A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

IBM NALAIYA THIRAN

PROJECT REPORT

SUBMITTED BY

TEAM ID : PNT2022TMID30996

HARIHARASUDHAN M	**(620819104032)**
NANDHA KISHORE B	**(620819104065)**
SANJAI RAJAA R T	**(620819104089)**
SOMA SURIYA D	**(620819104100)**

in partial fulfillment for the award of the

degreeof

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

GNANAMANI COLLEGE OF TECHNOLOGY, NAMAKKAL – 637 018

ANNA UNIVERSITY : : CHENNAI

600 025 NOVEMBER 2022

Table of Contents

1. INTRODUCTION

1.1 Project Overview

Handwritten digit recognition has recently been of very interest among the researchers because of the evolution of various Machine Learning, Deep Learning and Computer Vision algorithms. In this report, We compare the results of some of the most widely used Machine Learning Algorithms like CNN- convolution neural networks and with Deep Learning algorithm like multilayer CNN using Keras with Theano and TensorFlow. MNIST is a dataset which is widely used for handwritten digit recognition. The dataset consists of 60,000 training images and 10,000 test images. The artificial neural networks can all most mimic the human brain and are a key ingredient in image processing field. For example, Convolution Neural networks with back propagation for image processing. The applications where this handwritten digit recognition can be used are Banking sector where it can be used to maintain the security pin numbers, it can be also used for blind peoples by using sound output.

1.2 Purpose

By the end of the project we can:

- ✓ Know fundamental concepts and techniques of the Artificial Neural Network and Convolution Neural Networks.
- ✓ Know fundamental concepts and techniques of the python for machine learning.
- ✓ Gain a broad understanding of image data.
- ✓ Work with Sequential type of modelling.
- ✓ Work with Keras capabilities.
- ✓ Work with image processing techniques.
- ✓ Know how to build a web application using the Flask framework.

2. LITERATURE SURVEY

2.1 Existing Problem

- Huge variability and ambiguity of strokes from person to person
- Handwriting style of an individual person also varies time to time and is inconsistent.
- Poor quality of the source document/image due to degradation over time.
- Text in printed documents sit in a straight line whereas humans need not write a line of text in a straight line on white paper.
- Cursive handwriting makes separation and recognition of characters challenging.
- Text in handwriting can have variable rotation to the right which contrasts with printed text where all the text sits up straight.
- Collecting a good, labelled dataset to learn is not cheap compared to synthetic data.

2.2 References

1. **A NOVEL APPROCH FOR HANDWRITTEN DIGIT RECOGNITION USING MULTILAYER PERCEPTION NEUTRAL NETWORK [Toufik Datsi, Khalid Aznag, Ahmed El Oirrak, 2022]**

This paper is focused on field of Optical Character Recognition. It is also based on Artificial Neural Networks which are proved their effectiveness in the areas of image processing. It is about minimize the number of pixels by using as input the data extracted andcalculated from the initial image. The approach consists of transforming the image of the digit in the binary format then encode each column by value. The architecture of Artificial Neural Network used in this research is based on a multilayer perceptron neural network in order to recognize and predict the handwritten digit from 0 to 9. For better training and testing dataset, we have used the backpropagation as a learning algorithm. A dataset of 6000 sampleswas obtained from the MNIST database. For better training and testing dataset, we have usedthe backpropagation as a learning algorithm.

2. **A NOVEL HANDWRITTEN DIGIT CLASSIFICATION SYSTEMBASED ON CONVOLUTIONAL NEUTRAL NETWORK APPROACH [Ali Abdullah Yahya, Jieqing Tan, et al, 2021]**

An enormous number of CNN classification algorithms have been proposed in the literature. Nevertheless, in these algorithms, appropriate filter size selection, data preparation,limitations in datasets, and noise have not been taken into consideration. As a consequence, most of the algorithms have failed to make a noticeable improvement in classification accuracy. To address the shortcomings of these algorithms, our paper presents the following contributions: After taking the domain knowledge into consideration, the size of the effectivereceptive field (ERF) is calculated. Calculating the size of the ERF helps us to select a typicalfilter size which leads to enhancing the classification accuracy of our CNN. Unnecessary dataleads to misleading results and this, in turn, negatively affects classification accuracy. To
guarantee the dataset is free from any redundant or irrelevant variables to the target variable,data preparation is applied before implementing the data classification mission. To decrease the errors of training and validation, and avoid the limitation of datasets, data augmentation has been proposed to simulate the real-world problems.

3. **EFFECTIVE HANDWRITTEN DIGIT RECOGNITION USING CONVOLUTION NEURAL NETWORK [Yellapragada SS Bharadwaj, Rajaram P, et al, 2020]**

This paper proposed a simple neural network approach towards handwritten digit recognition using convolution. With machine learning algorithms like KNN, SVM/SOM, recognizing digits is considered as one of the unsolvable tasks due to its distinctiveness in thestyle of writing. In this paper, Convolution Neural Networks are implemented with an MNISTdataset of 70000 digits with 250 distinct forms of writings. The proposed method achieved 98.51% accuracy for real-world handwritten digit prediction with less than 0.1 % loss on training with 60000 digits while 10000 under validation. Advancements in the field of computer vision using deep neural networks attract attention; thus, many A.I. practitioners are moving towards it. One of the influencing projects that opted for deep learning is OCR. Handwritten digit recognition (HDR) is a snippet of OCR where instead of taking the wholecharacter's data, HDR detects digits. Comparing to OCR, HDR is light and faster. In fields like medical, banking, student management, and taxation process, HDR possesses great flexibility.

4. **IMPROVED HANDWRITTEN DIGIT RECOGNITION USING CONVOLUTUIONAL NEURAL NETWORK [Savita Ahlawat, Amit Choudhary, et al, 2020]**

Traditional systems of handwriting recognition have relied on handcrafted features and alarge amount of prior knowledge. Training an Optical character recognition (OCR) system based on these prerequisites is a challenging task. Research in the handwriting recognition field is focused around deep learning techniques and has achieved breakthrough performancein the last few years. Still, the rapid growth in the amount of handwritten data and the availability of massive processing power demands improvement in recognition accuracy anddeserves further investigation. Convolutional neural networks (CNNs) are very effective in perceiving the structure of handwritten characters/words in ways that help in automatic extraction of distinct features and make CNN the most suitable approach for solving handwriting recognition problems. Our aim in the proposed work is to explore the various

design options like number of layers, stride size, receptive field, kernel size, padding and dilution for CNN-based handwritten digit recognition. In addition, we aim to evaluate various SGD optimization algorithms in improving the performance of handwritten digit recognition.

5. **HANDWRITTEN DIGIT RECOGNITION USING ENSEMBLELEARNING [Kuppa Venkata Padmanabha Nandan, Manoj Panda, S. Veni, 2020]**

This paper is mainly focused on Ensemble learning. In pattern recognition, the recognition of handwritten digits has always been a very challenging and tedious task. In thiswork, a simple novel approach is proposed to recognize the handwritten digits. The primary goal of this work is recognition of the handwritten digits by using ensemble learning. Ensemble learning improves convergence by decreasing the complexity of the model to facilitate accurate and improved decision. This is also helpful to know about distribution of data in the random split and class-wise split. It's about analysis of how the load is distributedamong the base learners and how it impacts the model accuracy and training time. The overalltrends of the ensemble model have also been analysed in this paper.

6. **HYBRID CNN – SVM CLASSIFIER FOR HANDWRITTEN DIGIT RECOGNITION [Savita Ahlawat, Amit Choudhary,2020]**

The aim of this paper is to develop a hybrid model of a powerful Convolutional Neural Networks (CNN) and Support Vector Machine (SVM) for recognition of handwritten

digit fromMNIST dataset. The proposed hybrid model combines the key properties of both the classifiers.In the proposed hybrid model, CNN works as an automatic feature extractor and SVM works asa binary classifier. The MNIST dataset of handwritten digits is used for training and testing the algorithm adopted in the proposed model. The MNIST dataset consists of handwritten digits images which are diverse and highly distorted. The receptive field of CNN helps in automatically extracting the most distinguishable features from these handwritten digits. The experimental results demonstrate the effectiveness of the proposed framework by achieving a recognition accuracy of 99.28% over MNIST handwritten digits dataset. In the area of handwriting recognition, several methods have been proposed in the literature such as Artificial Neural Network (ANN), Neuro-Fuzzy Systems (NFS), Support Vector Machine (SVM) and deep learning-based classifiers [1-9]. Although decent recognition accuracy has been reported by theseclassifiers; handwriting digit recognition is still an open research problem and demands for exploring new techniques and methodologies that would further improve the performance in terms of recognition accuracy, running time and computational complexity.

7. MULTI-LANGUAGE HANDWRITTEN DIGIT RECOGNITION BASED ON NOVEL STRUCTURAL FEATURAL [Jaafar M. Alghazo, Ghazanfar Latif, 2019]

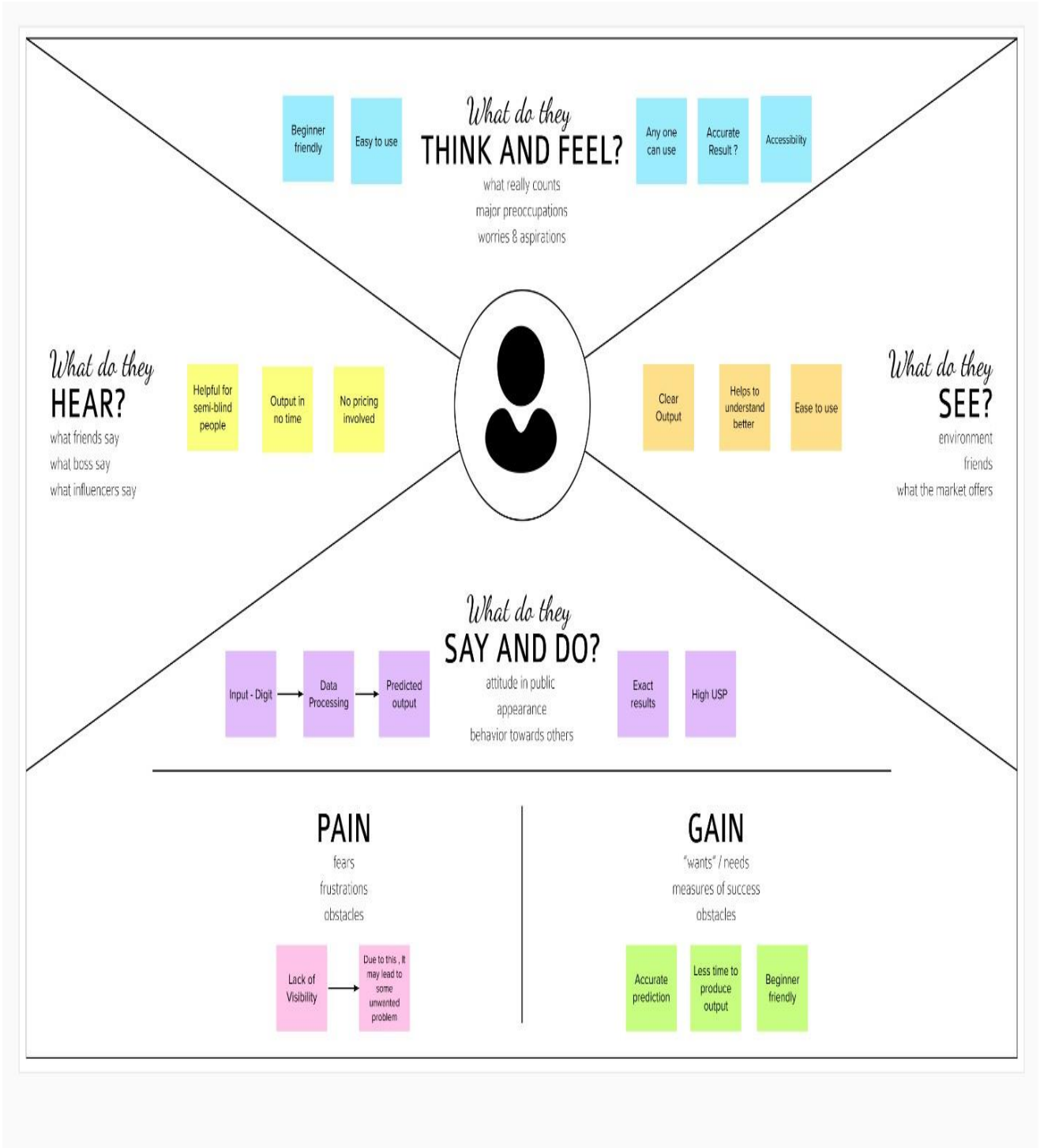
Automated handwritten script recognition is an important task for several applications. In this article, a multi-language handwritten numeral recognition system is proposed using novel structural features. A total of 65 local structural features are extracted and several classifiers are used for testing numeral recognition. Random Forest was found to achieve thebest results with an average recognition of 96.73%. The proposed method is tested on six different popular languages, including Arabic Western, Arabic Eastern, Persian, Urdu, Devanagari, and Bangla. In recent studies, single language digits or multiple languages with digits that resemble each other are targeted. In this study, the digits in the languages chosen do not resemble each other. Yet using the novel feature extraction method a high recognition accuracy rate is achieved. Experiments are performed on well-known available datasets of each language. A dataset for Urdu language is also developed in this study and introduced asPMU-UD. Results indicate that the proposed method gives high recognition accuracy as compared to other methods. Low error rates and low confusion rates were also observed usingthe novel method proposed in this study.

2.3 Problem Statement Definition

Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	Customer	Use software	It takes long time	Some issue in software	Disappointed
PS-2	Semi blind/Old people	find the number	Not sure whether result is correct or not	Not Cleareye sight	Uncertain
PS-3	Bank Employee	Conversio n to machine readable format	Not getting accuracy	Similarity indigits	Irritated
PS-4	Postal service clerk	Recogniz ezip code	Much time consumption	Struggling to read people handwriting	Depressed

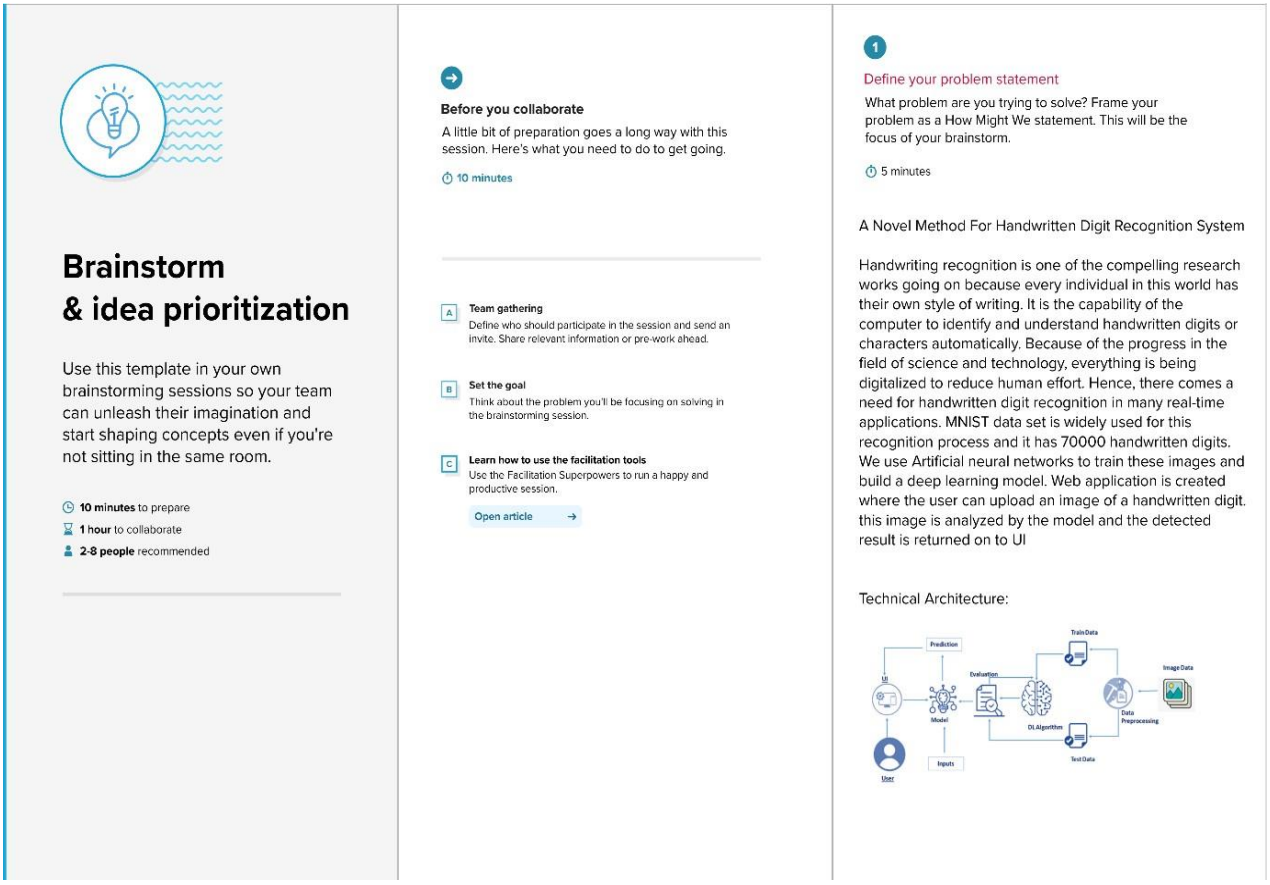
3. IDEATION & PROPOSSED SOLUTION

3.1 Empathy map canvas



3.2 Ideation & Brainstorm

Step-1: Team Gathering, Collaboration and Select the Problem Statement



Brainstorm & idea prioritization

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

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

Before you collaborate

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

10 minutes

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

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

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

Open article →

1

Define your problem statement

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

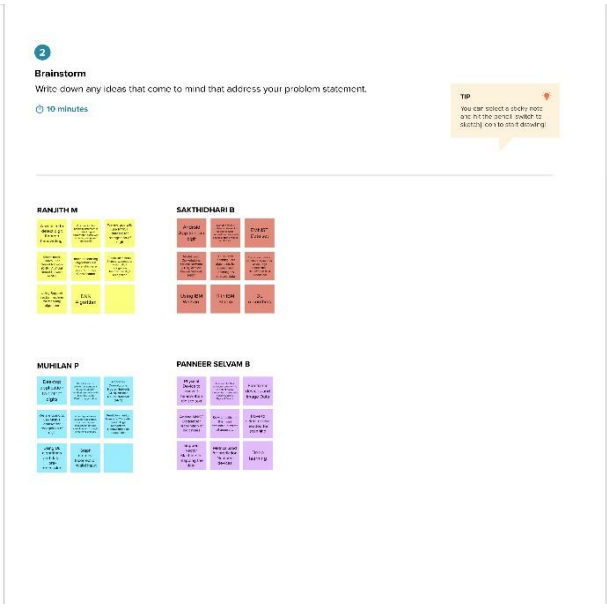
5 minutes

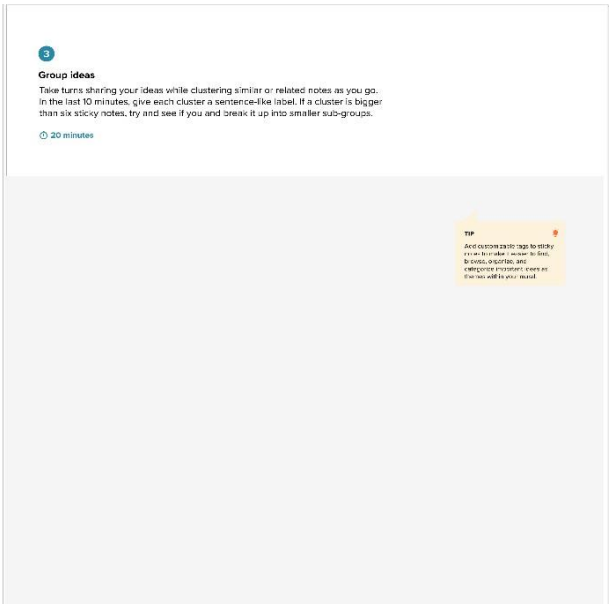
A Novel Method For Handwritten Digit Recognition System

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 digitalized to reduce human effort. Hence, there comes a need for handwritten digit recognition in many real-time applications. 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 on to UI

Technical Architecture:

Step-2: Brainstorm, Idea Listing and Grouping





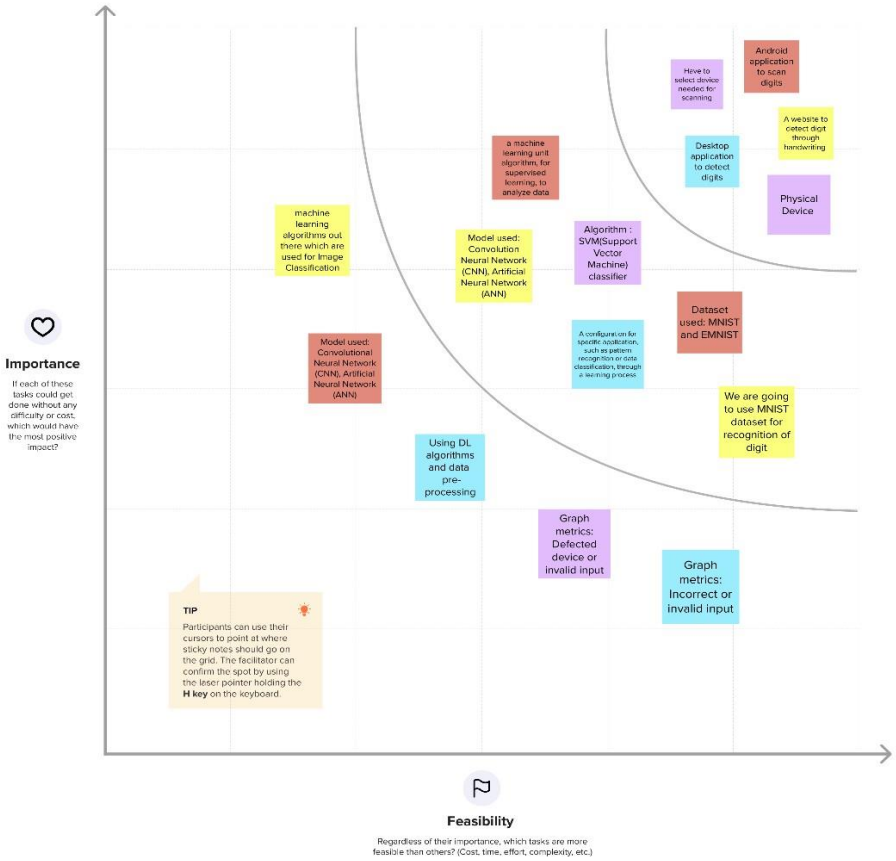
Step-3: Idea Prioritization

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.

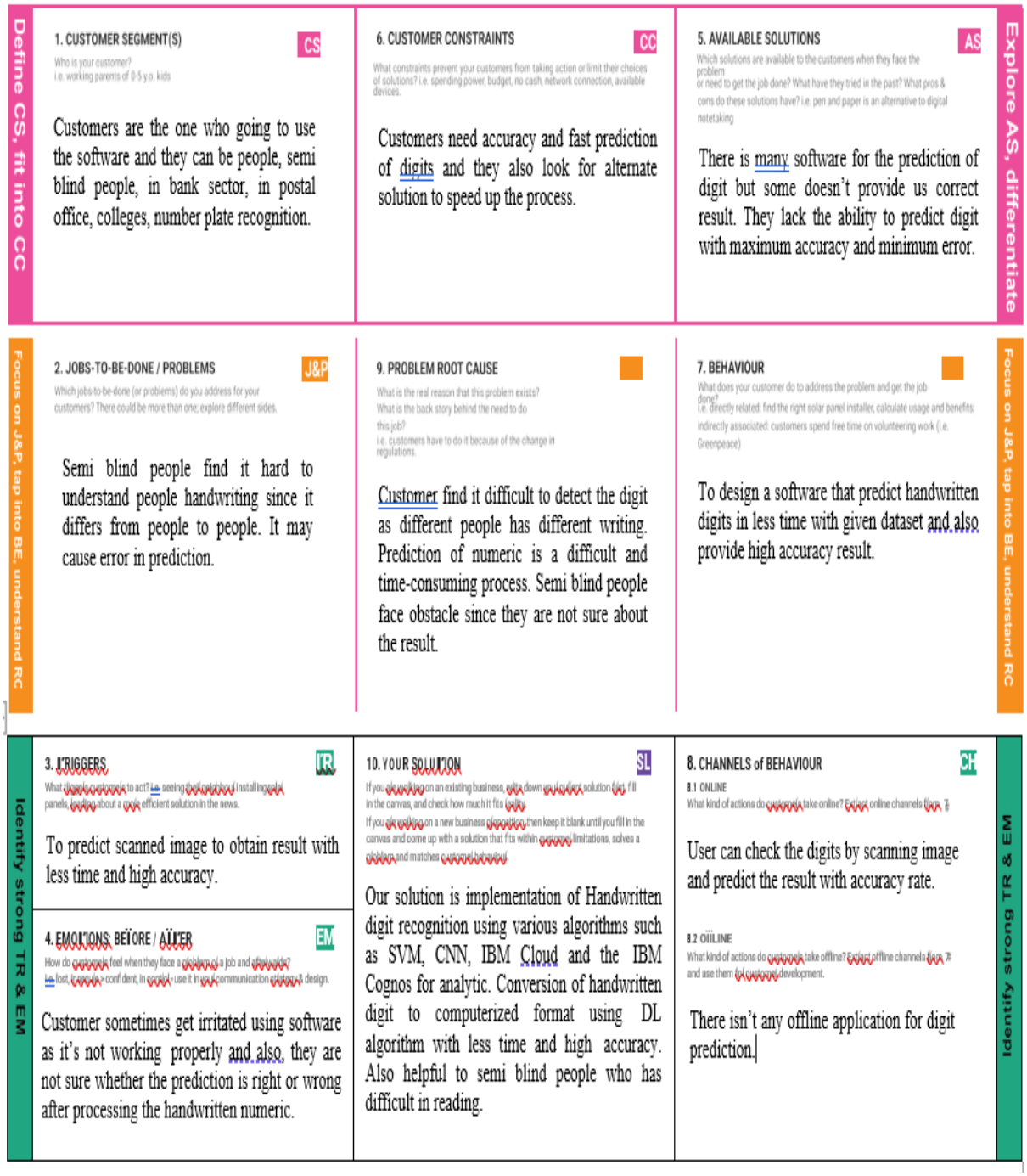
20 minutes



3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Human brain can adopt to work or task performance after few practices and they can also analysis image with accuracy. So, prediction of handwritten digit is performed with different algorithm for converting handwritten digit to digital form and attain maximum accuracy in less time. Prediction can be done with help of many algorithms for better analysis. Some find it difficult to understand people handwriting so digitalization is done to reduce the difficulty. More hundreds of datasets are used for recognition of handwritten digit for better analysis.
2.	Idea / Solution description	Implementation of Handwritten digit recognition using various algorithms such as SVM, CNN, IBM Cloud and the IBM Cognos for analytic. Conversion of handwritten digit to computerized format using DL algorithm.Recognition of digit with less time and high accuracy.
3.	Novelty / Uniqueness	The product will have high quality outcome within in no time. Also predict digits with high accuracy.
4.	Social Impact / Customer Satisfaction	Usually, semi-blinded people face the problem of identifying the exact digit wherein if they are using this software, they will be able to use it wisely.
5.	Business Model (Revenue Model)	Banking sector for recognition of account number from cheque. Can create a profitable business with research and expertise. Postal zip codes recognition in postal sector. Data entry in form through recognition of digit.
6.	Scalability of the Solution	This software can help in all the banking sectors such as private banks, Government banks, and Urban banks in which they will be able to figure out the exact digit in order to prevent fraudulent transactions. It not only solves problems in banks but it also helps many individuals.

3.4 Problem Solution fit



4. REQUIREMENT ANALYSIS

4.1 Functional Requirement

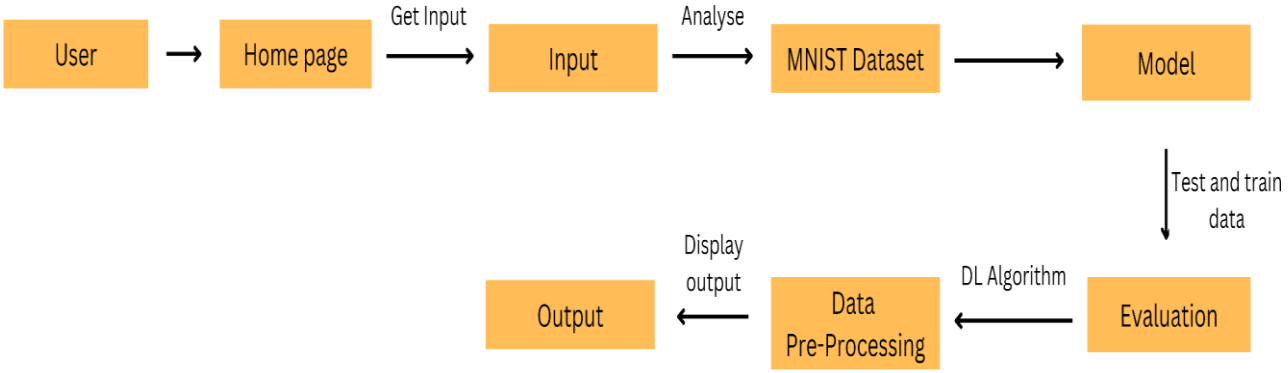
Functional Requirement No	Functional Requirement Description
FR-1	User Should be able to upload the image
FR-2	User should be able to preview the uploaded image
FR-3	Predicted number page

4.2 Non-Functional Requirements

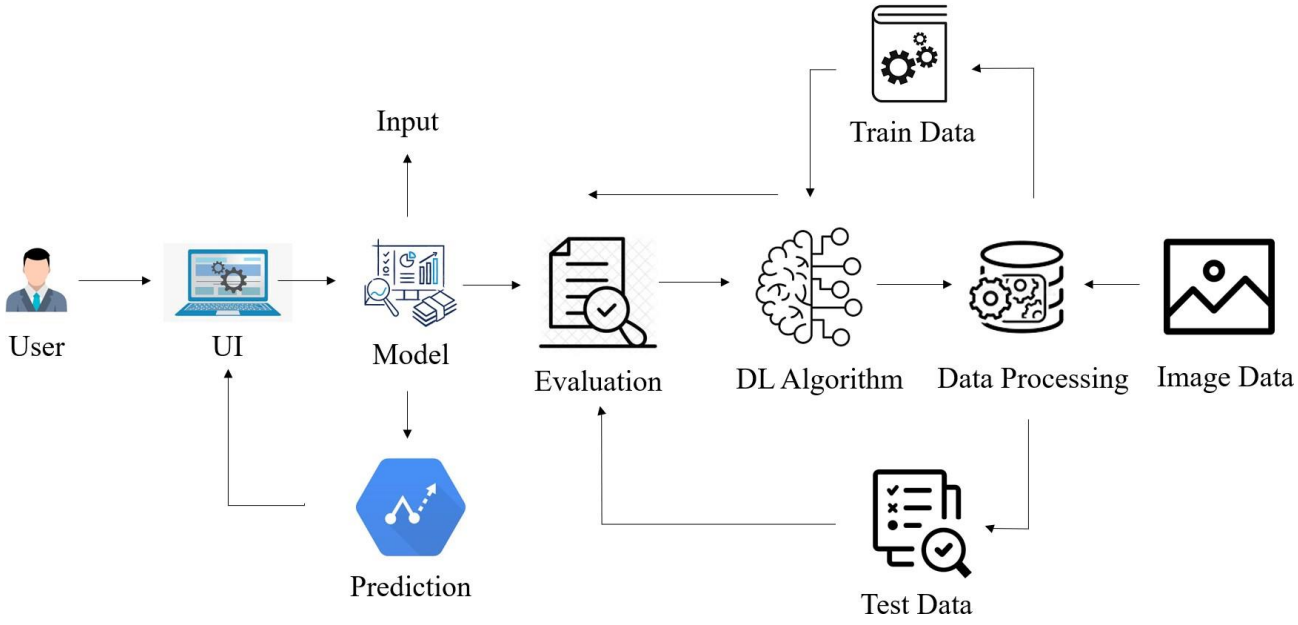
Non-Functional Requirement No	Non-Functional Requirement Description
NFR-1	The webpage should be load within 1.5 seconds
NFR-2	The uploaded data should be showing the name of the file
NFR-3	The website should be compatible independent of platform
NFR-4	There are 500+ users can able to access the webpage

5. PROJECT DESIGN

5.1 Data Flow Diagrams



5.2 Solution & Technical Architecture



5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Home page	USN-1	As a user, I can view user guidelines and it's functionality.	I can view the guidelines	Low	Sprint-1
		USN-2	As a user, I can read user manual to understand the process and workflow	I can read the user manual	Low	Sprint-2
		USN-3	As a user, I can watch a video that describeabout whole interface work	I can watch the video to understand the process ofhow to use the platform	Low	Sprint-1
	Input	USN-4	As a user, I can write the digits for prediction.	I can write digit that needto be predicted.	High	Sprint-1
	Recognition	USN-6	As a user, I will able to get the exact andaccurate output	I can choose handwritten image from system and predict the output	High	Sprint-2
	Predict	USN-7	As a user, I am allowed to upload handwrittenimage to predict output.	I can choose the imagefrom their own system	Medium	Sprint-3
		USN-8	As a user, I will get the output with the help of MNIST data pre-processing to ensure maximum accurate result.	MNIST dataset providesinput of handwritten digit to provide more insights.	High	Sprint-4
		USN-9	As a user, I can view the accuracy rate of thedigit predicted.	I can view accuracy rate	Medium	Sprint-3
Customer (Webuser)	Access	USN-10	As a user, I can get to use software virtually and it is user friendly.	I can view awareness of this application and its accessibility.	Low	Sprint-1

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data Collection	USN-1	As a user, I can collect the dataset from various resources with different handwritings.	10	Low	Hari Hara Sudhan M Sanjai Rajaa R T
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	Hari Hara Sudhan M Sanjai Rajaa R T
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	Nandha Kishore B
Sprint-2	Add CNN layers	USN-4	Creating the model and adding the input, hidden, and output layers to it.	5	High	Nandha Kishore B
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	Nandha Kishore B
Sprint-2	Train & test the model	USN-6	As a user, let us train our model with our image dataset.	6	Medium	Nandha Kishore B
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	Nandha Kishore B

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
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	Soma Suriya D
Sprint-3		USN-9	As a user, I can know the details of the fundamental usage of the application.	5	Low	Soma Suriya D
Sprint-3		USN-10	As a user, I can see the predicted / recognized digits in the application.	5	Medium	Soma Suriya D
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	Sanjai Rajaa R T
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	Sanjai Rajaa R T

6.2 Sprint Delivery Schedule

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

7. CODING AND SOLUTIONING

7.1 Feature 1

```
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\\nandh\IBM_NK\\Final Deliverables\\Final Code\\A-novel-method-for-digit-recognition-system\\flask_app\\uploads'


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

model = load_model("mnistCNN.h5")


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

```python
@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=False, threaded=False)
```

8. TESTING

8.1 Test Cases

Test case ID	Feature Type	Component	Test Scenario
Homepage_TC_OO1	Functional	Home Page	Verify user is able to see the Homepage when user enter into the website
Homepage_TC_OO2	UI	Home Page	Verify the UI elements in Homepage
Homepage_TC_OO3	Functional	Home page	Verify user is able to upload rather than a image file
Homepage_TC_OO4	Functional	Home Page	Verify user is able to upload a image files
PredictionPage_TC_OO1	Functional	PredictPage	Verify that the Page Shows the exact number as uploaded picture
PredictionPage_TC_OO2	Functional	PredictPage	Verify that the Page Shows the exact number as uploaded picture
PredictionPage_TC_OO3	Functional	PredictPage	Verify that the Page Shows the exact number as uploaded picture

PredictionPage_TC_OO4	Functional	PredictPage	Verify that the Page Shows the exact number as uploaded picture
PredictionPage_TC_OO5	Functional	PredictPage	Verify that the Page Shows the exact number as uploaded picture
PredictionPage_TC_OO6	Functional	PredictPage	Verify that the Page Shows the exact number as uploaded picture
PredictionPage_TC_OO7	Functional	PredictPage	Verify that the Page Shows the exact number as uploaded picture
PredictionPage_TC_OO8	Functional	PredictPage	Verify that the Page Shows the exact number as uploaded picture
PredictionPage_TC_OO9	Functional	PredictPage	Verify that the Page Shows the exact number as uploaded picture
PredictionPage_TC_OO10	Functional	PredictPage	Verify that the Page Shows the exact number as uploaded picture while uploading a blur image

Steps To Execute	Test Data	Expected Result	Actual Result	Status
1.Enter URL and click go 2.Verify that the routes to the specific wepage which shows the contents	https://github.com/IBM-EPBL/IBM-Project-47843-1660802851/tree/main/Final%20Deliverables/Final%20Code/A-novel-method-for-digit-recognition-system/flask_app/app.py	Homepage should display	Working as expected	Pass
1.Enter URL and click go 2.Click on My Account dropdown button 3.Verify Homepage with below UI elements : a.Contents b.upload button with Choose file c.A small area which showing the uploaded image d.Predict button	https://github.com/IBM-EPBL/IBM-Project-47843-1660802851/tree/main/Final%20Deliverables/Final%20Code/A-novel-method-for-digit-recognition-system/flask_app/app.py	Application should show below UI elements : a.Contents b.upload button with Choose file c.A small area which showing the uploaded image d.Predict button	Working as expected	Pass
1.Enter URL and click go 2.Click on Choose	NK.pdf	It should Predict the Number	Working as not expected	Fail

file button 3.upload a .pdf file 4.Press the predict button				
1.Enter URL and click go 2.Click on Choose file button 3.upload a 1.png file 4.Press the predict button	1.png	It should Predict the Number	Working as expected	Pass
1.Enter URL and click go 2.Click on Choose file button 3.upload a 1.png file 4.Press the predict button	1.png	It should Predict the Number as 1	Working as expected	Pass
1.Enter URL and click go 2.Click on Choose file button 3.upload a 2.png file 4.Press the	2.png	It should Predict the Number as 2	Working as expected	Pass

predict button				
1.Enter URL and click go 2.Click on Choose file button 3.upload a 3.png file 4.Press the predict button	3.png	It should Predict the Number as 3	Working as expected	Pass
1.Enter URL and click go 2.Click on Choose file button 3.upload a 4.png file 4.Press the predict button	4.png	It should Predict the Number as 4	Working as expected	Pass
1.Enter URL and click go 2.Click on Choose file button 3.upload a 5.png file 4.Press the predict button	5.png	It should Predict the Number as 5	Working as expected	Pass

1.Enter URL and click go 2.Click on Choose file button 3.upload a 6.png file 4.Press the predict button	6.png	It should Predict the Number as 6	Working as expected	Pass
1.Enter URL and click go 2.Click on Choose file button 3.upload a 7.png file 4.Press the predict button	7.png	It should Predict the Number as 7	Working as expected	Pass
1.Enter URL and click go 2.Click on Choose file button 3.upload a 8.png file 4.Press the predict button	8.png	It should Predict the Number as 8	Working as expected	Pass
1.Enter URL and click go 2.Click on Choose	9.png	It should Predict the Number as 9	Working as expected	Pass

file button 3.upload a 9.png file 4.Press the predict button				
1.Enter URL and click go 2.Click on Choose file button 3.upload a blur.png file 4.Press the predict button	blur.png	It should Predict the Number as 9	It returns the Number as 1	Fail

8.2 User Acceptance Testing

DEFECT ANALYSIS

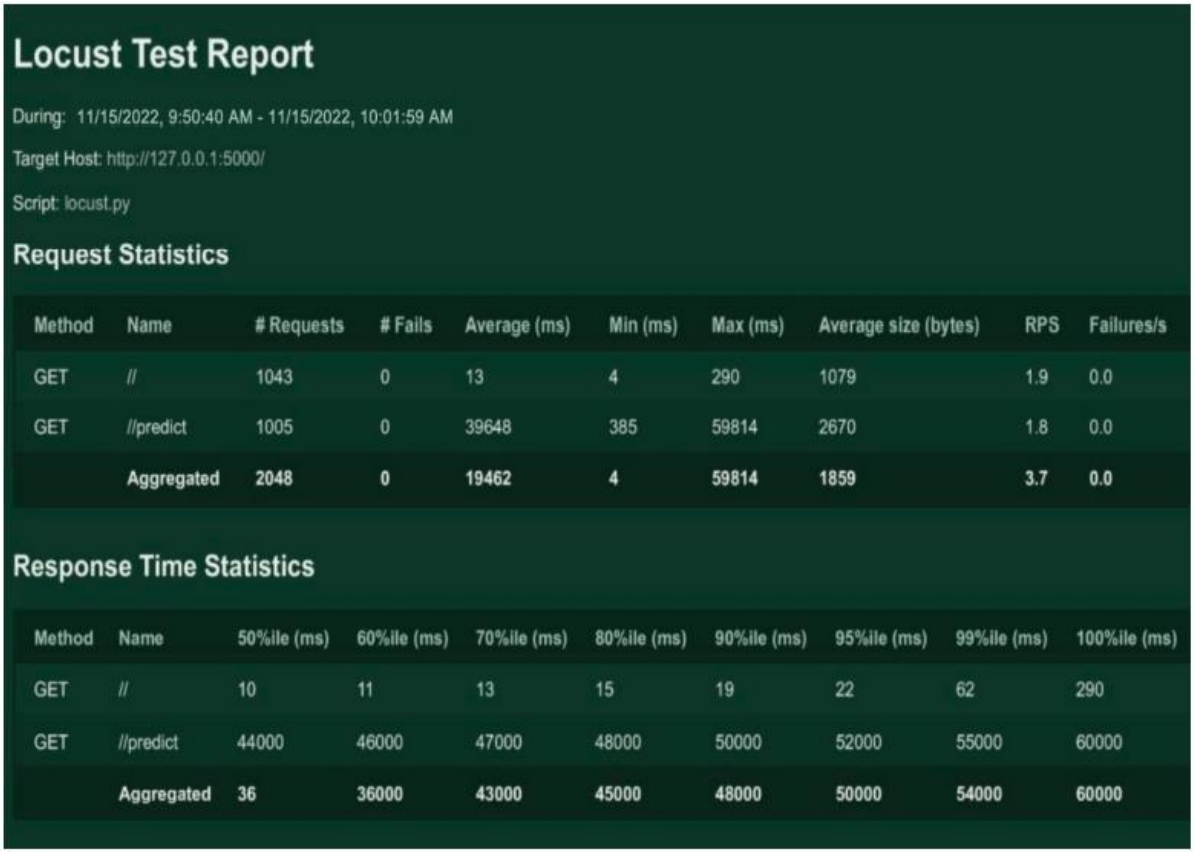
Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Total
By Design	1	0	1	0	2
Duplicate	0	0	0	0	0
External	0	0	2	0	2
Fixed	4	1	0	1	6
Not Reproduced	0	0	0	1	1
Skipped	0	0	0	1	1
Won't Fix	1	0	1	0	2
Total	6	1	4	3	14

TEST CASE ANAYSIS

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 Metrices



Locust Test Report

During: 11/15/2022, 9:50:40 AM - 11/15/2022, 10:01:59 AM

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**

Response Time Statistics

Method	Name	50%ile (ms)	60%ile (ms)	70%ile (ms)	80%ile (ms)	90%ile (ms)	95%ile (ms)	99%ile (ms)	100%ile (ms)
GET	//	10	11	13	15	19	22	62	290
GET	//predict	44000	46000	47000	48000	50000	52000	55000	60000
	Aggregated	**36**	**36000**	**43000**	**45000**	**48000**	**50000**	**54000**	**60000**


Charts
Total Requests per Second
RPS
Failures/s
6
5
4
3
2
1
0
9:50:40 AM
9:51:33 AM
9:52:38 AM
9:53:27 AM
9:54:22 AM
9:55:18 AM
9:58:27 AM
9:59:22 AM
10:00:18 AM
10:01:05 AM
Response Times (ms)
Median Response Time
95% percentile
60,000
50,000
40,000
30,000
20,000
10,000
0
9:50:40 AM
9:51:33 AM
9:52:38 AM
9:53:27 AM
9:54:22 AM
9:55:18 AM
9:58:27 AM
9:59:22 AM
10:00:18 AM
10:01:05 AM
Number of Users
Users
120
100
80
60
40
20
0
9:50:40 AM
9:51:33 AM
9:52:38 AM
9:53:27 AM
9:54:22 AM
9:55:18 AM
9:58:27 AM
9:59:22 AM
10:00:18 AM
10:01:05 AM

10. ADVANTAGES AND DISADVANTAGES

ADVANTAGES

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

DISADVANTAGES

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

11. CONCLUSION

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

12. FUTURE SCOPE

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

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

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

13. APPENDIX

SOURCE CODE

MODEL CREATION

Importing the required libraries

```
[46]
import numpy
import tensorflow #open source used for both ML and DL for computation
from tensorflow.keras.datasets import mnist #mnist dataset
from tensorflow.keras.models import Sequential #it is a plain stack of layers
from tensorflow.keras import layers #A Layer consists of a tensor- in tensor-out computat ion funct ion
from tensorflow.keras.layers import Dense, Flatten #Dense-Dense Layer is the regular deeply connected r
#faltten -used fot flattening the input or change the dimension
from tensorflow.keras.layers import Conv2D #onvoLutiona l Layer
from keras.optimizers import Adam #opt imizer
from keras. utils import np_utils #used for one-hot encoding
import matplotlib.pyplot as plt   #used for data visualization
```

load data

```
[47]
(x_train, y_train), (x_test, y_test)=mnist.load_data () #splitting the mnist data into train and test
```

+ Code + Markdown

```
[48]
print (x_train.shape)  #shape is used for give the dimens ion values #60000-rows 28x28-pixels
print (x_test.shape)
```

```
(60000, 28, 28)
(10000, 28, 28)
```

```python
x_train[0]
```

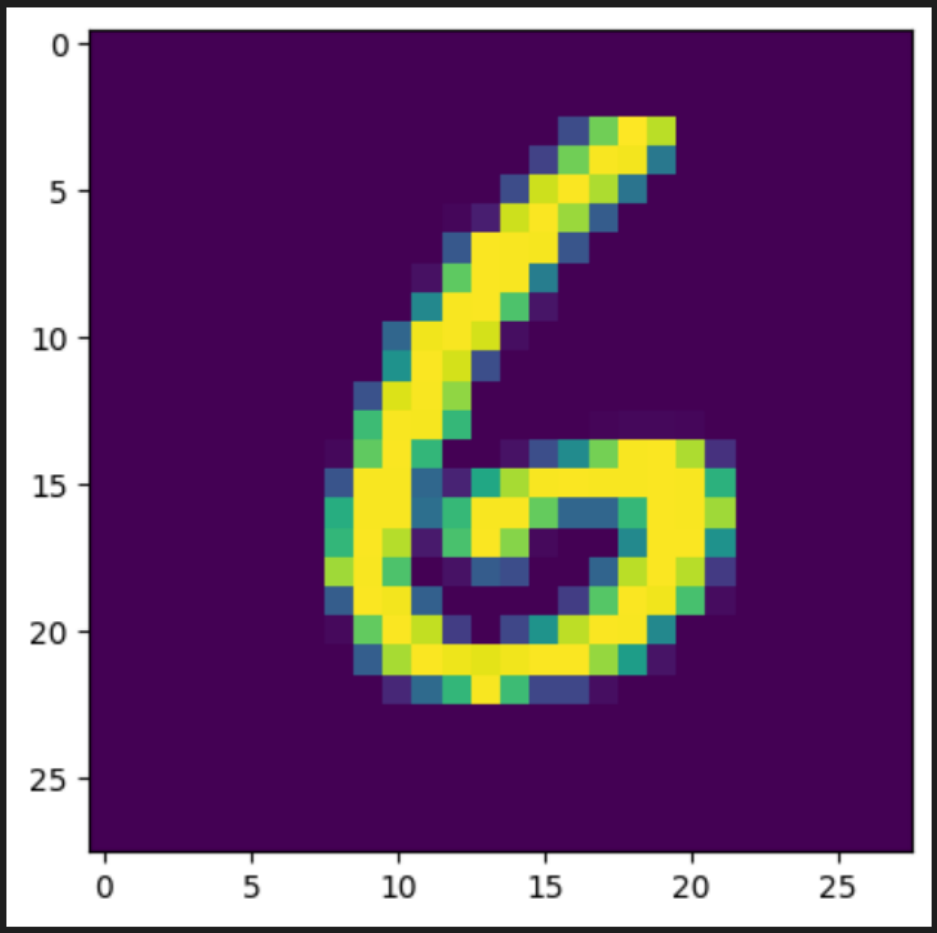
[49]

```
Output exceeds the size limit. Open the full output data in a text editor
array([[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   3,
         18,  18,  18, 126, 136, 175,  26, 166, 255, 247, 127,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,  30,  36,  94, 154, 170,
        253, 253, 253, 253, 253, 225, 172, 253, 242, 195,  64,   0,   0,
```

```
[50]  plt.imshow(x_train[6000])     #ploting the index=image
```

```
...  <matplotlib.image.AxesImage at 0x222c71e0250>
```

</>




```
[51]  np.argmax(y_train[6000])
```

Reshaping Dataset

[52]
```
#Reshaping to format which CNN expects (batch, height, width, channels)
x_train=x_train.reshape (60000, 28, 28, 1).astype('float32')
x_test=x_test.reshape (10000, 28, 28, 1).astype ('float32')
```

Applying One Hot Encoding

[53]
```
number_of_classes = 10  #storing the no of classes in a variable
```

[54]
```
y_train = np_utils.to_categorical (y_train, number_of_classes) #converts the output in binary format
y_test = np_utils.to_categorical (y_test, number_of_classes)
```

Add CNN Layers

[55]
```
#create model
model=Sequential ()
```

[56]
```
#adding modeL Layer
model.add(Conv2D(64, (3, 3), input_shape=(28, 28, 1), activation='relu'))
model.add(Conv2D(32, (3, 3), activation = 'relu'))
```

[57]
```
#flatten the dimension of the image
model.add(Flatten())
```

[58]
```
#output layer with 10 neurons
model.add(Dense(number_of_classes,activation = 'softmax'))
```

Compiling the model

[59]
```
#Compile model
model.compile(loss= 'categorical_crossentropy', optimizer="Adam", metrics=['accuracy'])
```

[62]
```
x_train = np.asarray(x_train)
y_train = np.asarray(y_train)
```

Train the model

[63]
```
#fit the model
model.fit(x_train, y_train, validation_data=(x_test, y_test), epochs=5, batch_size=32)
```

```
Epoch 1/5
1875/1875 [==============================] - 88s 47ms/step - loss: 0.2299 - accuracy: 0.9506 - val_loss: 0.0899 - val_accuracy: 0.9730
Epoch 2/5
1875/1875 [==============================] - 81s 43ms/step - loss: 0.0682 - accuracy: 0.9796 - val_loss: 0.0765 - val_accuracy: 0.9780
Epoch 3/5
1875/1875 [==============================] - 80s 43ms/step - loss: 0.0493 - accuracy: 0.9838 - val_loss: 0.0888 - val_accuracy: 0.9755
Epoch 4/5
1875/1875 [==============================] - 80s 43ms/step - loss: 0.0357 - accuracy: 0.9887 - val_loss: 0.0835 - val_accuracy: 0.9791
Epoch 5/5
1875/1875 [==============================] - 89s 48ms/step - loss: 0.0288 - accuracy: 0.9907 - val_loss: 0.1145 - val_accuracy: 0.9708

<keras.callbacks.History at 0x222d90d9db0>
```

Observing the metrics

```
[64]
# Final evaluation of the model
metrics = model.evaluate(x_test, y_test, verbose=0)
print("Metrics (Test loss &Test Accuracy) : ")
print(metrics)
```

```
Metrics (Test loss &Test Accuracy) :
[0.1144733875989914, 0.97079998254776]
```

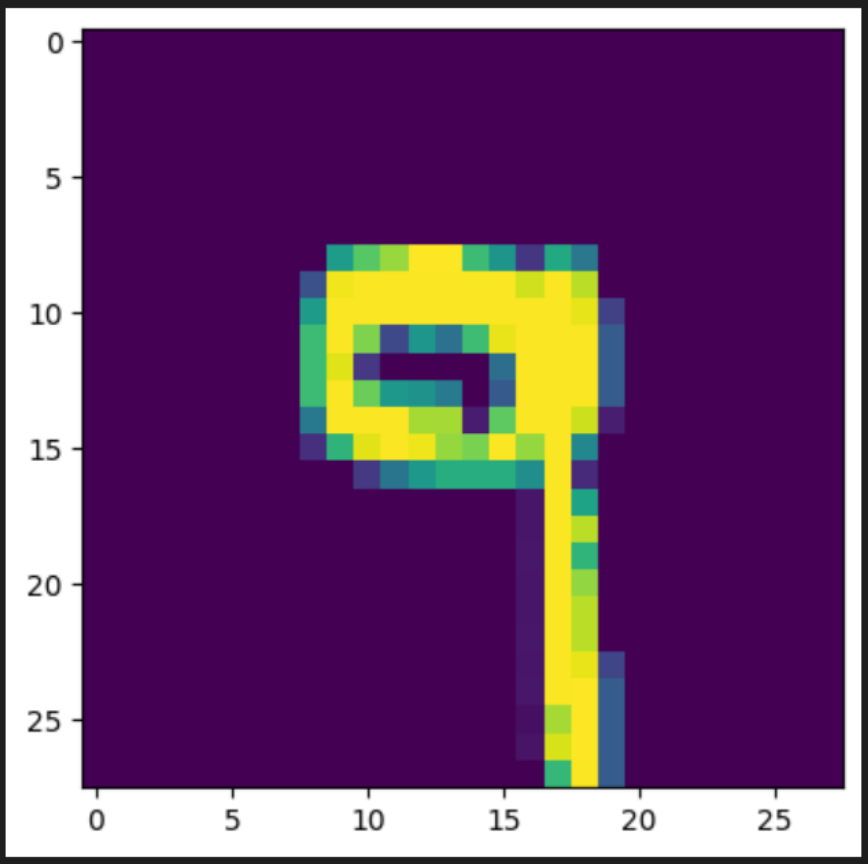
Test The Model

```
[65]
prediction=model.predict(x_test[6000:6001])
print(prediction)
```

```
1/1 [==============================] - 0s 84ms/step
[[1.4689527e-14 1.8748821e-17 2.3109615e-13 9.5624716e-07 3.1683821e-04
  2.4785629e-07 6.1842380e-18 6.2185841e-06 8.8746788e-07 9.9967492e-01]]
```

```
[66]
plt.imshow(x_test[6000])
```

```
<matplotlib.image.AxesImage at 0x222c9987430>
```




```
[67]
import numpy as np
print(np.argmax(prediction, axis=1)) #printing our Labels from first 4 images
```

```
[9]
```

```
[68]
np.argmax(y_test[6000:6001]) #printing the actual labels
```

```
9
```

Save The model

```
[70]
# Save the model
model.save('models/mnistCNN.h5')
```

FLASK APP

```
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\\nandh\IBM_NK\\Final Deliverables\\Final Code\\A-novel-method-for-digit-recognition-system\\flask_app\\uploads'


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

model = load_model("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=False, threaded=False)
```

Index.html

```
<html>

<head>
  <title>Digit Recognition WebApp</title>

  <meta name="viewport" content="width=device-width">
  <!-- GoogleFont -->
  <link href="https://fonts.googleapis.com/css2?family=Prompt:wght@600&display=swap" rel="stylesheet">
  <link href="https://fonts.googleapis.com/css2?family=Varela+Round&display=swap" rel="stylesheet">
  <link href="https://fonts.googleapis.com/css2?family=Source+Code+Pro:wght@500&display=swap" rel="stylesheet">
  <link href="https://fonts.googleapis.com/css?family=Calistoga|Josefin+Sans:400,700|Pacifico&display=swap" rel="stylesheet">
  <!-- bootstrap -->
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">
  <link rel="stylesheet" type= "text/css" href= "{{ url_for('static',filename='css/style.css') }}">
  <!-- fontawesome -->
  <script src="https://kit.fontawesome.com/b3aed9cb07.js" crossorigin="anonymous"></script>

  <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo" crossorigin="anonymous"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js" integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1" crossorigin="anonymous"></script>
  <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js" integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM" crossorigin="anonymous"></script>
  <script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@latest"></script>

</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',"");
          });
        });
```

```
<body>

  <h1 class="welcome">IBM PROJECT
  <div id="team_id">TEAM ID : PNT2022TMID30996</div>
  </h1>
  <section id="title">
    <h4 class="heading">Handwritten Digit Recognition Website</h4>
    <br><br>
      <p>
        The website is designed to predict the handwritten digit.
      </p>
    <p>
        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 digitalized to reduce human effort.</p>

      <br>
    <p> Hence, there comes a need for handwritten digit recognition in many real-time applications.
        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 on to UI</p>

  </section>

  <section id="content">

        <div class="leftside">
        <form action="/predict" method="POST" enctype="multipart/form-data">
        <label>Select a 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">  Clear  </button>
          </div>
        </form>
        </div>
  </section>

</body>

</html>
```

Index.css

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

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

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

.welcome{
  text-align: center;
  position: relative;
  color: honeydew;
  background-color: greenyellow;
  padding-top: 1%;
  padding-bottom: 1%;
  font-weight: bold;
  font-family: 'Prompt', sans-serif;
}

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

```css
#predict_button{
  margin-right: 15px;
  color: blue;
  font-weight: bold;
}

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

#result{
  font-size: 5rem;
}

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

.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;
}
```

```css
.heading{
  font-family: 'Varela Round', sans-serif;
  font-weight: 700;
  font-size: 2rem;
  display: inline;
}

.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%; */
}

p{
  font-family: 'Source Code Pro', monospace,sans-serif;
  margin-top: 1%;
}

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

Predict.html

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

<style>
    body{
     background-image: url('static/images/index6.jpg');
     background-repeat: no-repeat;
     background-size: cover;
    }

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

    #ans{
  text-align: center;
  font-size: 40px;
  margin: 0 auto;
  padding: 3% 5%;
  padding-top: 15%;
  color: white;
    }
```

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

Git hub link : https://github.com/IBM-EPBL/IBM-Project-47843-1660802851

Project Demo link : https://github.com/IBM-EPBL/IBM-Project-47843-1660802851/tree/main/Final%20Deliverables/Demo%20video