A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

A PROJECT REPORT

Submitted by

AJAY M	**830119104002**
SACHIN SIYAM P	**830119104036**
SURYA R	**830119104048**
TAMIL ARASU S	**830119104049**

Team id: **PNT2022TMID47342**

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

GOVERNMENT COLLEGE OF ENGINEERING SRIRANGAM

INDEX

1. INTRODUCTION

1.1 Project Overview

Handwritten digit recognition is the ability of a computer system to recognize the handwritten inputs like digits, characters etc. from a wide variety of sources like emails, papers, images, letters etc. This has been a topic of research for decades. Some of the research areas include signature verification, bank check processing, postal address interpretation from envelopes etc. With the humanization of machines, there has been a substantial amount of research and development work that has given a surge to deep learning and machine learning along with artificial intelligence. Handwriting recognition system is the most basic and an important step towards this huge and interesting area of Computer Vision. With time, machines are getting more and more sophisticated, from calculating the basic sums to doing retina recognition they have made our lives more secure and manageable. This project illustrates handwritten digit recognition with the help of MNIST datasets using Convolution Neural Network (CNN) models. Deep Learning has emerged as a central tool for self-perception problems like understanding images, voice from humans, robots exploring the world. The project aims to implement the concept of Convolution Neural Network which is one of the important architectures of deep learning. Understanding CNN and applying it to the handwritten recognition system, is the major target of the proposed system. The main objective of this project is to the accuracy of the models stated above along with their execution time to get the best possible model for digit recognition.

1.2 Purpose

Images of handwritten digits as 10 digits (09). Handwritten digits from the MNIST database are already famous among the community for many recent decades now, as decreasing the error rate with different classifiers and parameters. Digit recognition system is the working of a machine to train itself or recognizing the digits from different sources like emails, bank cheque, papers, images, etc. and in different real-world scenarios for online handwriting recognition on computer tablets or system, recognize number plates of vehicles, processing bank cheque amounts, numeric entries in forms filled up by hand (say tax forms) and so on. The

handwritten digits are not always of the same size, width, orientation and justified to margins as they differ from writing of person to person, so the general problem would be while classifying the digits due to the similarity between digits such as 1 and 7, 5 and 6, 3 and 8, 2 and 5, 2 and 7, etc. This problem is faced more when many people write a single digit with a variety of different handwritings. Lastly, the uniqueness and variety in the handwriting of different individuals also influence the formation and appearance of the digits. In the current age of digitization, handwriting recognition plays an important role in information processing. A lot of information is available on paper, and processing of digital files is cheaper than processing traditional paper files.Handwritten digit recognition for banking systems aims at ensuring effective and reliable approaches for recognition of handwritten digits.The aim of a handwriting recognition system is to convert handwritten characters into machine readable formats. Handwritten digit recognition has not only professional and commercial applications, but also has practical application in our daily life and can be of great help to the visually impaired. It also helps us to solve complex problems easily thus making our lives easier.

2. LITERATURE SURVEY

2.1 Existing problem

As handwriting varies from person to person, handwritten numbers do not always have the same height, weight, orientation, or marginal justification. The general problem is to distinguish the digits due to the similarity of digits such as 1 and 7, 5 and 6, 3 and 8, 2, five, two and seven. We would like to produce a model that transforms the handwritten digits into a typical type in various types so that no misunderstanding is created. 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. The challenge concerning the shape recognition problem such as handwritten character recognition remain in finding features that maximize the interclass variability while minimizing the intra-class variability. Feature extraction methods can be categorized into two classes: Structural features, which extract geometrical and topological properties such as the number and position of dots, the presence of loops, the orientation of curves...etc. Statistical features, such as histograms of projection profile and transitions, moments, histograms of gray level distribution, Fourier descriptors and chain code...etc After applying some pre-processing technique and normalizing the image, zoning and crossing counts are combined to represent the feature set. Self-organized map is used to cluster the classes and for creation of binary decision tree. For each node the classifier among SVM, KNN and neural networks who gives the best recognition rate is considered as the main classifier of the node .

2.2 References

Title: Real Time Handwritten Digits Recognition Using Convolutional Neural Network.
Author:Kaveti Upender; Venkata Siva Kumar Pasupuleti.
Reading handwritten information like examination answer sheets is still a difficult task for many of us, because each one of us is having a different interpretation style. As the world is moving towards digitization, converting the handwritten information to a readable digital format reduces the difficulty. This approach will be beneficial for the readers as it gives a better understanding of the information. With the help of machine learning and deep learning algorithms, the handwritten patterns can be recognized and classify them accordingly to a digital format with human level accuracy. This research paper deals with predicting the real time handwritten digits only. To classify the handwritten digits MNIST data set is used for training the model. OpenCV python library is used for detecting the patterns in the real time handwritten digits. These detected patterns are predicted to human level accuracy with the help of a Convolutional Neural Network model.

Title: Handwritten English Character and Digit Recognition(2021)
Author:Al-Mahmud; Asnuva Tanvin; Sazia Rahman
one of the most sought-after technologies is a handwritten character recognition system. It has the potential to solve a wide range of issues and bring about radical change in our lives. We used Convolutional Neural Networks (CNNs) to recognize handwritten English capital letters and digits in this research. We improved a previously developed CNN architecture by adjusting hyperparameters and minimizing the model's overfitting. The MNIST digit dataset is used to evaluate the experiments, which are then compared to different methods. On the MNIST dataset, 99.47 percent test accuracy was attained,which is superior to other approaches. The research was then expanded upon by the addition of a new dataset for recognizing English capital letters. 98.94 percent accuracy was achieved on this extended dataset.

Title: An Efficient And Improved Scheme For Handwritten Digit Recognition Based On Convolutional Neural Network (2019)
Author: Ali, Saqib and Shaukat, Zeeshan and Azeem, Muhammad and Sakhawat, Zareen and Mahmood, Tariq and others
This study uses rectified linear units (ReLU) activation and a convolutional neural network (CNN) that incorporates the Deeplearning4j (DL4J) architecture to recognize handwritten digits. The proposed CNN framework has all the necessary parameters for a high level of MNIST digit classification accuracy. The system's training takes into account the time factor as well. The system is also tested by altering the number of CNN layers for additional accuracy verification. It is important to note that the CNN architecture consists of two convolutional layers, the first with 32 filters and a 5x5 window size and the second with 64 filters and a 7x7 window size. In comparison to earlier proposed systems, the experimental findings show that the proposed CNN architecture for the MNIST dataset demonstrates great performance in terms of time and accuracy. As a result, handwritten numbers are detected with a recognition rate of 99.89% and high precision (99.21%) in a short amount of time.

Title: Handwritten Digit Recognition Using Machine And Deep Learning Algorithms (2021)
Author: Pashine, Samay and Dixit, Ritik and Kushwah, Rishika
In this study, they developed three deep and machine learning-based models for handwritten digit recognition using MNIST datasets. To determine which model was the most accurate, they compared them based on their individual properties.Support vector machines are among the simplest classifiers, making them faster than other algorithms and providing the highest training accuracy rate in this situation. However, due to their simplicity, SVMs cannot categorize complicated and ambiguous images as accurately as MLP and CNN algorithms can. In their research, they discovered that CNN produced the most precise outcomes for handwritten digit recognition. This led them to the conclusion that CNN is the most effective solution for all types of prediction issues, including those using picture data. Next, by comparing the execution times of the algorithms, they determined that increasing the number of epochs without changing the

configuration of the algorithm is pointless due to the limitation of a certain model, and they discovered that beyond a certain number of epochs, the model begins overfitting the dataset and provides biased predictions

2.3 Problem Statement Definition

➢ The problem is recognizing the human handwritten digits by system. The goal is to upload the image of the handwritten digit and identify the digit with accuracy.

➢ Digits' are a part of our everyday life, be it License plates on our cars or bikes, the price of a product, speed limit on a road, or details associated with a bank account. The recognition of digits is important because humans can't remember all these numbers ,so there is a need for a system to recognize them.

➢ Handwritten digit recognition is the ability of computers to recognize human handwritten digits. It is a hard task for the machine because handwritten digits are not perfect and can be made with many different flavors.

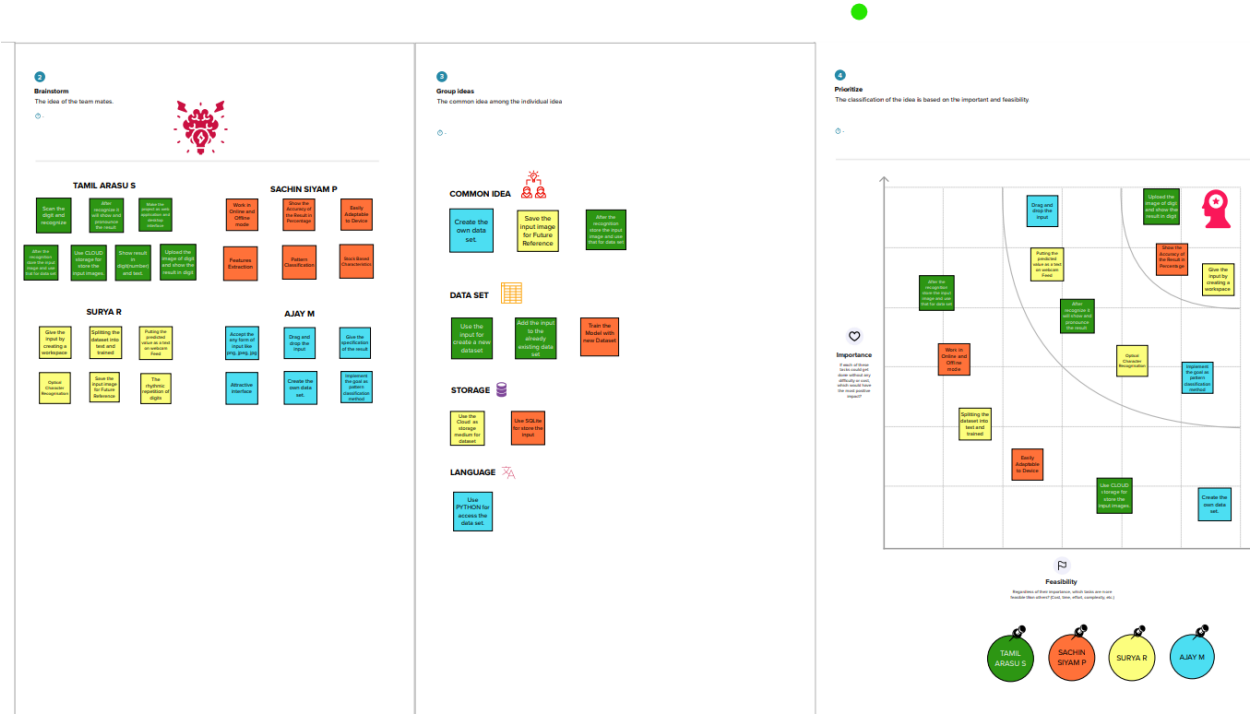
➢ The main objective is to compare the accuracy of the models stated above along with their execution time to get the best possible model for digit recognition.

➢ The digits are used in postal mail sorting, bank check processing, form data entry, etc. For this type of work there is a need for a system which can handle the data (digits) easily as well as accurately.

3. IDEATION & PROPOSED SOLUTION

3.1Empathy Map Canvas

3.2 Ideation and Brainstorming

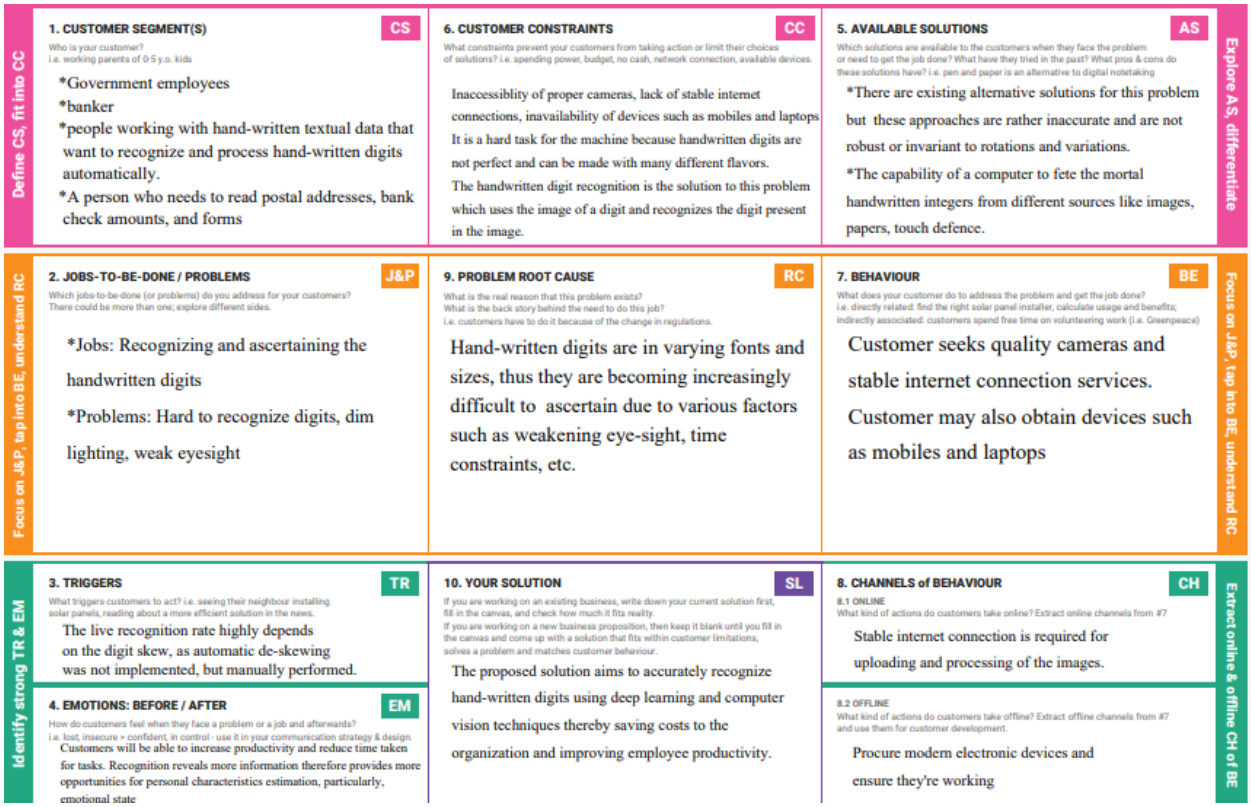


3.3 Proposed Solution

S.No	PARAMETER	DESCRIPTION
1.	Problem Statement	The problem is recognizing the human handwritten digits by system. The goal is to upload the image of the handwritten digit and identify the digit with accuracy.
2.	Idea/Solution Description	The training and testing has been conducted from publicly available MNIST handwritten databases.Web based, offline and online handwritten digitrecognition system is developed by usingConvolutional Neural Network.
3.	Novelty/Uniqueness	OCR technology provides higher than 99% accuracy with typed characters in high-quality images. However, the diversity in human writing, spacing differences, and irregularities of handwriting causes less accurate character recognition, as you can see in the featured image.

4.	Social Impact/Customer Satisfaction	Handwritten Digit Recognition has various usessuch as less time consumption. It is used in the detection of vehicle numbers, banks for reading cheques, post offices for arranging letters, and other tasks.
5.	Business Model	The main objective of this work is to ensure effective and reliable approaches for recognition of handwritten digits and make banking operations

3.4 Problem Solution Fit



4. REQUIREMENT ANALYSIS

4.1 Functional Requirements

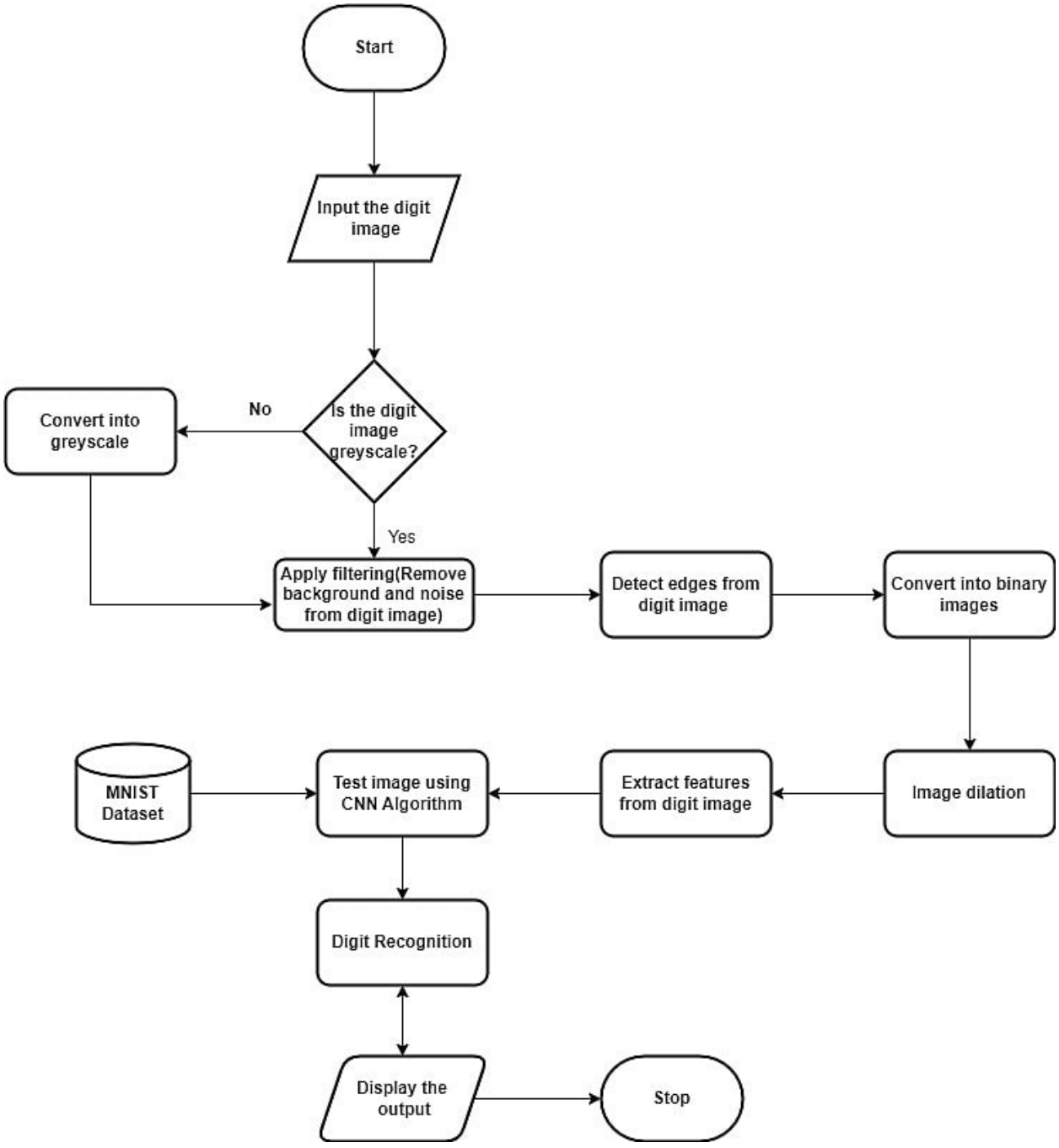
FR No:	FUNCTIONAL REQUIREMENTS & DESCRIPTION
FR-1	Image Data: Handwritten digit recognition is the ability of a computer to recognize the human handwritten digits from different sources like images, papers, touch screens, etc, and classify them into 10 predefined classes (0-9). This has been a topic of boundless-research in the field of deep learning.
FR-2	Website: Web hosting makes the files that comprise a website (code, images, etc.) available for viewing online. Every website you've ever visited is hosted on a server. The amount of space allocated on a server to a website depends on the type of hosting. The main types of hosting are shared, dedicated, VPS and reseller.
FR-3	Digit_Classifier_Model: Use the MNIST database of handwritten digits to train a convolutional network to predict the digit given an image. First obtain the training and validation data.
FR-4	MNIST dataset: The MNIST dataset is an acronym that stands for the Modified National Institute of Standards and Technology dataset.
FR-5	Databases, software, virtual storage, and networking, among others. In layman's terms, Cloud Computing is defined as a virtual platform that allows you to store and access your data over the internet without any limitations.

4.2 Non-Functional Requirements

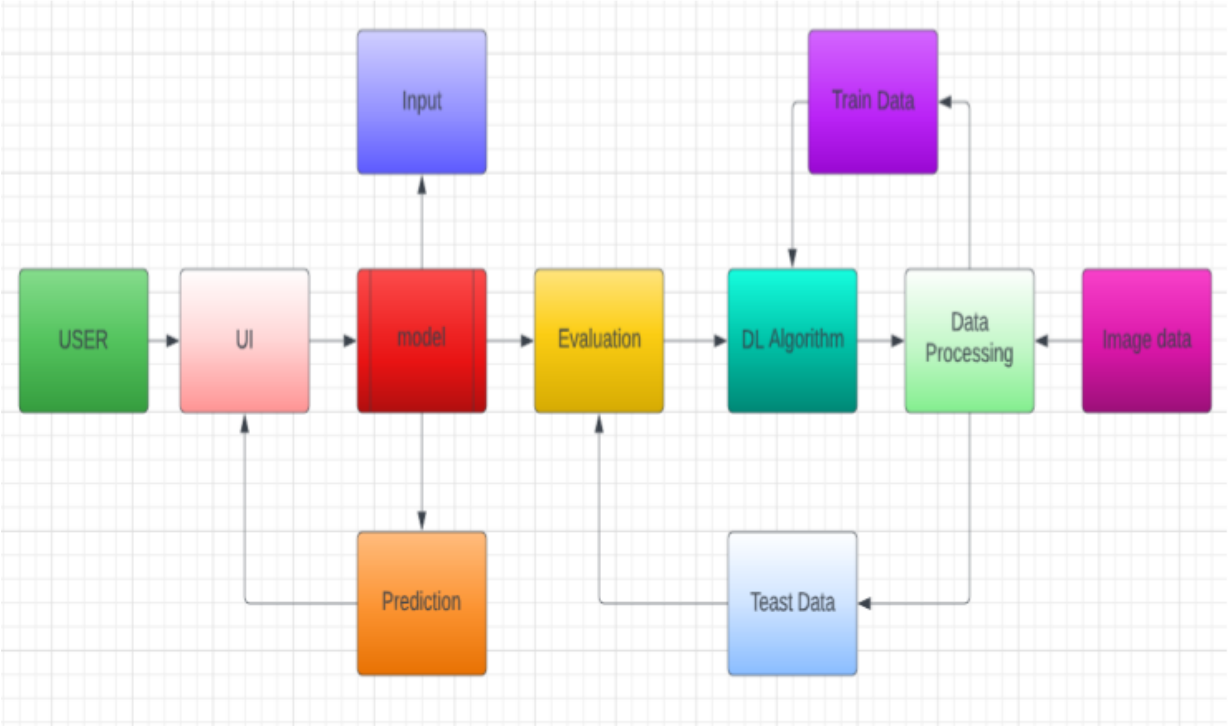
NFR No:	NON FUNCTIONAL REQUIREMENTS & DESCRIPTION
NFR-1	Usability: Handwritten character recognition is one of the practically important issues in pattern recognition applications. The applications of digit recognition include postal mail sorting, bank check processing, form data entry, etc.
NFR-2	Reliability: 1) The system not only produces a classification of the digit but also a rich description of the instantiation parameters which can yield information such as the writing style. 2)The generative models can perform recognition driven segmentation. 3) The method involves a relative.
NFR-3	Performance: The neural network uses the examples to automatically infer rules for recognizing handwritten digits. Furthermore, by increasing the number of training examples, the network can learn more about handwriting, and so improve its accuracy. There are a number of ways and algorithms to recognize handwritten digits, including Deep Learning/CNN, SVM, Gaussian Naive Bayes, KNN, Decision Trees, Random Forests, etc.
NFR-4	Accuracy: Optical Character Recognition (OCR) technology provides higher than 99% accuracy with typed characters in high quality images. However, the diversity in human writing types, spacing differences, and irregularities of handwriting causes less accurate character recognition.

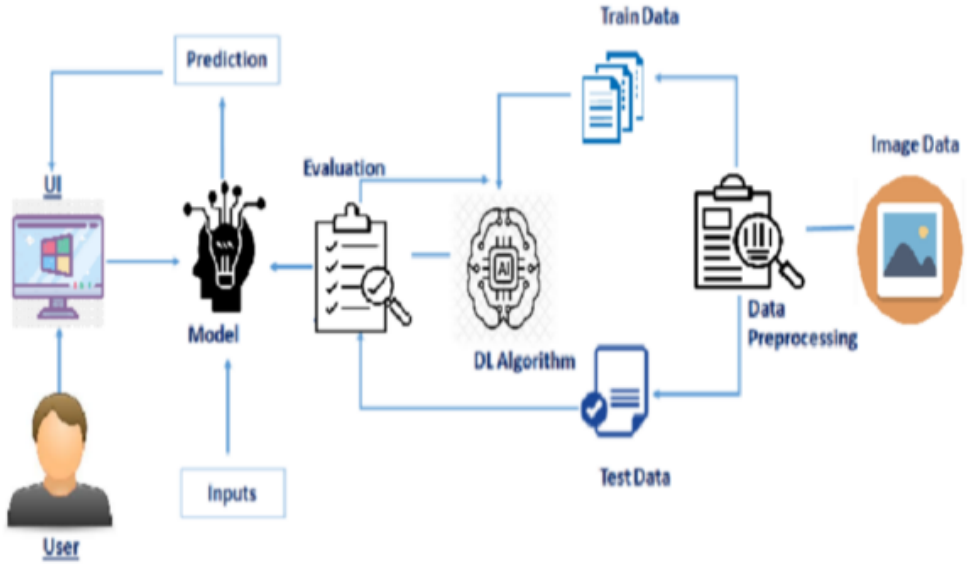
5. PROJECT DESIGN

5.1 Data Flow Diagrams



5.2 Solution & Technical Architecture





Components & Technologies:

S. No	Component	Description	Technology
1.	User Interface	How the user interacts with applications e.g. Web UI, Mobile App, Chatbot etc.	HTML, CSS, JavaScript
2.	Application Logic-1	Logic for a process in the application	Python
3.	Application Logic-2	Logic for a process in the application	IBM Watson STT service
4.	Application Logic-3	Logic for a process in the application	IBM Watson Assistant
5.	Database	Data Type, Configurations etc.	MySQL, NoSQL, etc.
6.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant
7.	File Storage	File storage requirements	IBM Block Storage
8.	External API-1	Purpose of External API used in the application	IBM Weather API
9.	External API-2	Purpose of External API used in the application	Aadhar API
10.	Machine Learning Model	Purpose of Machine Learning Model	Object Recognition Model
11.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration Cloud Server Configuration	Local, Cloud Foundry

Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used	Technology of Open Source framework
2.	Security Implementations	List all the security / access controls implemented, use of firewalls etc.	SHA-256, Encryptions, IAM Controls, OWASP
3.	Scalable Architecture	Justify the scalability of architecture	3 – tier, Micro-services
4.	Availability	Abstract and Figures. The features for handwritten digit recognition have been introduced. These features are based on shape analysis of the digit image and extract slant or slope information. They are effective in obtaining good recognition accuracies	Distributed servers, IBM cloud
5.	Performance	The standard implementations of neural networks achieve an accuracy of ~ (98–99) percent in correctly classifying the handwritten digits	Number of requests per sec, use of Cache, use of CDN’s

5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Home	USN-1	As a user, I can view the guide and awareness to use this application.	I can view the awareness to use this application and its limitations.	Low	Sprint-1
		USN-2	As a user, I'm allowed to view the guided video to use the interface of this application.	I can gain knowledge to use this application by a practical method.	Low	Sprint-1
		USN-3	As a user, I can read the instructions to use this application.	I can read instructions also to use it in a user-friendly method.	Low	Sprint-2

	Recognize	USN-4	As a user, In this prediction page I get to choose the image.	I can choose the image from our local system and predict the output.	High	Sprint-2
	Predict	USN-6	As a user, I'm Allowed to upload and choose the image to be uploaded	I can upload and choose the image from the system storage and also in any virtual storage.	Medium	Sprint-3
		USN-7	As a user, I will train and test the input to get the maximum accuracy of output.	I can able to train and test the application until it gets maximum accuracy of the result.	High	Sprint-4
		USN-8	As a user, I can access the MNIST data set	I can access the MNIST d result.	Medium	Sprint-3
Customer (Web user)	Home	USN-9	As a user, I can view the guide to use the web app.	I can access the MNIST d result.	Low	Sprint-1

	Recognize	USN-10	As a user, I can use the web application virtually anywhere.	I can use the application portably anywhere.	High	Sprint-1
		USN-11	As it is an open source, you can use it freely.	I can use it without any payment to be paid for it to access.	Medium	Sprint-2
		USN-12	As it is a web application, it is installation free	I can use it without the installation of the application or any software.	Medium	Sprint-4
	Predict	USN-13	As a user, I'm Allowed to upload and choose the image to be uploaded	I can upload and choose the image from the system storage and also in any virtual storage.	Medium	Sprint-3

Components & Technologies:

S. No	Component	Description	Technology
1.	User Interface	How the user interacts with application e.g. Web UI, Mobile App, Chatbot etc.	HTML, CSS, JavaScript
2.	Application Logic-1	Logic for a process in the application	Python
3.	Application Logic-2	Logic for a process in the application	IBM Watson STT service
4.	Application Logic-3	Logic for a process in the application	IBM Watson Assistant
5.	Database	Data Type, Configurations etc.	MySQL, NoSQL, etc.
6.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant
7.	File Storage	File storage requirements	IBM Block Storage
8.	External API-1	Purpose of External API used in the application	IBM Weather API
9.	External API-2	Purpose of External API used in the application	Aadhar API
10.	Machine Learning Model	Purpose of Machine Learning Model	Object Recognition Model
11.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration Cloud Server Configuration	Local, Cloud Foundry

Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used	Technology of Open Source framework
2.	Security Implementations	List all the security / access controls implemented, use of firewalls etc.	SHA-256, Encryptions, IAM Controls, OWASP
3.	Scalable Architecture	Justify the scalability of architecture	3 – tier, Micro-services
4.	Availability	Abstract and Figures. The features for handwritten digit recognition have been introduced. These features are based on shape analysis of the digit image and extract slant or slope information. They are effective in obtaining good recognition accuracies	Distributed servers, IBM cloud
5.	Performance	The standard implementations of neural networks achieve an accuracy of ~ (98–99) percent in correctly classifying the handwritten digits	Number of requests per sec, use of Cache, use of CDN’s

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

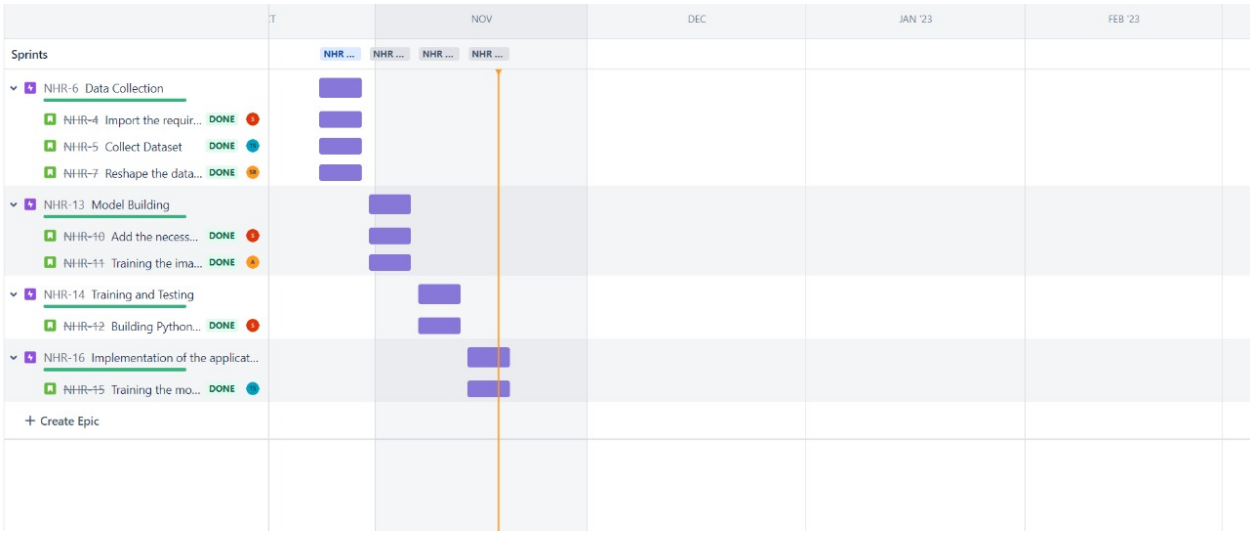
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data Collection	USN-1	Import the required libraries and Collect Dataset .	2	High	Tamil Arasu S, Sachin Siyam P
Sprint-1		USN-2	Reshape the data and apply one hot encoding	1	Medium	Surya R, Ajay M
Sprint-2	Model Building	USN-3	Add the necessary layers and compile the model	2	High	Tamil Arasu S, Ajay M
Sprint-2		USN-4	Training the image classification model using CNN	1	Medium	Surya R, Sachin Siyam P
Sprint-3	Training and Testing	USN-5	Building Python code and run the application	2	High	Tamil Arasu S, Surya R, Sachin Siyam P, Ajay M

Sprint-4	Implementation of the application and deployment on cloud	USN-6	Training the model on IBM cloud.	2	High	Tamil Arasu S, Surya R, Sachin Siyam P, Ajay M

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 Nov2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

6.3 Reports from JIRA



7. CODING AND SOLUTIONING

7.1 Feature 1

- ➢ Multi User Authentication
- ➢ Access through Internet
- ➢ Reduced Time and Computational complexity
- ➢ Implement in real time environments

7.2 Feature 2

- ➢ Create more stable model
- ➢ More accuracy than average Humans

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	The Home page must be displayed properly in all sizes	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	FAIL

					and 768 x 630	
HP_TC_003	Functional	Home Page	Check if user can upload their file	The input image should be uploaded to the application successfully	Working as expected	PASS
HP_TC_004	Functional	Home Page	Check if user cannot upload unsupported files	The application should not allow user to select a non image file	Working as expected	PASS
HP_TC_005	Functional	Home Page	Check if the pages redirects to the result page once the	The page should redirect to the results page	Working as expected	PASS

			input is given			
BE_TC_001	Functional	Backend	Check if all the routes are working properly	Check if all the routes are working properly	Working as expected	PASS
M_TC_001	Functional	Model	Check if the model can handle various image sizes	Check if the model can handle various image sizes	Working as expected	PASS
M_TC_002	Functional	Model	Check if the model predicts the digit	Check if the model predicts the digit	Working as expected	PASS
M_TC_003	Functional	Model	Check if the model can handle complex input image	Check if the model can handle complex input image	The model fails to identify the digit since the	FAIL

					model is not built to handle such data	
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 result is displayed properly	The result should be displayed properly	Working as expected	PASS

8.2 User Acceptance Testing

8.2.1 Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved.

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

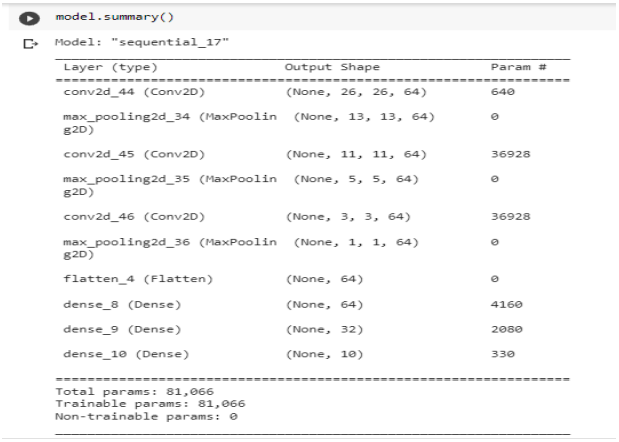
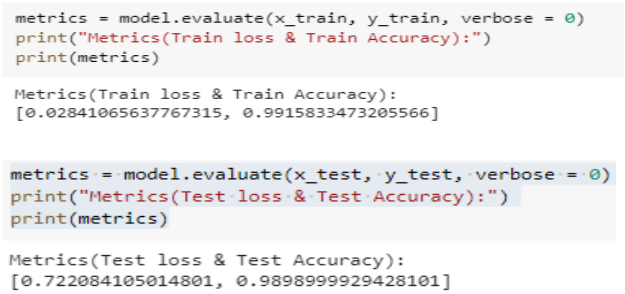
8.2.2 Test Case Analysis

This report shows the number of test cases that have passed, failed, and untested.

Section	Total Cases	Not Tested	Fail	Pass
Client Application	11	0	2	9
Security	2	0	1	1
Performance	3	0	0	3
Exception Reporting	2	0	0	2

9. RESULTS

9.1 Performance Metrics

S.No	Parameter	Values	Screenshots
1.	Model Summary	Model: ”sequential”	
2.	Accuracy	Training Accuracy - 0.99 Validation Accuracy -0.98	

```
model.summary()
```

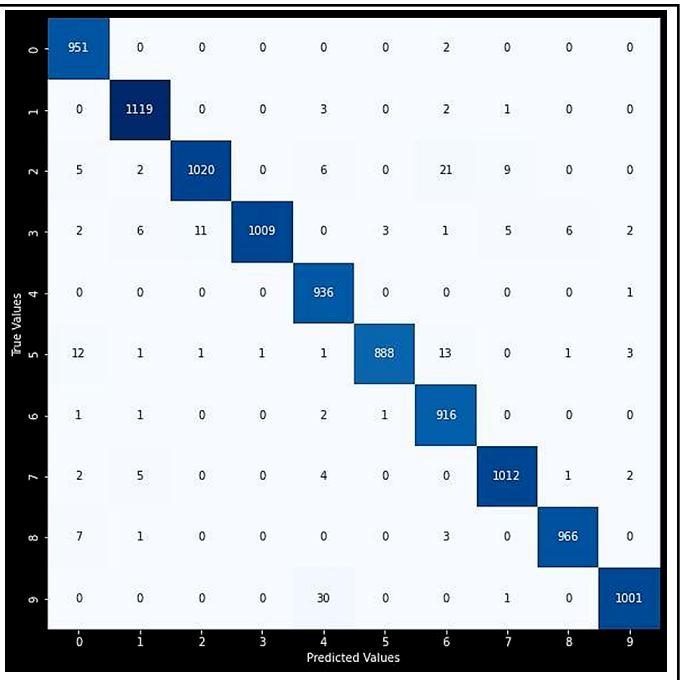
```
Model: "sequential_17"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_44 (Conv2D)          (None, 26, 26, 64)        640

 max_pooling2d_34 (MaxPoolin  (None, 13, 13, 64)       0
 g2D)

 conv2d_45 (Conv2D)          (None, 11, 11, 64)        36928

 max_pooling2d_35 (MaxPoolin  (None, 5, 5, 64)         0
 g2D)

 conv2d_46 (Conv2D)          (None, 3, 3, 64)          36928

 max_pooling2d_36 (MaxPoolin  (None, 1, 1, 64)         0
 g2D)

 flatten_4 (Flatten)         (None, 64)                0

 dense_8 (Dense)             (None, 64)                4160

 dense_9 (Dense)             (None, 32)                2080

 dense_10 (Dense)            (None, 10)                330

=================================================================
Total params: 81,066
Trainable params: 81,066
Non-trainable params: 0
_________________________________________________________________
```

```
metrics = model.evaluate(x_train, y_train, verbose = 0)
print("Metrics(Train loss & Train Accuracy):")
print(metrics)
```

```
Metrics(Train loss & Train Accuracy):
[0.02841065637767315, 0.9915833473205566]
```

```
metrics = model.evaluate(x_test, y_test, verbose = 0)
print("Metrics(Test loss & Test Accuracy):")
print(metrics)
```

```
Metrics(Test loss & Test Accuracy):
[0.722084105014801, 0.9898999929428101]
```

3	Metrics	Confusion Matrix	
4.	Metrics	Classification Model	

	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

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

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

A recognition system for handwritten digit has attracted some interests in the research community by introduction of large dataset. In this work, convolutional neural networks were applied for handwritten digit recognition. The goal was the recognition of patterns taken from the MNIST database. CNNs were modified by the use of contour features, which are known as good feature extractors. The results demonstrated that CNNs perform pattern recognition effectively, incorporating in its structure some feature extraction and feature mapping characteristics, which are extremely adapted to invariances usually found in pattern recognition problems. Likewise, it was shown that Contour features can be appropriately incorporated in a CNN architecture, because such methods own similar principles. The effectiveness of the soft max method in classification improvement is also confirmed.

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 digits from manual writing in canvas.
- ➢ 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

CODING

MODEL

```
import numpy as np
import tensorflow
from tensorflow.keras.datasets import mnist
from tensorflow.keras.models import Sequential
from tensorflow.keras import layers
from tensorflow.keras.layers import Dense, Flatten, MaxPooling2D, Dropout

from tensorflow.keras.layers import Conv2D
from keras.optimizers import Adam
from keras.utils import np_utils
import matplotlib.pyplot as plt

(x_train, y_train), (x_test, y_test) = mnist.load_data()

x_train = x_train.reshape(60000, 28, 28, 1).astype('float32')
x_test = x_test.reshape(10000, 28, 28, 1).astype('float32')

model= Sequential()

model.add(Conv2D(64, (3,3),input_shape=(28, 28, 1), activation="relu"))
model.add(MaxPooling2D(pool_size=(2,2)))

model.add(Conv2D(64, (3,3), activation="relu"))
model.add(MaxPooling2D(pool_size=(2,2)))

model.add(Conv2D(64, (3,3), activation="relu"))
model.add(MaxPooling2D(pool_size=(2,2)))
```

```
model.add (Flatten())

model.add (Dense(64, activation="relu"))

model.add (Dense(32, activation="relu"))

model.add(Dense(10, activation="softmax"))

model.summary()

model.compile(loss = 'sparse_categorical_crossentropy', optimizer = "Adam",
metrics = ['accuracy'])

test_loss, test_acc = model.evaluate(x_test, y_test)

predictions = model.predict([x_test])

print(np.argmax(predictions[0]))

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

APP.PY

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

```
from flask import send_from_directory

UPLOAD_FOLDER = 'C:/Users/ELCOT/Desktop/IBM/Final
Deliverables/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=True, threaded=False)
```

INDEX.HTML

```
<!DOCTYPE html>
<html>
    <head>
        <title>Handwritten Digits Recognition System</title>
        <link rel="stylesheet" href="{{url_for('static',filename='css/index.css')}}">
        <script type="text/javascript" src
="{{url_for('static',filename='js/web.js')}}"></script>
        <!-- <link rel="stylesheet" href="C:\Users\ELCOT\Desktop\IBM\Final
Deliverables\static\css\index.css">
        <script type="text/javascript" src = "C:\Users\ELCOT\Desktop\IBM\Final
Deliverables\static\js\web.js"></script> -->


    </head>
    <body>
        <div id="menu" class="menu"><!--This is menu bar-->
            <li><a href="#d5">Recognize</a></li>
            <li><a href="/">Home</a></li>
        </div>
        <div id="ddd"></div>
        <div id="d1">
            <h1 id="hand">Handwritten Digit Recognition System</h1>
```

<h2>TEAM ID : PNT2022TMID47342</h2>

<section id="s1">

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

<h1>HANDWRITTEN DIGIT RECOGNITION</h1>
<img src="static/images/hand.png">
 Handwritten digit recognition can be performed using the Convolutional neural network from Machine Learning. Using the MNIST (Modified National Institute of Standards and Technologies) database and compiling with the CNN gives the basic structure of my project development. So, basically to perform the model we need some libraries such as NumPy, `Pandas', TensorFlow, Keras. These are the main structure on which my main project stands. MNIST data contains about 70,000 images of handwritten digits from 0-9. So, it is a class 10 classification model. This dataset is divided into 2 parts i.e. Training and

Test dataset. Image representation as 28*28 matrix where each cell contains grayscale pixel value.

<h1>CNN</h1>

CNN is a deep learning technique to classify the input automatically (well, after you provide the right data). Over the years, CNN has found a good grip over classifying images for computer visions and now it is being used in healthcare domains too. This indicates that CNN is a reliable deep learning algorithm for an automated end-to-end prediction. CNN essentially extracts ‘useful’ features from the given input automatically making it super easy for us!

<h1>MNIST</h1>

The dataset that is being used here is the MNIST digits classification dataset. Keras is a deep learning API written in Python and MNIST is a dataset provided by this API. This dataset consists of 60,000 training images and 10,000 testing images. It is a decent dataset for individuals who need to have a go at pattern recognition as we will perform in just a minute! When the Keras API is called, there are four values returned namely- x_train, y_train, x_test, and y_test.

</section>

<section id="s2">

<h1>RECOGNITION</h1>

<P>Upload the image and click PREDICT button to Recognize</P>

<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>
            <div id="d62"><img id="frame" src="" width="200px"
height="200px"/></div>
            <div id="d61">
               <button type="submit"  id="predict">Predict</button>
               <button type="reset"  id="clear" onclick="clear()">Clear</button>
            </div>
         </form>
      </div>
   </section>
   <section id="s3">
      <h1>TEAM MEMBERS</h1>
      <div id="d7">
         <h2> Tamil Arasu S       
         -     
       830119104049</h2>
         <h2> Surya
R            
           
 -            
830119104048</h2>
         <h2> Sachin Siyam P     
        -     
        830119104036</h2>
         <h2> Ajay M          
              
```

```
  -            
830119104002</h2>
            </div>
        </section>
    </body>
</html>
```

PREDICT.HTML

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

<style>
    body{
        background-color: rgb(250, 242, 232);
    }

    #rectangle{
        width:600px;
        height:250px;
        background-color:rgb(90, 35, 35);
        border-radius: 55px;
        position:absolute;
        top:55%;
        left:50%;
        transform:translate(-50%,-50%);
```

```
    box-shadow: 17px 17px 15px  rgb(97, 47, 47);
}

#ans{
    text-align: center;
    font-size: 50px;
    margin: 0 auto;
    padding: 3% 5%;
    padding-top: 15%;
    color: white;
}
#d1{
    margin-left: auto;
    margin-right:auto;
    width: 900px;
    text-align: center;
    margin-top: 60px;
}

#d1 h1{
    color: rgb(247, 202, 57);
    font-size: 50px;
    text-shadow: 2px 2px  rgb(90, 35, 35);
}

#d1 h2{
    color: rgb(90, 35, 35);
    margin-top: -10px;
}
```

```
</style>
<body>
   <div id="d1">
      <h1 id="hand">Handwritten Digit Recognition System</h1>
      <h2>TEAM ID : PNT2022TMID47342</h2>
   </div>
   <div id="rectangle">
      <h1 id="ans">Predicted Number : {{num}}</h1>
   </div>
</body>
</html>
```

INDEX.CSS

```
li a {
   display: block;
   color: rgb(255, 255, 255);
   text-align: center;
   padding: 14px 20px;
   text-decoration: none;
   font-size: 20px;
 }


 li a:hover {
   background-color: rgb(247, 202, 57);
   color: rgb(90, 35, 35);
 }
 #menu {
   list-style-type: none;
```

```
    margin: 0;
    padding: 0;
    overflow: hidden;
    background-color: rgb(90, 35, 35);
    border-radius: 15px;
    position:fixed;
    width: 1330px;
    margin-top: -50px;
  }
  li {
    float: right;
  }

  /*body{
    background-image: url(home_bg.jpg);
    background-repeat: no-repeat;
    background-size: cover;
    background-attachment: fixed;
  }*/
body{
  background-color: rgb(250, 242, 232);
}
  #d1{
    margin-left: auto;
    margin-right:auto;
    width: 900px;
    text-align: center;
    margin-top: 60px;
  }
```

```
#d1 h1{
  color: rgb(247, 202, 57);
  font-size: 50px;
  text-shadow: 2px 2px  rgb(90, 35, 35);
}

#d1 h2{
  color: rgb(90, 35, 35);
  margin-top: -10px;
}

#d2{
  margin-left: auto;
  margin-right:auto;
  width: 1200px;
  text-align: justify;
  font-size: 20px;
}
#d3{
  margin-left: auto;
  margin-right:auto;
  width: 1200px;
  text-align: justify;
  font-size: 20px;
}
#d31{
   margin-left: 900px;
   color: rgb(247, 202, 57);
   text-decoration: overline underline;
```

```
   text-decoration-color:  rgb(90, 35, 35);
}
#d32{
  width: 850px;
  margin-top: -200px;
}
#d3 h1{
  text-align: center;
  color:  rgb(90, 35, 35);
  text-decoration-line: underline;
  text-decoration-style: double;
   text-decoration-color: rgb(247, 202, 57);
}
#d4{
  margin-left: auto;
  margin-right:auto;
  width: 1200px;
  text-align: justify;
  font-size: 20px;
}
#d4 h1{
  text-align: center;
  color:  rgb(90, 35, 35);
  text-decoration-line: underline;
  text-decoration-style: double;
   text-decoration-color: rgb(247, 202, 57);
}
#d5{
  margin-left: auto;
```

```
  margin-right:auto;
  width: 1200px;
  text-align: justify;
  font-size: 20px;
}
#d5 h1{
  text-align: center;
  color:  rgb(90, 35, 35);
  text-decoration-line: underline;
  text-decoration-style: double;
   text-decoration-color: rgb(247, 202, 57);
}

#d6{
  margin-left: auto;
  margin-right: auto;
  width: 1200px;
  font-size: 20px;
  text-align: center;
}
#d6 h1{
  text-align: center;
  color:  rgb(90, 35, 35);
  text-decoration-line: underline;
  text-decoration-style: double;
   text-decoration-color: rgb(247, 202, 57);
}
#predict{
  height: 35px;
```

```
  width: 150px;
  background-color:  rgb(90, 35, 35);
  color: white;
  border: none;
  border-radius: 15px;
  box-shadow: 7px 7px 5px  rgb(97, 47, 47);
  font-size: 20px;

  /* margin-left: 100px; */
}

#clear{
  height: 35px;
  width: 150px;
  background-color:  rgb(90, 35, 35);
  color: white;
  border: none;
  border-radius: 15px;
  box-shadow: 7px 7px 5px  rgb(97, 47, 47);
  font-size: 20px;
  margin-left: 100px;

}
#predict:hover {
  background-color: rgb(247, 202, 57);
  color: rgb(90, 35, 35);
}
#clear:hover {
  background-color: rgb(247, 202, 57);
```

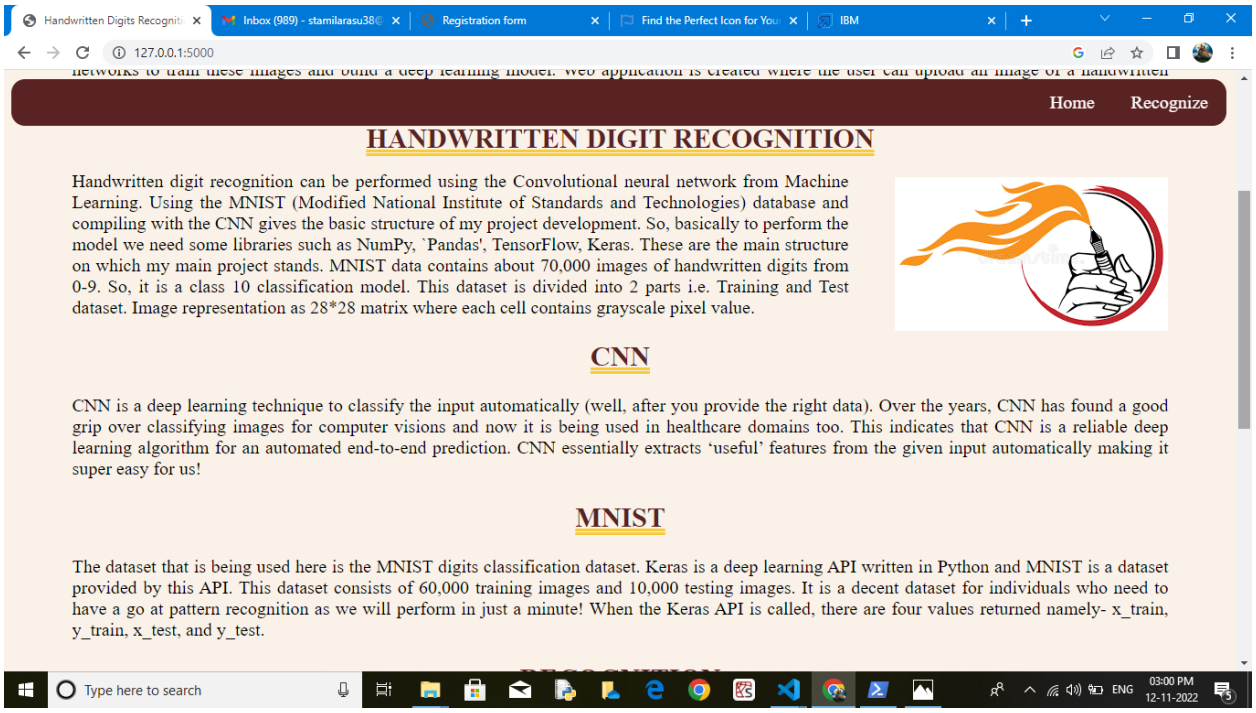
```
  color: rgb(90, 35, 35);
}
#frame{
  display: none;
  margin-left: auto;
  margin-right: auto;
}

#d61{
  margin-top: 50px;

}

#d62{
  display: block;
}
#s3{
  background-color:  rgb(90, 35, 35);
  color: white;
  padding: 20px;
  margin-top: 20px;


}
#s3 h1{
  width: 300px;
  margin-left: auto;
  margin-right:auto ;
  text-align: center;
  color: rgb(247, 202, 57);
  font-size: 30px;
```

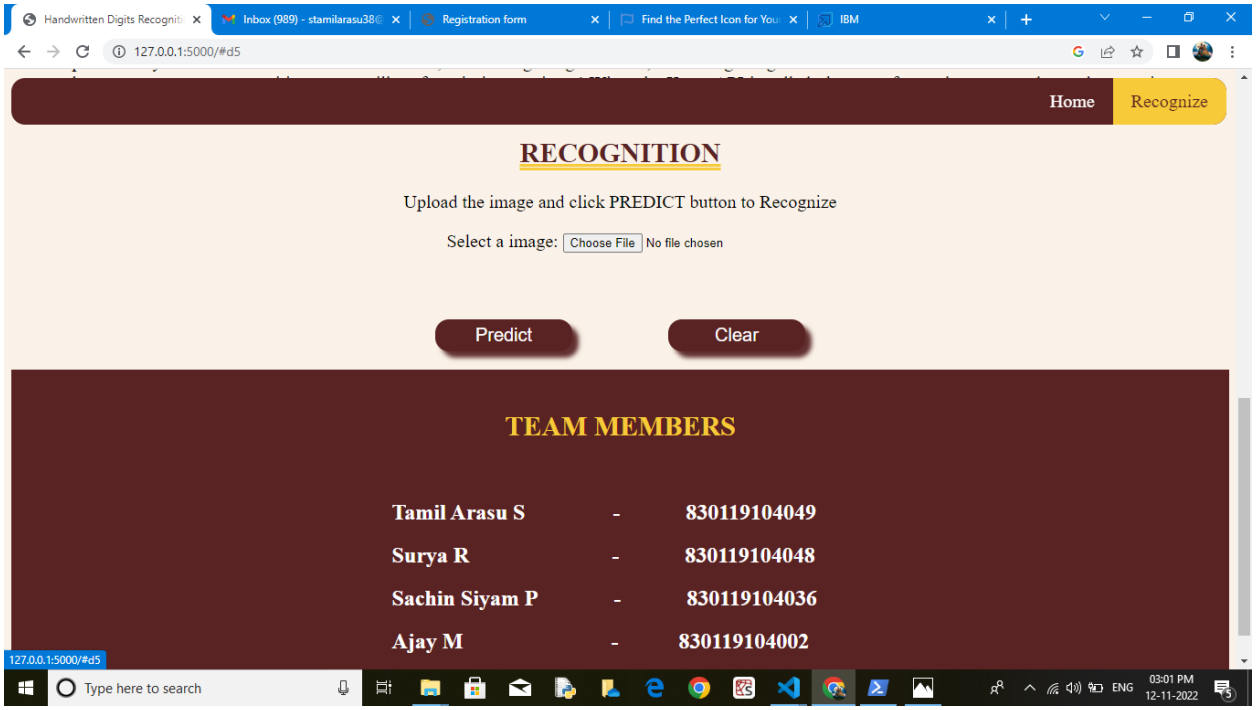
```
}
#d7{
  width: 500px;
  margin-left: auto;
  margin-right: auto;
  padding: 20px;
}
```

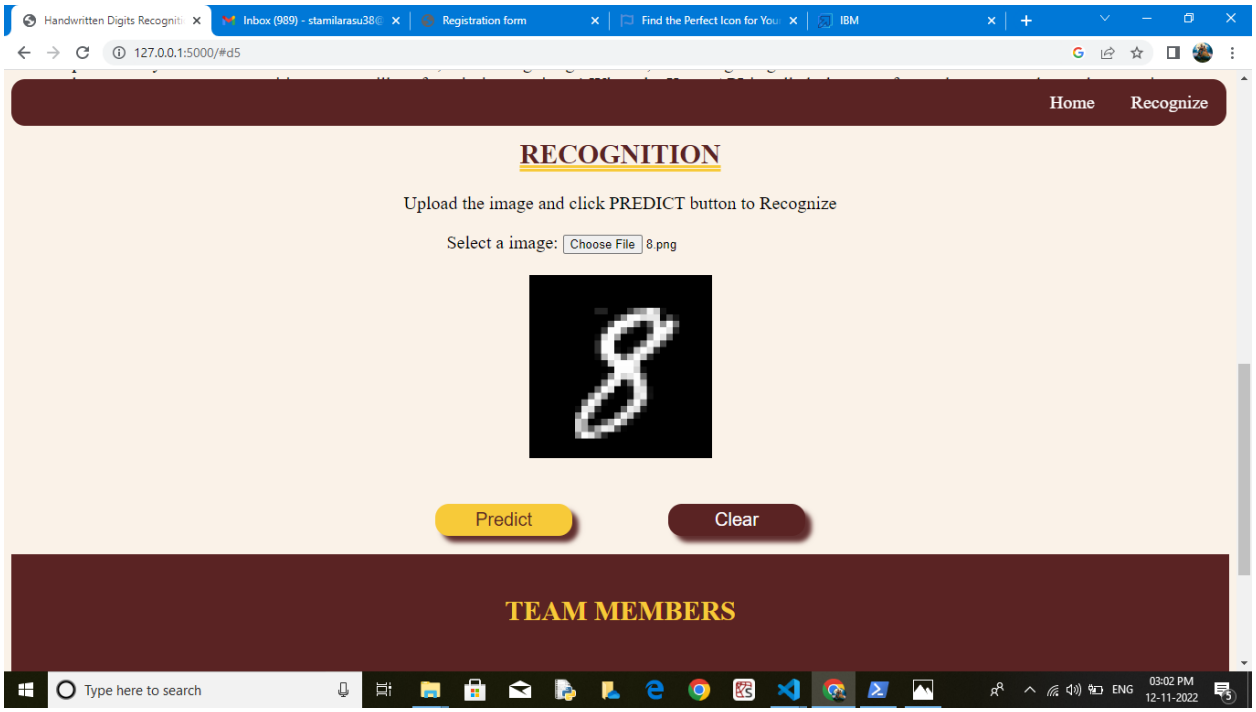
WEB.JS

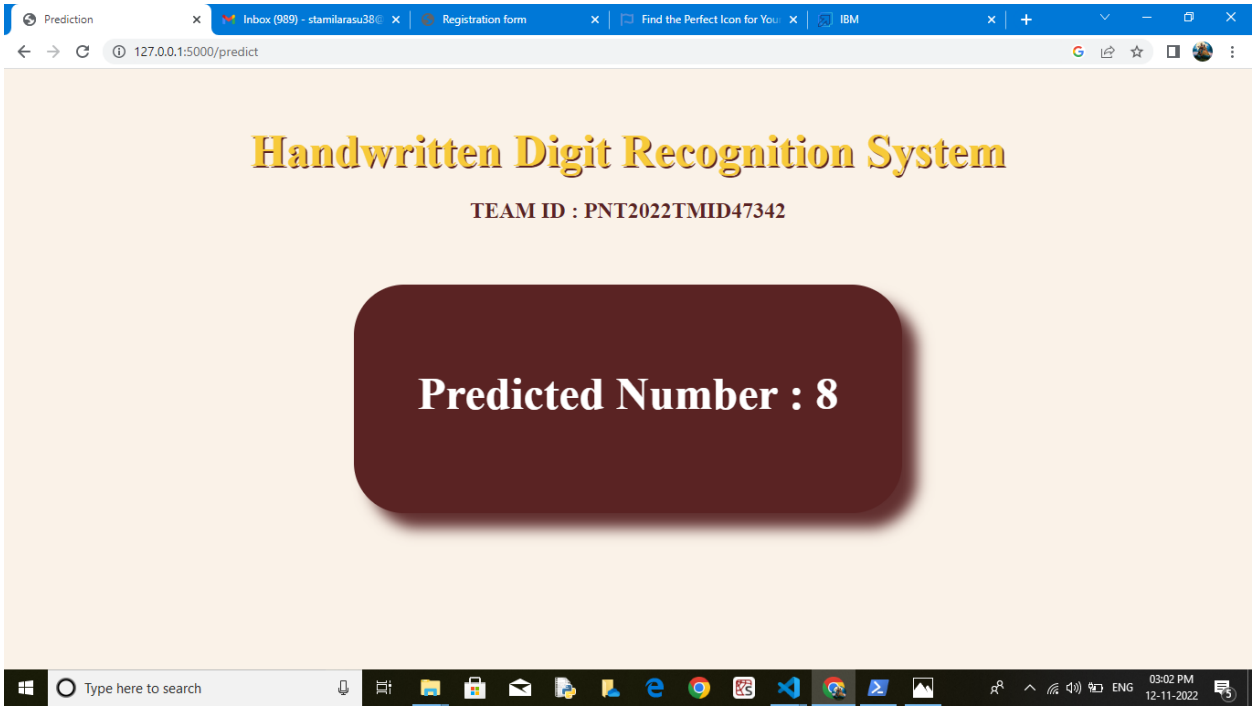
```
function preview() {
    frame.src=URL.createObjectURL(event.target.files[0]);
    document.getElementById("frame").style.display = "block";
}
$(document).ready(function() {
    $('#clear').on('click', function() {
        $('#image').val('');
        $('#frame').attr('src',"");
        });
});
function clear(){
    document.getElementById("d62").style.display="none";
}
```

SCREENSHOTS




Handwritten Digits Recogniti
Inbox (989) - stamilarasu38
Registration form
Find the Perfect Icon for You
IBM
127.0.0.1:5000/#d5
Home
Recognize
RECOGNITION
Upload the image and click PREDICT button to Recognize
Select a image: Choose File No file chosen
Predict
Clear
TEAM MEMBERS
Tamil Arasu S - 830119104049
Surya R - 830119104048
Sachin Siyam P - 830119104036
Ajay M - 830119104002
127.0.0.1:5000/#d5
Type here to search
ENG
03:01 PM
12-11-2022


Handwritten Digits Recogniti
Inbox (989) - stamilarasu38
Registration form
Find the Perfect Icon for You
IBM
127.0.0.1:5000/#d5
Home
Recognize
RECOGNITION
Upload the image and click PREDICT button to Recognize
Select a image: Choose File 8.png
Predict
Clear
TEAM MEMBERS
Type here to search
ENG
03:02 PM
12-11-2022



GITHUB LINK:

➤ https://github.com/IBM-EPBL/IBM-Project-37818-1660326535

DEMO VIDEO LINK:

➤ https://drive.google.com/folderview?id=10Bw7f9FETyZ2h0XaZYjhZYkNF0OUGoEB