Project Report Format

1. **INTRODUCTION**
   - **1.1 Project Overview**
   - **1.2 Purpose**
2. **LITERATURE SURVEY**
   - **2.1 Existing problem**
   - **2.2 References**
   - **2.3 Problem Statement Definition**
3. **IDEATION & PROPOSED SOLUTION**
   - **3.1 Empathy Map Canvas**
   - **3.2 Ideation & Brainstorming**
   - **3.3 Proposed Solution**
   - **3.4 Problem Solution fit**
4. **REQUIREMENT ANALYSIS**
   - **4.1 Functional requirement**
   - **4.2 Non-Functional requirements**
5. **PROJECT DESIGN**
   - **5.1 Data Flow Diagrams**
   - **5.2 Solution & Technical Architecture**
   - **5.3 User Stories**
6. **PROJECT PLANNING & SCHEDULING**
   - **6.1 Sprint Planning & Estimation**
   - **6.2 Sprint Delivery Schedule**
   - **6.3 Reports from JIRA**
7. **CODING & SOLUTIONING (Explain the features added in the project along with code)**
   - **7.1 Feature 1**
   - **7.2 Feature 2**
   - **7.3 Database Schema (if Applicable)**
8. **TESTING**
   - **8.1 Test Cases**
   - **8.2 User Acceptance Testing**
9. **RESULTS**
   - **9.1 Performance Metrics**
10. **ADVANTAGES & DISADVANTAGES**
11. **CONCLUSION**
12. **FUTURE SCOPE**
13. **APPENDIX**
    - **Source Code**

GitHub & Project Demo Link

1.1.INTRODUCTION:

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 recognise handwritten digits from various sources, such as images, documents, and so on. This project aims to let users take advantage of machine learning to reduce manual tasks in recognizing digits.

1.2. Purpose

Handwritten digits recognition is a well-studied subfield of the field concerned with learning models to distinguish pre-segmented handwritten digits. It is one of the most important issues in data mining, machine learning, pattern recognition, and many other artificial intelligence disciplines. Handwritten digit datasets are ambiguous in nature because not all lines are sharp and perfectly straight.

2.LITERATURE SURVEY

2.1 What is the purpose of a handwritten recognition project?

In a fast paced digital world where data is consumed at such a rapid rate, it is essential that it should be in a compatible form given the rapid change in technology. Handwritten data will pose a hinderance in terms of this incompatibility in addition to being difficult to preserve. Hence it is essential that manual transcripts of data should be in digital format in order to tackle the problems stated.

Due to the large amount of handwritten data that needs to be digitalised, Artificial Intelligence became an effective tool to do this translation. Many methods have been studied to accurately translate manual transcripts to their digital versions.

In order to improve accuracy of the output, it is essential that new methods be further studied and implemented.

Methods studied

1. Image Segmentation based handwritten character recognition system
2. Character Segmentation using LSTM
3. Handwritten character recognition using ML algorithms
4. Handwriting Recognition using Unsupervised Learning

2.2 Methods explained

Image Segmentation based handwritten character recognition system

1. A publicly available dataset NIST which contains samples of handwritten characters from thousands of writers is used. The neural network model used is Convolutional Neural Networks (CNN).
2. CNN was trained using TensorFlow model, which is an open source library used for deep learning applications.
3. OpenCV, an open source library for image processing, was used to perform various operations like segmentation, thresholding and Morphological operations.
4. Obtained an accuracy of upto 94%.

Character Segmentation using LSTM

1. A publicly available dataset IAM Handwriting dataset was used which contains handwritten text of over 1500 forms including 5500+ sentences and 11500+ words.
2. An alternative approach to CNN where instead of classifying a dataset directly based on words, it is segmented based on characters and classified individually.
3. The characters are then reconstructed to form a word.
4. We used Tesseract LSTM with convolution model retrained on the aforementioned dataset.
5. Character level showed better test accuracy of 31% compared to word level which gave an average test accuracy of 22% (low accuracy due to a mini batch considered).

6. Yet it is not feasible to apply this method in practical applications due to imperfections in segmentation of the model.

Handwritten character recognition using ML algorithms

1. Digit dataset provided by Austrian Research Institute for Artificial Intelligence is used.

It contains 1893 training samples and 1796 testing samples.

2. In this method, different machine learning methods such as
    1. Multilayer Perceptron
    2. Support Vector Machine
    3. NaFDA5
    4. Naive Bayes
    5. Bayes Net
    6. Random Forest
    7. J48
    8. Random Tree
3. These models are used to classify the images and compared based on accuracy.
4. Multilayer Perceptron gave the best accuracy with a score of 90%.

Handwriting Recognition using Unsupervised Learning

1. A publicly available dataset IAM Handwriting dataset was used which contains handwritten text of over 1500 forms including 5500+ sentences and 11500+ words.
2. Hidden Markov Model, an unsupervised probabilistic model, is used for classification.
3. Character error rate and word error rates are compared and the former shows lesser error rates. The rates for both gradually decrease indicating better learning by the model.
4. Comparing these results to supervised learning systems, it is shown to be higher.
5. Hence supervised learning system is better.

2.3 Technologies needed for development

1. HTML/CSS/JavaScript/Bootstrap - Front end development
2. Python
3. TensorFlow
4. PyTorch
5. Image Processing Basics
6. Flask - Back end development
7. Git and GitHub - Project management
8. IBM Cloud - Hosting
9. IBM Watson - Training the Deep Learning Model

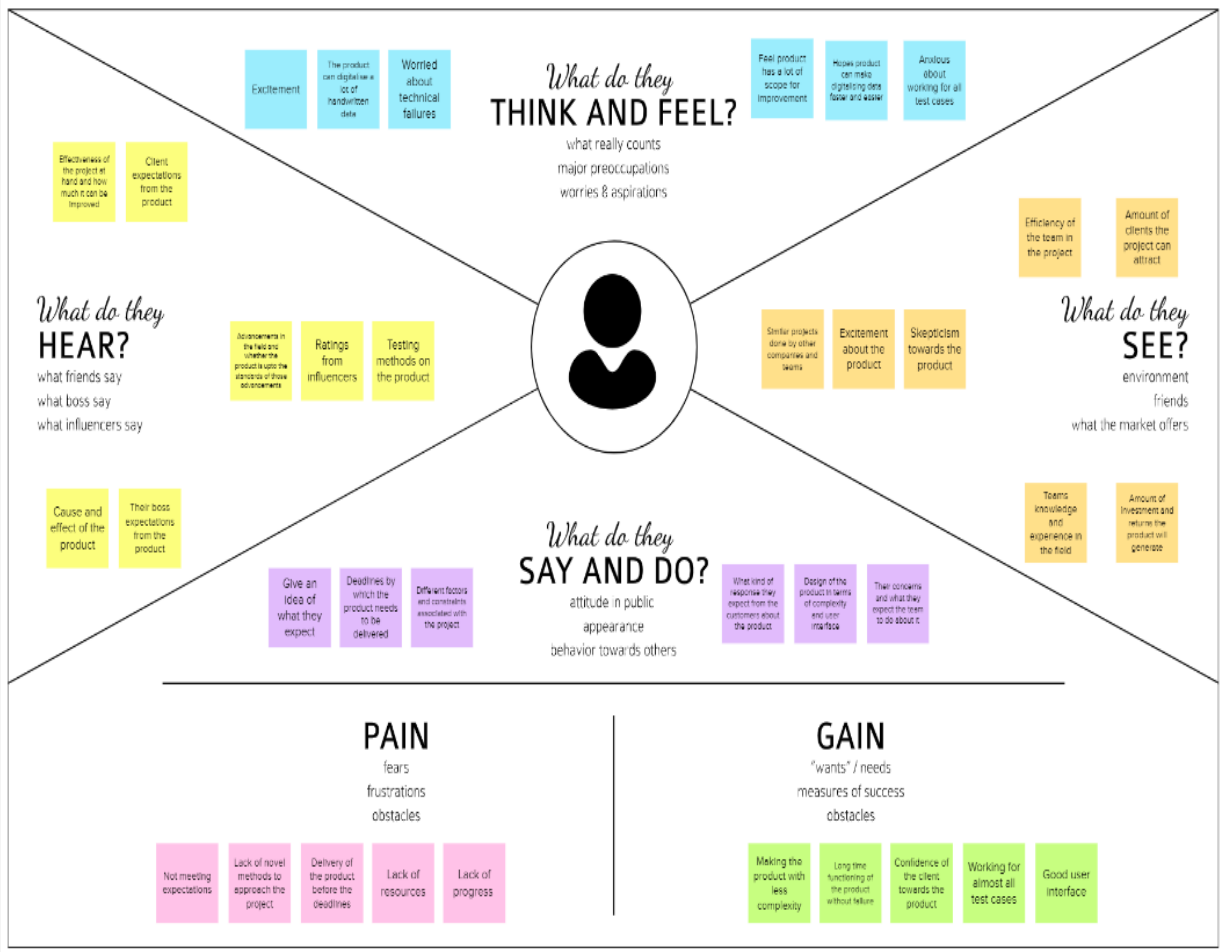
References

1. Image Segmentation based handwritten character recognition system : https://ieeexplore.ieee.org/abstract/document/8473291?casa_token=1Qitrq5Qj5YAAAAA:TQe6tCr1Ujonvl49Le4QnAMaxOfGuKLPzgpwlaHqZzI9cnquVudvxmRnleSV1ipcAbu7uC1FUQ
2. Character Segmentation using LSTM : http://cs231n.stanford.edu/reports/2017/pdfs/810.pdf
3. Handwritten character recognition using ML algorithms : https://computerresearch.org/index.php/computer/article/view/1685/1669
4. Handwriting Recognition using Unsupervised Learning : https://ieeexplore.ieee.org/abstract/document/6981077?casa_token=jcwM4VT6dmQAAAAA:b980YTZGAfVzgaqPKaN16Xfx9Xq90LyEgSnjlYri-bdrZqfF4LsuSnVV9OLfQ8zUjyUMEGr6GA

3.IDEATION AND PROPOSED SOLUTION

3.1 Empathy Map

Build empathy and keep your focus on the user by putting yourself in their shoes.



3.2 Ideation and Brainstorm

Brainstorm

Ideas related to Handwritten Digit Recognizer

10 minutes

Shreyas

Convert image to binary form by applying grayscale.	Use dropout layer to deactive neurons	Feed images directly to trained CNN
Image inversion could be done		

Sanjay

Detect horizontal and vertical edges	Part based method can be used	Use a 3 layered Neural Network
Reshaping of the image is possible		

Vasudhevan

Use different models to do a comparitive study	Use KNN to analyse the efficiency	Preprocess data to do dimentionality reduction
Thereshold the binary value obtained		

Aravind

Semi Incremental recognition model	Visualize data using mathPlotLib	Convert the result to uint8 range.
Result can be extracted from the transmitted image		

3.3 Proposed Solution

Project team shall fill the following information in proposed solution template.

S.No.	Parameter	Description
	Problem Statement (Problem to be solved)	In the era of digitalisation, handwritten data can be a huge hinderance. There is a huge need to convert handwritten numerical data into digital data. However doing this process manually can be daunting due to the large amount of handwritten data that needs to be digitalised. Hence a method for computers converting handwritten numerical data to digital data is required. Various machine learning methods are available for this purpose. An efficient model will be used to detect numerical data in images.
	Idea / Solution description	Our team aims to develop a Novel Handwritten Digit recognition system that allows user to pass an input image and get an accurate prediction of the digits present in the image. The application will be efficient in time and space and also detect complex digit patterns accurately.
	Novelty / Uniqueness	While many methods exist to detect numerical data in images, our method proposes using auto encoders along with a classifier for the purpose of detecting numerical data. Reasons for choosing the above is that the method is unsupervised, which makes training large amounts of data easier and also the features of the output image will not be necessary as we need only the predicted value of numerical data.
	Social Impact / Customer Satisfaction	This application will help organisations in effectively detecting numerical data in images as auto encoders have very low space complexity thereby allowing large amounts of data being easily processed for the organisation.

	Business Model (Revenue Model)	**Key partners:** The members of the team alongside SSN and IBM mentors will work towards the development of this application. **Key resources:** The resources for the development are obtained using our personal equipment, various IDE, IBM's database and software, image dataset(numerical handwritten data), college systems etc. **Activities:** The main activities include building an auto encoder model using Jupyter Notebook or Google Colab, developing an application for the model using Flask, interfacing it with IBM DB2, SendGrid, containerising the application, and hosting it on the cloud. **Value Proposition:** The users will be provided with a web application that has a friendly GUI and serves all the tasks of the application in a transparent manner. Security compliance will be strictly monitored to ensure that the user's data is safeguarded against any form of threats. **Cost Structure:** Cost is levied due to the usage of proprietary software. However, IBM's software is provided to us due to the fortunate initiative. Other such software that are non-IBM may add on to the expenses if inevitable. **Revenue Streams:** Subscription fees, unlocking premium features, expanding the storage of the application etc. **Customer Segments:** Students, Interested individuals, Family Members, Working Professionals, Organisations **Customer Relationships:** All the customer segments will be treated alike. Thus, all the users will be treated in a strictly professional manner, i.e every user will be treated in a fair manner, a prospective customer with no additional priorities etc. **Channels:** The application will be publicised through the usage of various social media platforms and through word of mouth. As users begin to use the application, ratings in Google, Play Store, and App Store would increase, resulting in a huge influx of customers.

3.4 Solution Fit

Project Title: A Novel Method for Handwritten Digit Recognition System

Project Design Phase-I - Solution Fit Template

Define CS, fit into CC

CUSTOMER SEGMENT(S) CS

Working professionals in banks and Post office form a major part of the customer base.

Students working in the field of data entry can also use this software.

A group of staffs working in speed post sorting form a part of the use case.

CUSTOMER CONSTRAINTS CC

Clerks in Aadhar office will be reluctant to use this as it is not 100% efficient and a single change in the digits will lead to huge problems.

Professionals need to have a powerful computer to run the Machine learning algorithm and run the GUI properly.

The algorithms should be run in the cloud to increase device compatibility and efficiency.

AVAILABLE SOLUTIONS AS

Available solutions include having a separate person in the bank and post office to look for handwritten digits in cheques, DDs and letters and note them accordingly.

Existing solutions just provide an algorithm to create a solution. There is no globally available website or app to use the functionality directly.

Explore AS, differentia

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

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

The most efficient way to identify hand written numbers is to have a person manually go through all the numbers.

This will lead to increased labor and costs. This task could be automated thus reducing the human intervention.

With predicting the hand-written digits, the trivial task of checking for numbers is bypassed and the process reaches the next level of execution.

PROBLEM ROOT CAUSE RC

It is not easy to manually check for hand written digits and record them accordingly.

The mundane method of checking manually leads to resource wastage and lesser efficiency.

Bank and Post Office clerks spend a lot of time for manually checking the cheque number and postal number and thus resulting in poor customer experience.

BEHAVIOUR BE

Users want an automated system to guess the handwritten digits and process the next steps automatically without human intervention.

They want a solution which is highly precise and efficient.

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

TRIGGERS TR

Reducing unwanted human labor and reducing the time for manually checking the digits will increase the efficiency of the tasks.

YOUR SOLUTION SL

Create a ML algorithm to efficiently sort and scale the images accordingly, to predict the hand-written digits with high precision. A model using CNN and MLP will be trained and shared globally across networks.

CHANNELS of BEHAVIOUR CH

ONLINE

The algorithms and its doing could be shared or published in an online site where users could share their reviews.

The idea could be published as a research paper and by doing so, making it available to public.

4. EMOTIONS: BEFORE / AFTER

EM

Before:

Users manually check for the handwritten digits and predict the digits using human knowledge.

After:

User just has to upload a pic of the handwritten digit and the user gets the output automatically

OFFLINE

Offline workshops or seminars could be conducted at college level to make the students aware about the machine learning algorithm.

The work staffs could be given an introductory demo about the usage of the new algorithm and thus making them ready for the near future.

4. REQUIREMENTS ANALYSIS

4.1 Functional Requirement

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Image Scanner	Scan an Image using the mobile camera
FR-2	Upload Image	Upload an image using the upload photo button
FR-3	Convert Image to Number	An option to convert an image into the required digit using the ML algorithms.
FR-4	Image Processor	Input image Get various sub-components
FR-7	An HTML Webpage	A local/cloud website where the algorithm runs in the background and provides the predicted output to the user

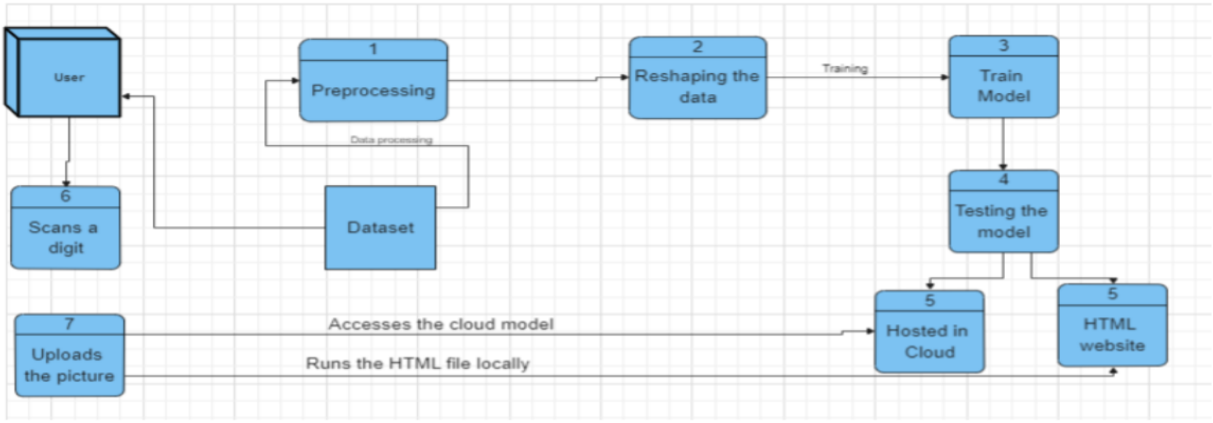
4.2 Non Functional Requirement

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	User-friendly UI
NFR-2	Security	Data integrity and safeness of data in DB
NFR-3	Reliability	The application will be consistent and reliable
NFR-4	Performance	Fast image processing and providing accurately predicted digits

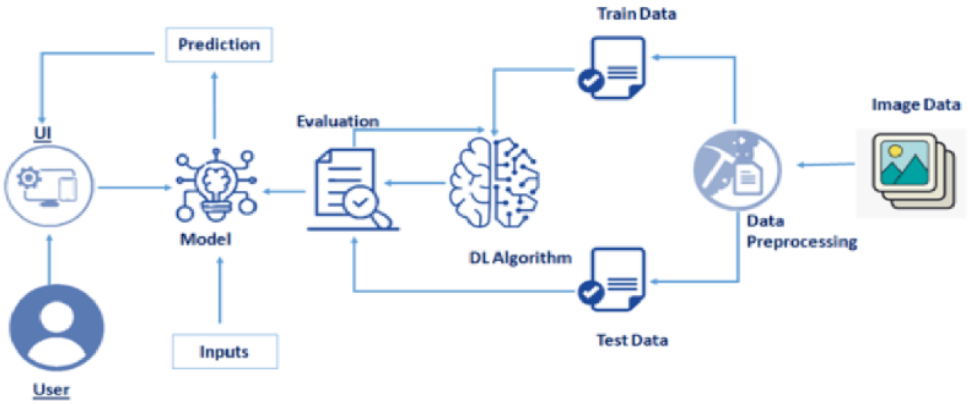
		Quick to provide the output for the user
NFR-5	**Availability**	**Available at all time for usage**
NFR-6	**Scalability**	**Large number of users can be using at the same time with proper functioning.**

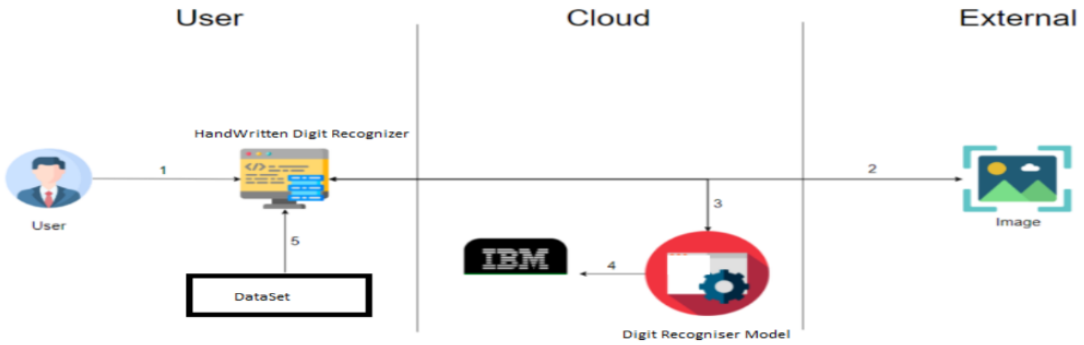
5. PROJECT DESIGN

5.1 Data Flow Design



5.2 Solution and Technology Architecture





5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Web User)	Upload Pic	USN-1	As a user, I can upload the picture from my laptop or desktop	I can access my desktop files	High	Sprint-3
		USN-2	As a user, I can see the mini document bar when I click on the upload button	I can see the document bar	Medium	Sprint-3
		USN-3	As a user, I can traverse through the list of folders and documents in my hard-drive	I can see the folders	High	Sprint-3
	Language	USN-4	As a user I should be able to view the website in the language of my choice.	I'm able to view the website after clicking translate	Low	Sprint-4
	Home Page	USN-5	The home page has the upload pic and the submit button where the user can upload pics.	I can see the upload and submit buttons	High	Sprint-2

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	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	5	Medium	Shreyas
		USN-2	As a user, I will receive confirmation email once I have registered for the application	1	Medium	Sanjay
Sprint-2	Model	USN-3	As a user, I want my ML model to be as accurate as possible.	5	Low	Vasudevan
		USN-4	As a user, I want good quality data to be collected for training purposes. Image processing methods must be employed to process the dataset.	5	Medium	Aravind
Sprint-1	Login	USN-5	As a user, I can login using email and password.	5	Medium	Shreyas
Sprint-4	Homepage	USN-6	As a user, the homepage must contain the details of input and an option to input an image in an acceptable format.	7	High	Aravind

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
		USN-7	As a user, I want my homepage to contain information regarding the functionalities of the application.	5	High	Sanjay
Sprint-4	Results Page	USN-8	As a user, I must be able to view the predicted output for the respective input image.	13	High	Aravind
Sprint-3	Hosting	USN-9	As a user, I must be able to host the application on mobile and PC in any operating system.	8	Medium	Vasudevan

6.2 Sprint Delivery Plan

Project Tracker, Velocity & Burndown Chart: (4 Marks)

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	10	6 Days	24 Oct 2022	29 Oct 2022	10	29 Oct 2022
Sprint-2	10	6 Days	31 Oct 2022	05 Nov 2022	10	05 Nov 2022
Sprint-3	8	6 Days	07 Nov 2022	12 Nov 2022		
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022		

7. CODING AND SOLUTIONING

7.1 Feature 1

```
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()
```

7.2 Feature 2

```
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
```

8. TESTING

8.1 Test Cases

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	Working as expected	PASS

				successfully		
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

8.2.1 DEFECT ANALYSIS

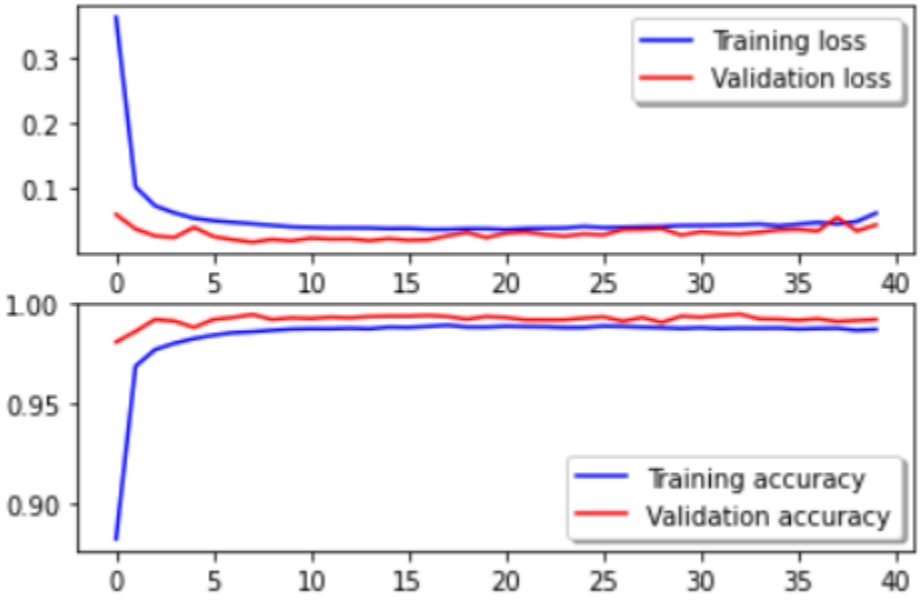
Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Total
By Design	1	0	1	0	2
Duplicate	0	0	0	0	0

8.2.2 TEST CASE ANALYSIS

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

9. RESULTS

9.1 Performance Metrics



Model Accuracy : 97.89

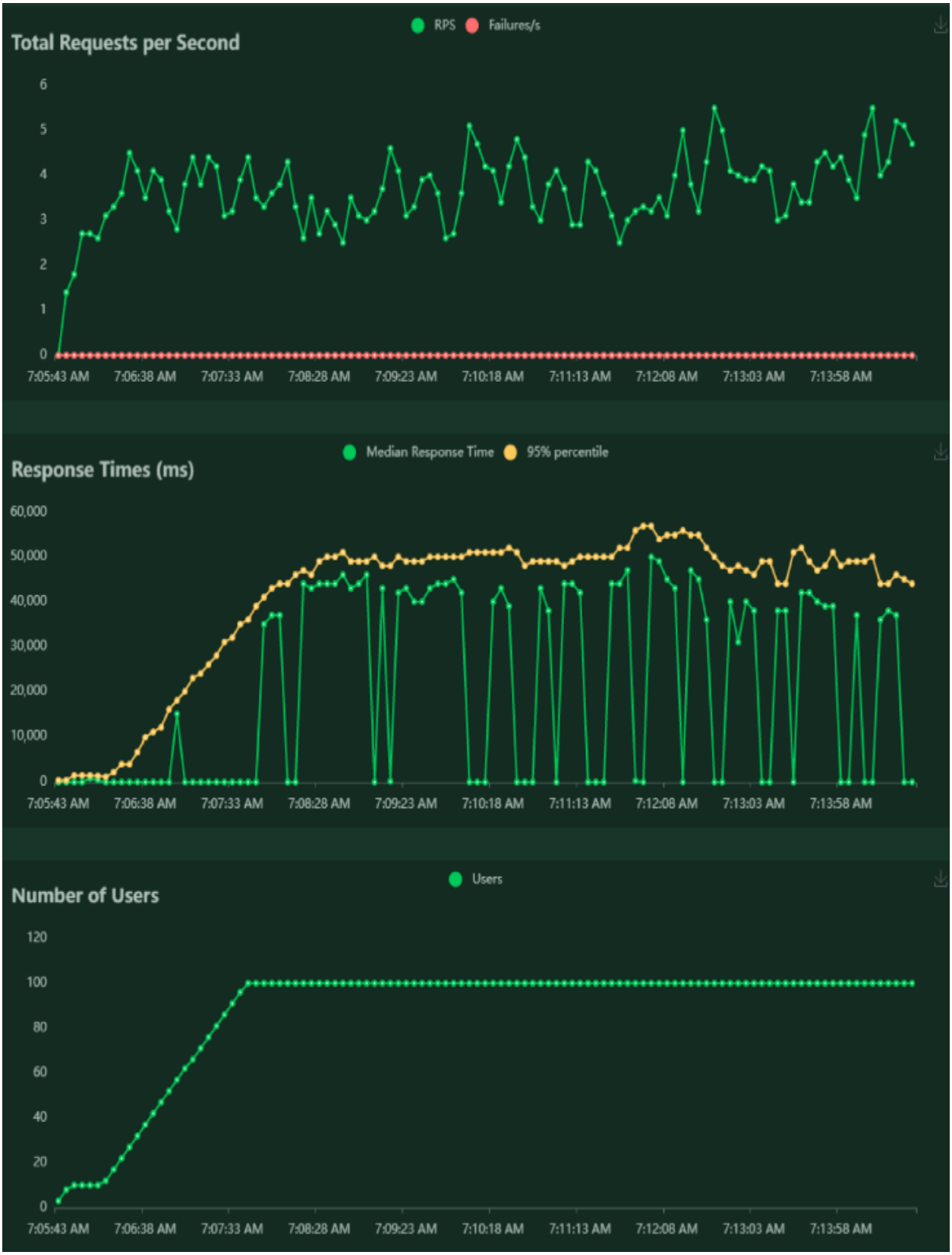
Locust Test Report

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**


RPS
Failures/s
Total Requests per Second
6
5
4
3
2
1
0
7:05:43 AM
7:06:38 AM
7:07:33 AM
7:08:28 AM
7:09:23 AM
7:10:18 AM
7:11:13 AM
7:12:08 AM
7:13:03 AM
7:13:58 AM
Median Response Time
95% percentile
Response Times (ms)
60,000
50,000
40,000
30,000
20,000
10,000
0
7:05:43 AM
7:06:38 AM
7:07:33 AM
7:08:28 AM
7:09:23 AM
7:10:18 AM
7:11:13 AM
7:12:08 AM
7:13:03 AM
7:13:58 AM
Users
Number of Users
120
100
80
60
40
20
0
7:05:43 AM
7:06:38 AM
7:07:33 AM
7:08:28 AM
7:09:23 AM
7:10:18 AM
7:11:13 AM
7:12:08 AM
7:13:03 AM
7:13:58 AM

10. ADVANTAGES & DISADVANTAGES

10.1 Advantages

- Reduces manual work
- More accurate than average human
- Capable of handling a lot of data
- Can be used anywhere from any device

10.2.Disadvantages

- Cannot handle complex data
- All the data must be in digital format
- Requires a high performance server for faster predictions
- Prone to occasional errors

11 . CONCLUSION

This project demonstrated a web application that recognises handwritten numbers using machine learning. This project was built with Flask, HTML, CSS, JavaScript, and a few other technologies. Using a CNN network, the model predicts the handwritten digit. During testing, the model achieved a recognition rate of 99.61%. The proposed project is easily scalable and can handle a large number of users.

It is compatible with any device that can run a browser because it is a web application. This project is extremely useful in real-world scenarios such as recognising vehicle licence plates,

processing bank cheque amounts, numeric entries in hand-written forms (tax forms), and so on. There is a lot of room for improvement that can be implemented in future 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.