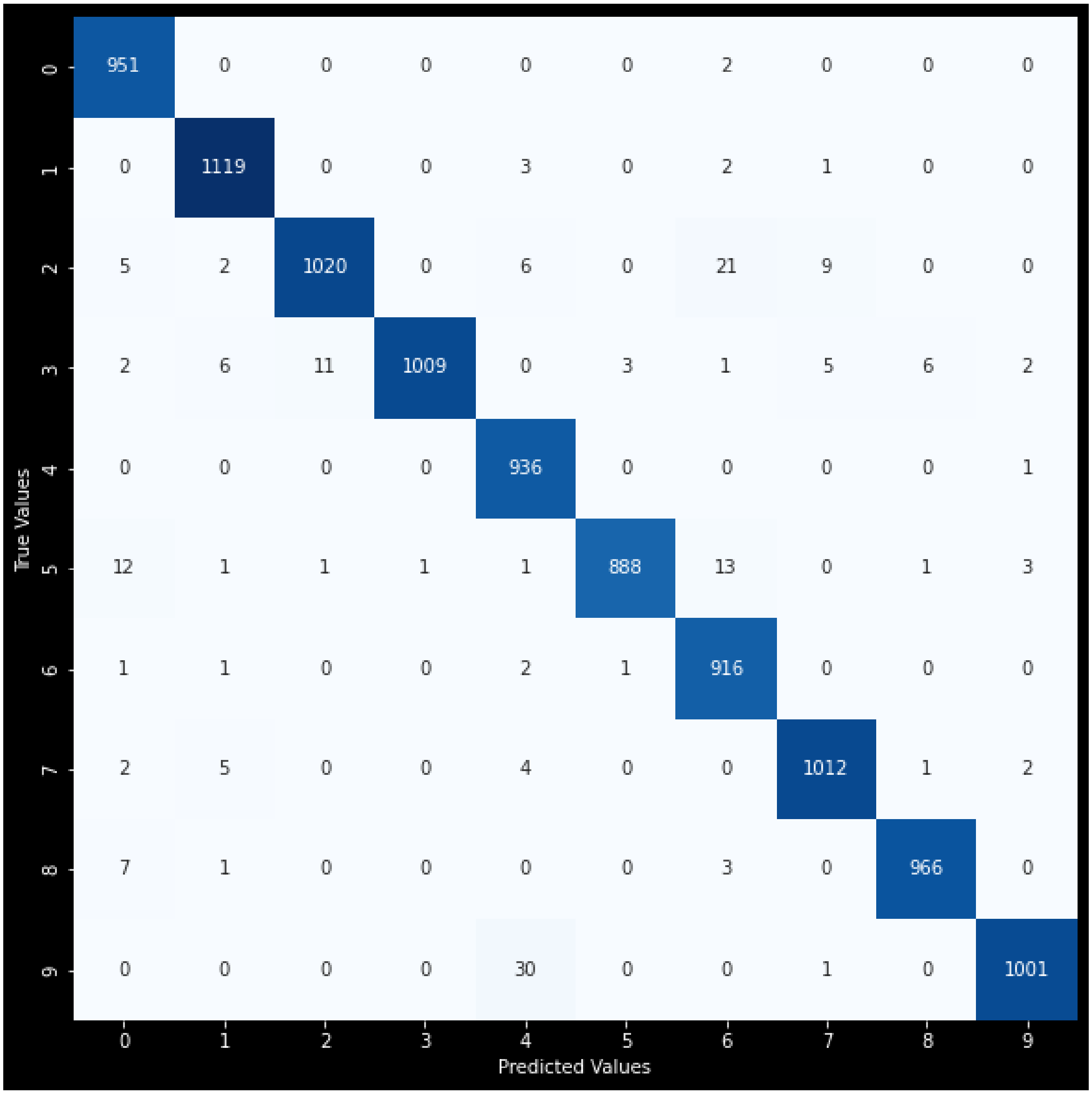
```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 26, 26, 64)        640

 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
_________________________________________________________________
```

ACCURACY

CONTENT	VALUE
Training Accuracy	99.14%
Training Loss	2.70%
Validation Accuracy	97.76%
Validation Loss	10.36%

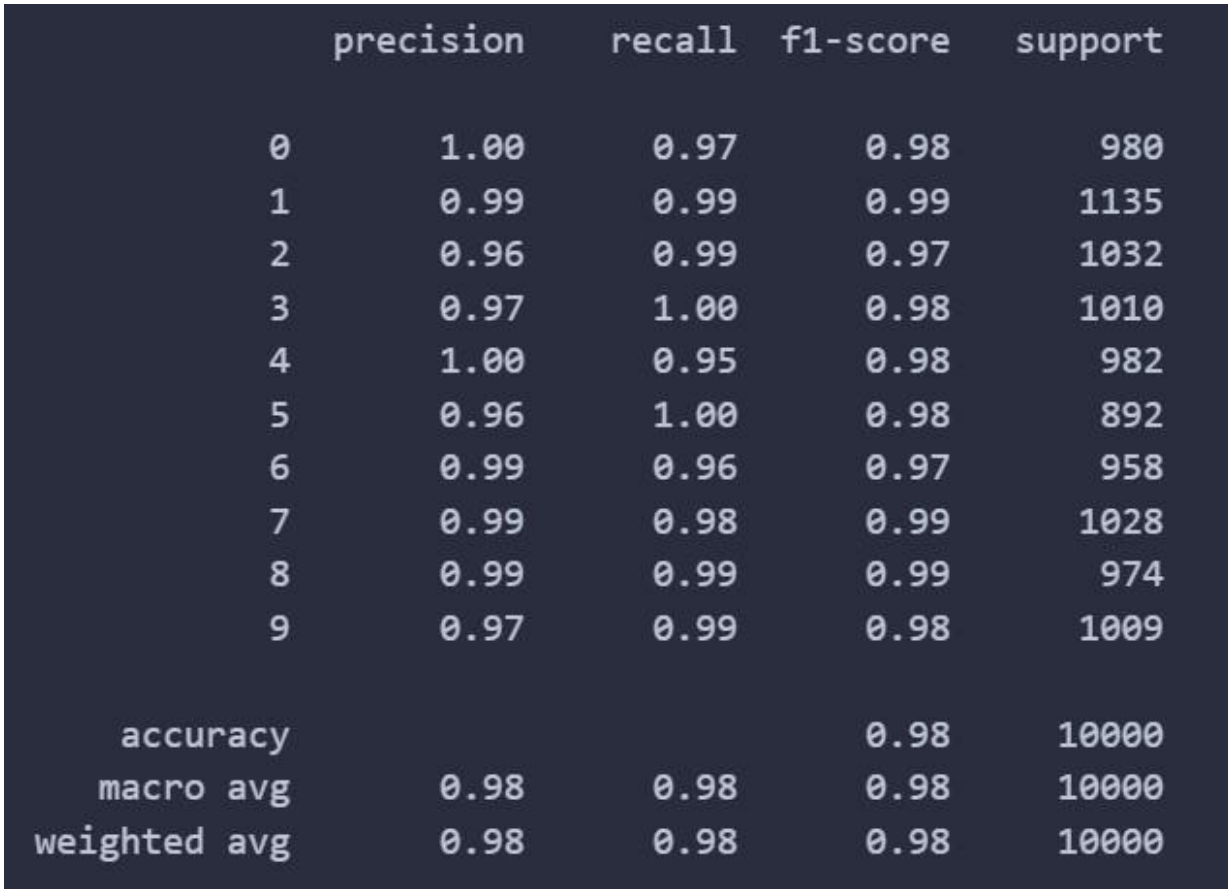


CONFUSION MATRIX



	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

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
```

CHAPTER 10

ADVANTAGES & DISADVANTAGES

ADVANTAGES

- Reduces manual work
- More accurate than average human
- Capable of handling a lot of data
- Can be used anywhere from any device
- Speed of data entry
- Easier data retrieval
- The accuracy rate is very high

DISADVANTAGES

- Unclear image will not give accurate results
- Cannot handle complex data
- All the data must be in digital format
- Requires a high performance server for faster predictions
- Prone to occasional errors

CHAPTER 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,answer scirpts) and so on. There is so much room for improvement, which can be implemented in subsequent versions.

CHAPTER 12

FUTURE SCOPE

This project can be enhanced with a great field of machine learning and artificial intelligence. The world can think of a software which can recognize the text from a picture and can show it to the others. Or this project can be extended to a greater concept of all the character sets in the world. This project has not gone for the total English alphabet because there will be more and many more training sets and testing values that the neural network model will not be enough to detect.

All of these enhancement is an application of the texture analysis where advanced image processing, Neural network model for training and advanced AI concepts will come. These applications can be modeled further .As this project is fully done by free and available resources and packages this can be also a limitation of the project. The fund is very important because all machine learning libraries and advanced packages are not available for free. Unless of those the most of the visualizing platforms like on which developers are doing some works like Watson Studio or Aws. These all are mainly paid platforms where a lot of ML projects are going on.

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

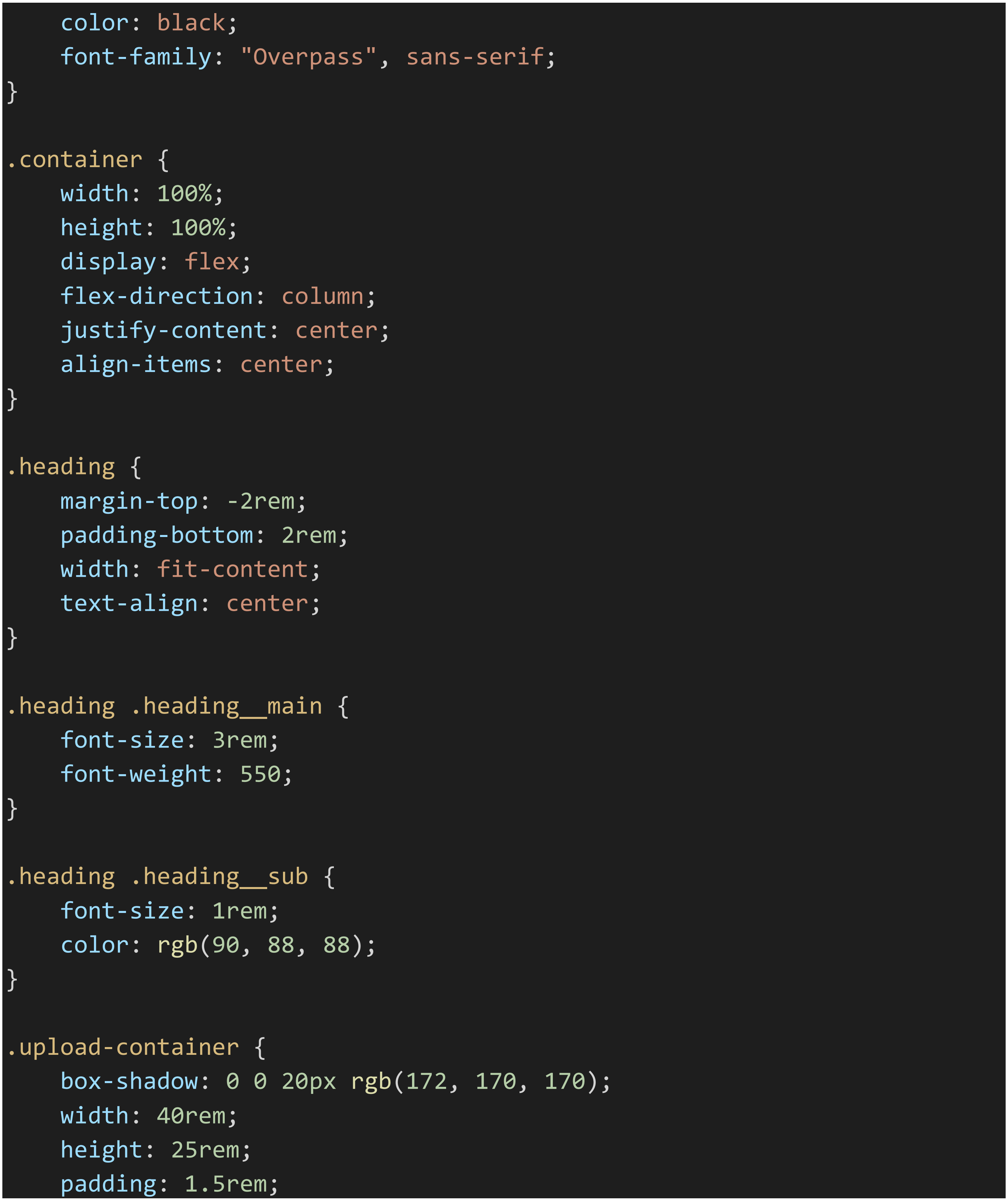
This project has endless potential and can always be enhanced to become better. Implementing this concept in the real world will benefit several industries and reduce the workload on many workers, enhancing overall work efficiency.

APPENDIX

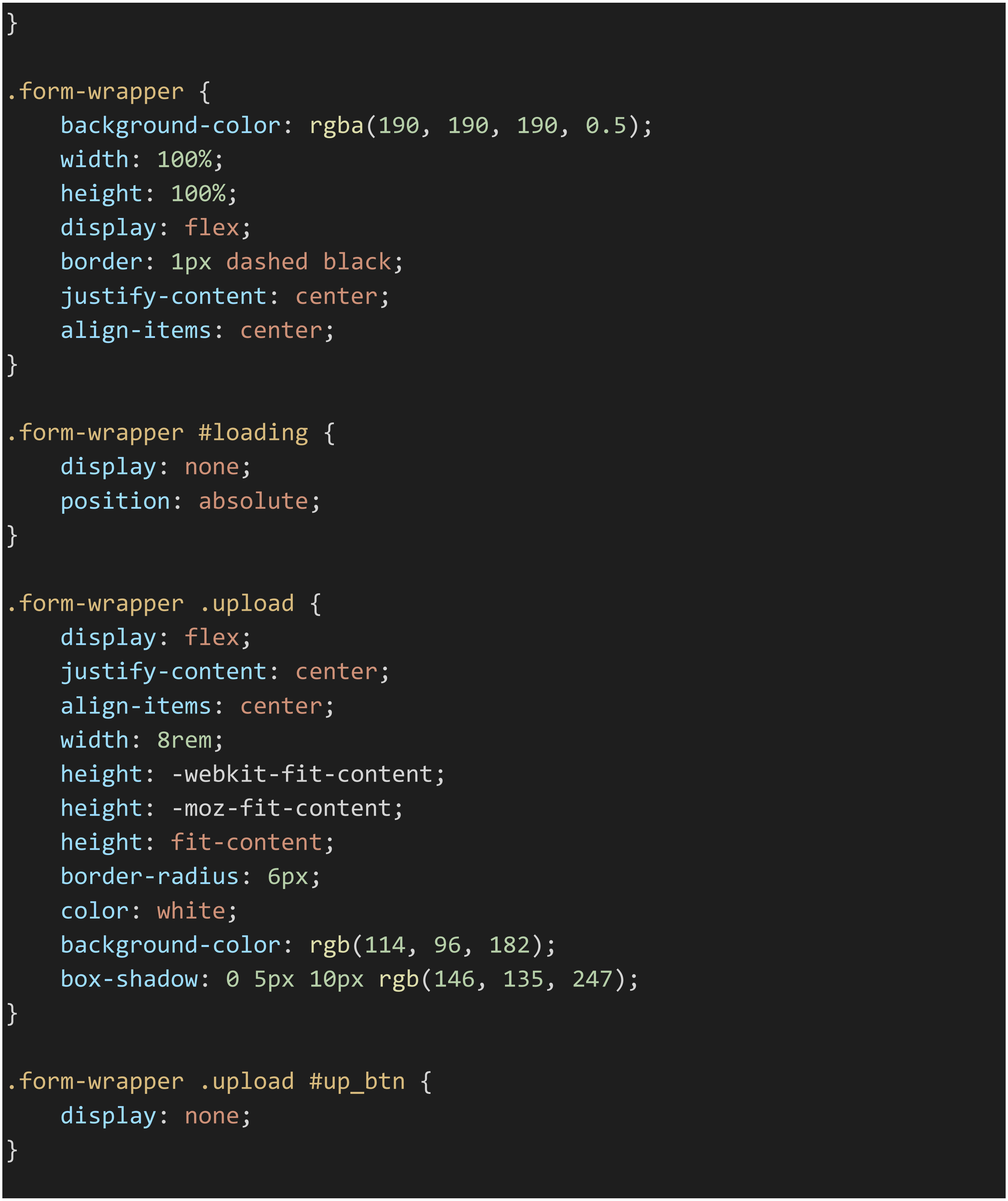
SOURCE CODE

home.html

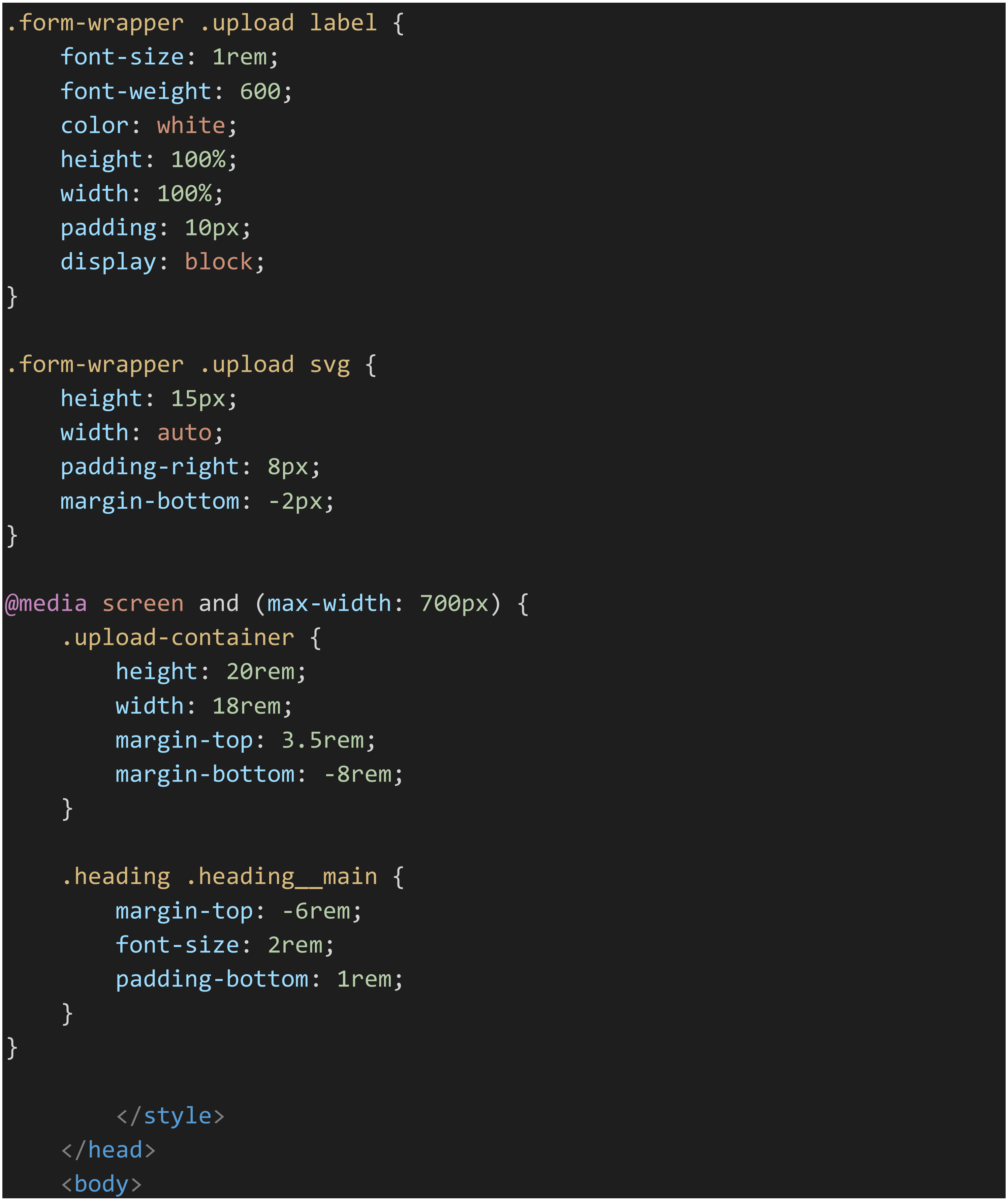
```
<html>
    <head>
        <meta name="viewport" content="width=device-width, initial-
scale=1.0" />
        <title>Handwritten Digit Recognition</title>
        <link rel="icon" type="image/svg" sizes="32x32"
href="{{url_for('static',filename='images/icon.svg')}}" />

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

* {
    padding: 0;
    margin: 0;
}

body {
    background-image: url('predback.jpg');
```



```css
    color: black;
    font-family: "Overpass", sans-serif;
}

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

.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;
```



```css
}

.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;
}
```



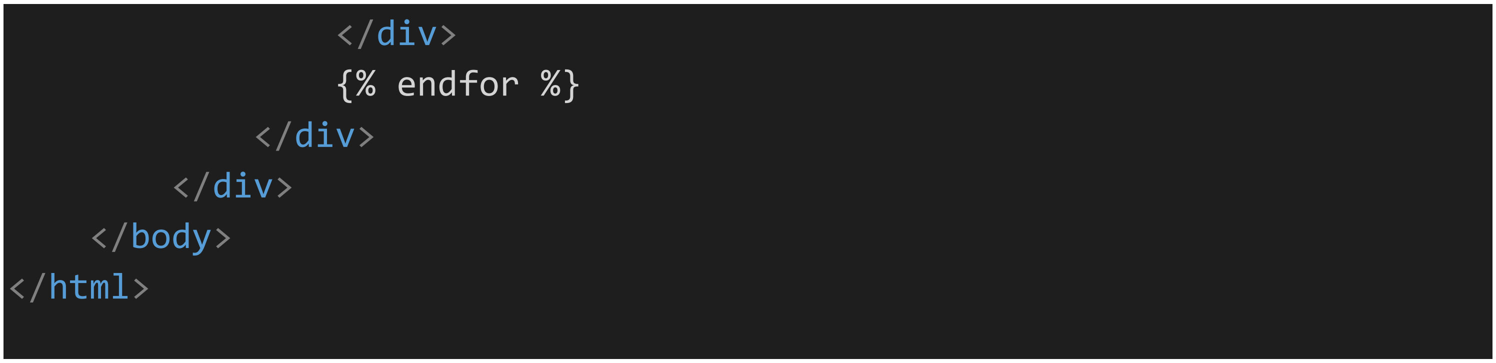
```
.form-wrapper .upload label {
    font-size: 1rem;
    font-weight: 600;
    color: white;
    height: 100%;
    width: 100%;
    padding: 10px;
    display: block;
}

.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;
    }
}

        </style>
    </head>
    <body>
```

```
        <div class="container">
            <div class="heading">
                <h1 class="heading__main">Handwritten Digit Recognition
System</h1><br>
                <h4>
                    Handwritten Digit Recognition is a technology that
is much needed in this world as of Today.This Digit Recognition System
is used to recognize the digits from different sources like email,
posts, cheque etc. Before proper implementation of this technology we
have relied on writing text with our own hands which can result in
error.It's difficult to store and access physical data with
efficiency.The project presents in representing the recognization of
handwritten digits (0 - 9) from the famous MNIST dataset. Here we will
be using AlexNet which is an architecture of Convolutional Neural
Network.
                </h2><br>
                <h2 class="heading__sub">Easily analyze and detect
handwritten digits</h2><br>
            </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>
```

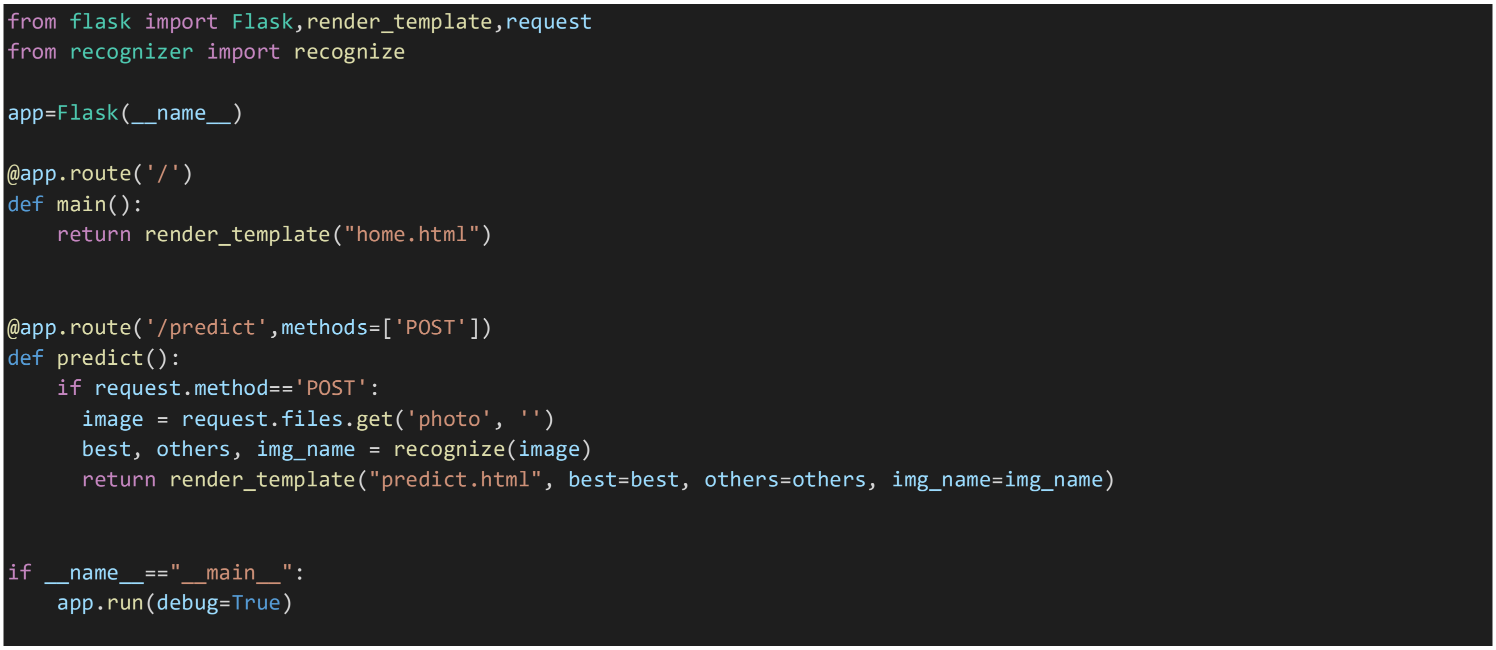
predict.html

```
<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>
```

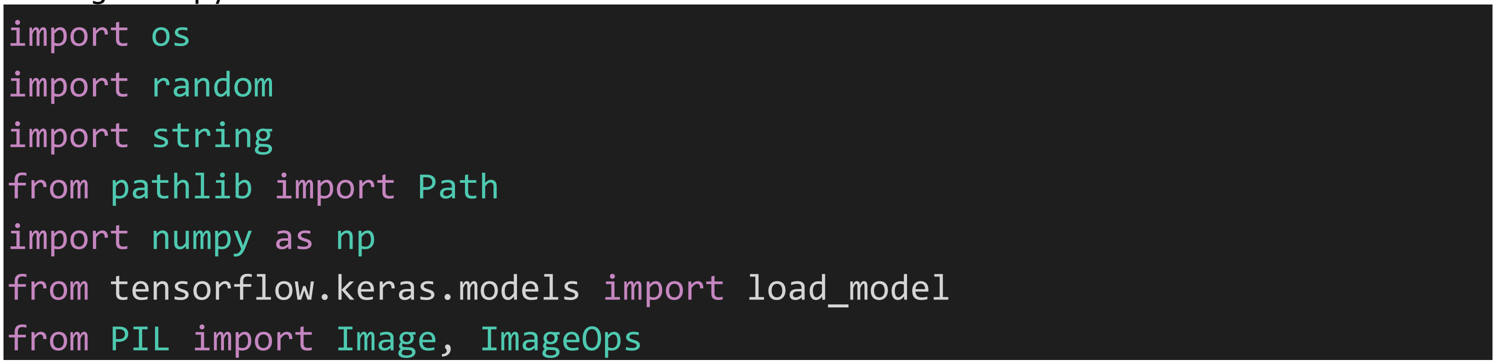
App.py



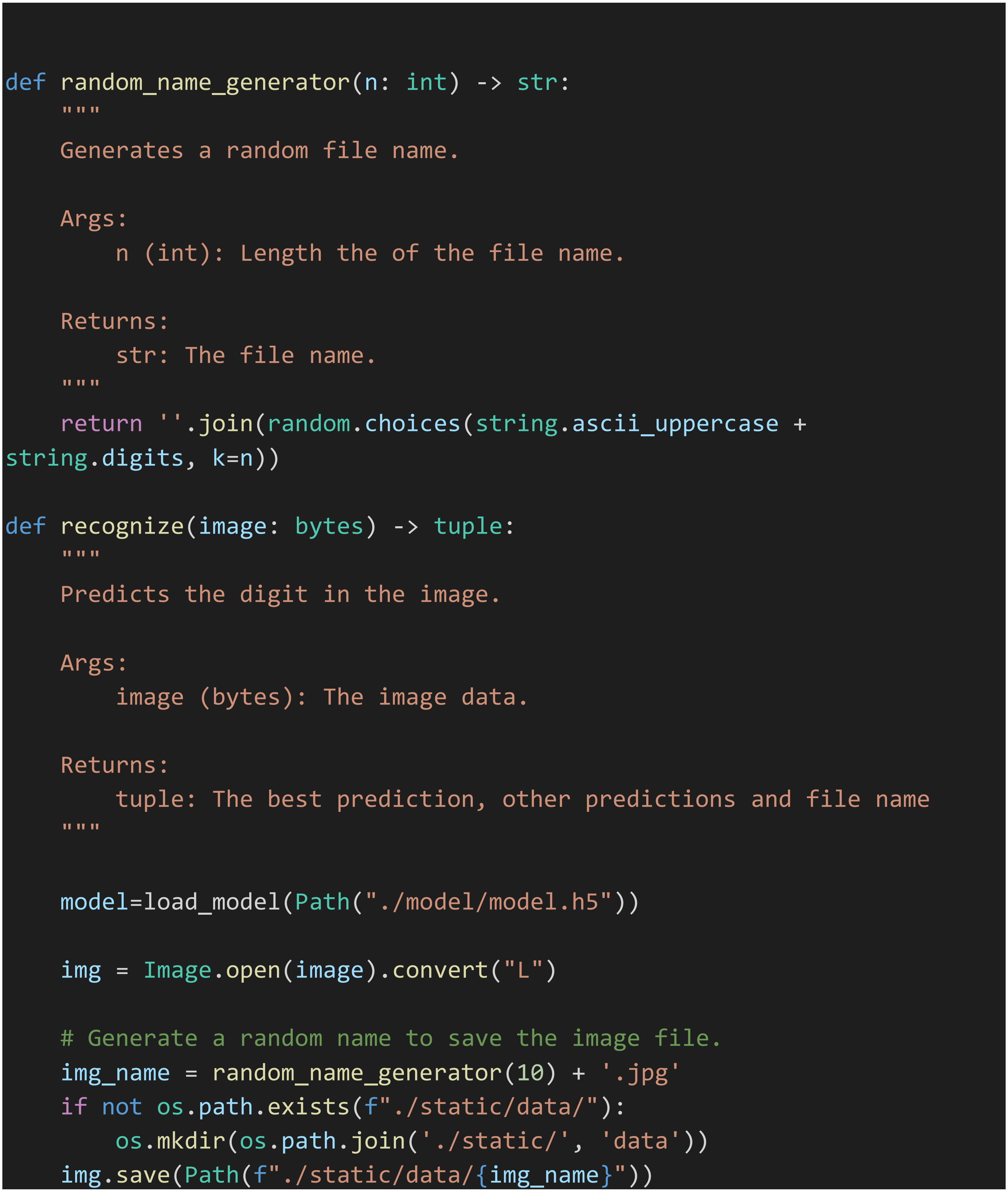
```
from flask import Flask,render_template,request
from recognizer import recognize

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(debug=True)
```

I

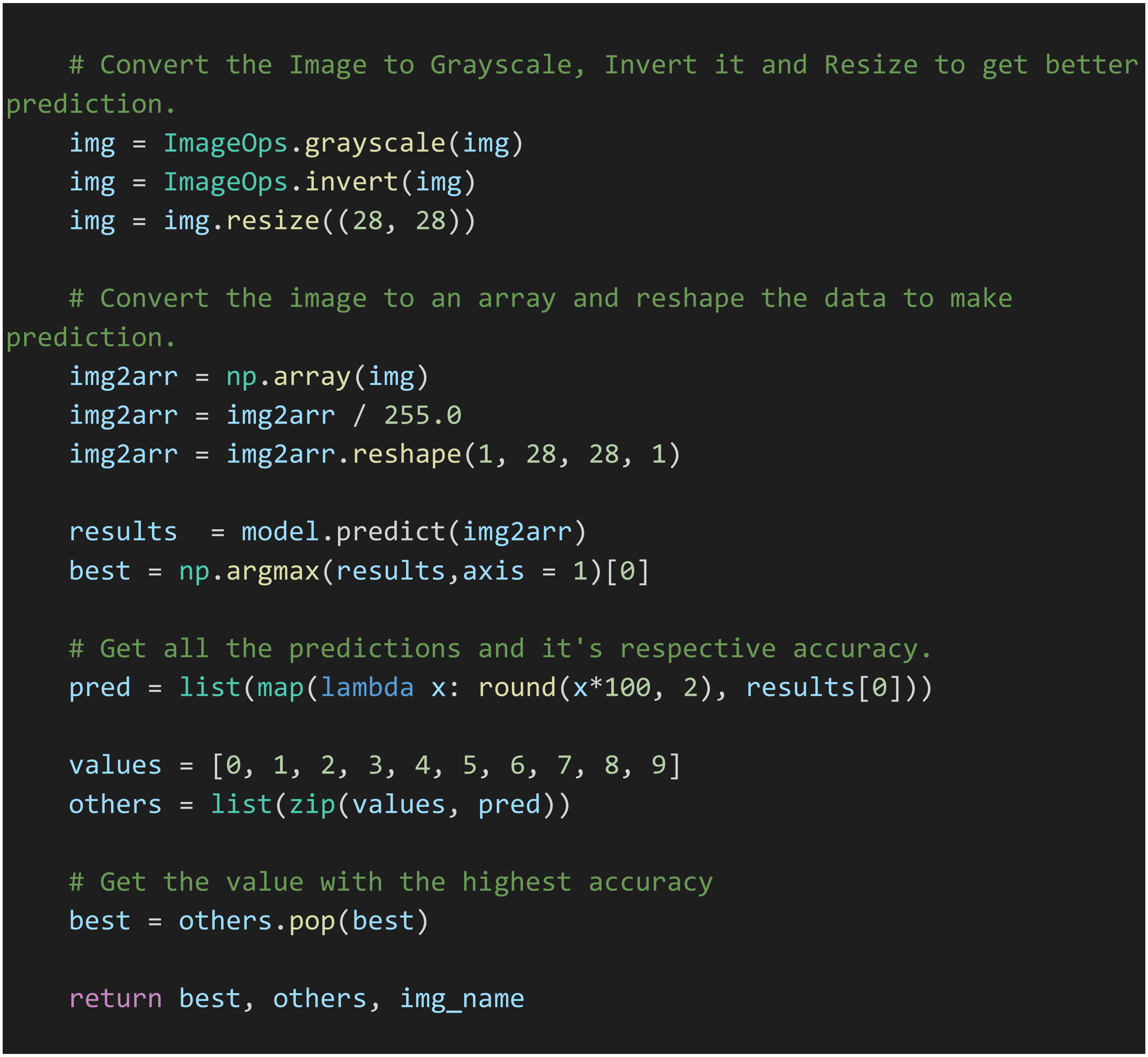
recognizer.py



```
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
```

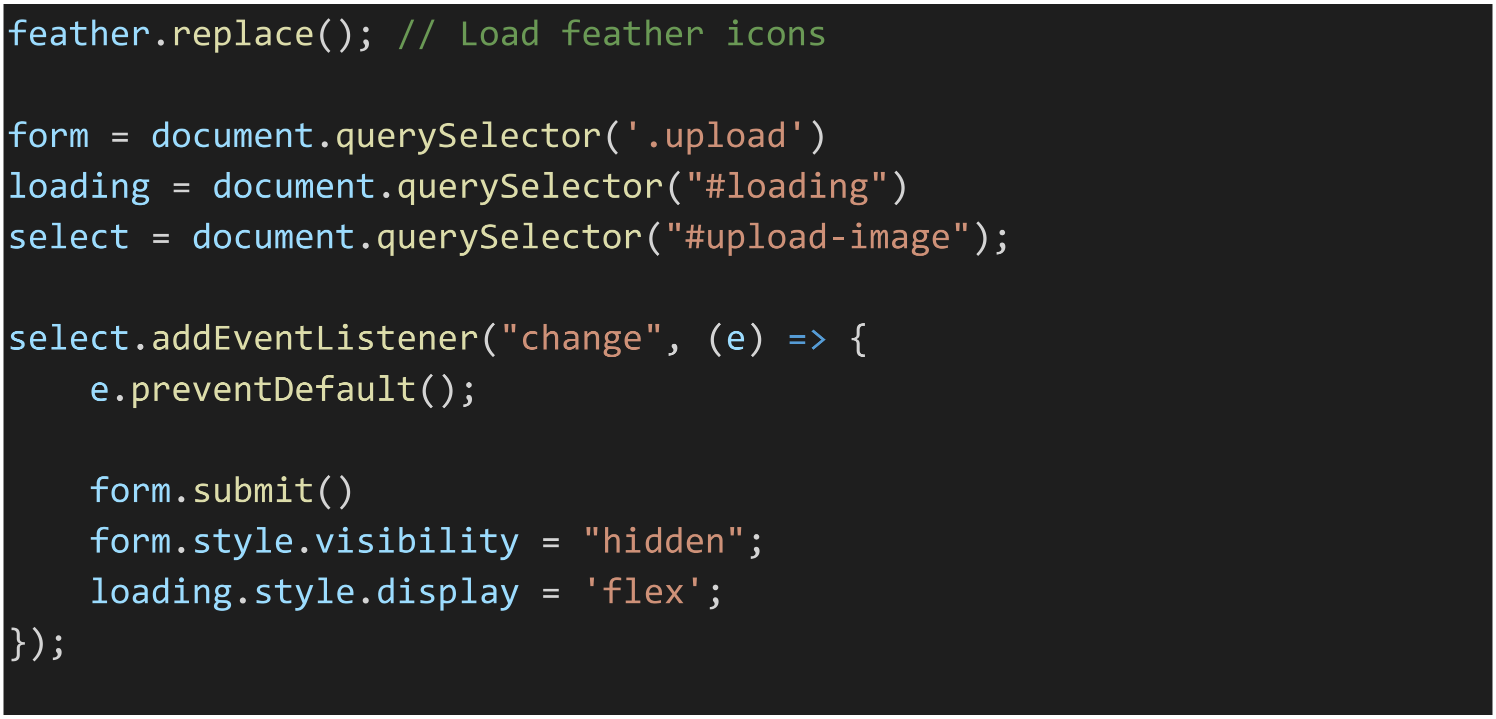


```python
def random_name_generator(n: int) -> str:
    """
    Generates a random file name.

    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
```

script.js



```
feather.replace(); // Load feather icons

form = document.querySelector('.upload')
loading = document.querySelector("#loading")
select = document.querySelector("#upload-image");

select.addEventListener("change", (e) => {
    e.preventDefault();

    form.submit()
    form.style.visibility = "hidden";
    loading.style.display = 'flex';
});
```

GITHUB

https://github.com/IBM-EPBL/IBM-Project-22680-1659856089

PROJECT DEMO

https://drive.google.com/file/d/1Bdr4rYBT7r8Dgh6JtRb0UTVdgcI1FQ84/view?usp=sharing