PROJECT REPORT

A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

submitted by

PNT2022TMID27693

SNEHA H - 311419104076

SOWMIYA B- 311419104078

SUBAJA E - 311419104081

SUBASHREE M - 311419104082

TABLE OF CONTENTS

A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

CHAPTER 1

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.

Here comes the use of Deep Learning. In the past decade, deep learning has become the hot tool for Image Processing, object detection, handwritten digit and character recognition etc. A lot of machine learning tools have been developed like scikit-learn, SciPy-image etc. and pyrans, Kera's, Theano, TensorFlow by Google, Learn etc. for Deep Learning. These tools make the applications robust and therefore more accurate. The Artificial Neural Networks can almost mimic the human brain and are a key ingredient in image processing field. For example, Convolutional Neural Networks with Back Propagation for Image Processing, Deep Mind by Google for creating Art by learning from existing artist styles etc..

Handwriting Recognition has an active community of academics studying it. The biggest conferences for handwriting recognition are the International Conference on Frontiers in Handwriting Recognition (ICFHR), held in even-numbered years, and the International Conference on Document Analysis and Recognition (ICDAR), held in odd-numbered years. Both of these conferences are endorsed by the IEEE. Active areas of research include: Online Recognition, Offline Recognition, Signature Verification, Postal-Address Interpretation, Bank-Check Processing, Writer Recognition

1.2 Purpose

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 consist of 60,000 training images and 10,000 test images.The artificial neural neworks 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 these 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.

CHAPTER 2

2.LITERATURE SURVEY

2.1 References

1.Richa Goswami and O.P Sharma

A review on character recognition techniques

International Journal of Computer Application (0975-8887)

Volume 83-NO 7, December 2013:

In this study, main approaches used in the CR field during the last decade are overviewed. Different pre-processing segmentation techniques and various classifiers with different features are also discussed. It is found that neither the structural nor the statistical and structural information can represent a complex pattern alone. Therefore, one needs to combine statistical and structural information supported by semantic information.

2.Kimura.F and Shiridhar.M (1991)

Handwritten numerical recognition based on multiple algorithms, pattern Recognition,

No 10, vol.24, pp 969-983

In this study the main approach is to identify numerals that may be isolated or connected, broken or continuous. Here, the recognition algorithm is derived as a tree classifier. In an extensive test experiment, an accuracy of 99% was realized with isolated numerals when connected numerals were also included a recognition accuracy of 93% was obtained.

3.Ayush purohit and Shardul Singh Chauhan

A literature survey on handwritten character recognition,

International journal of computer science and information technologies,

Volume.7 (1), 2016, 1-5

This paper discusses in detail all advances in the area of handwritten character recognition. From the study, it is analyzed that the selection of the classification as well as the feature extraction techniques needs to be proper in order to attain good rate in recognizing the character. Studies in the paper reveals that there is still scope of enhancing the algorithms as well as enhancing the rate of recognition of characters.

2.2 Problem statement definition

A well-articulated customer problem statement allows our team to find the ideal solution for the challenges your customers face. Throughout the process, we will also be able to empathize with our customers, which helps us better understand how they perceive our product or service.

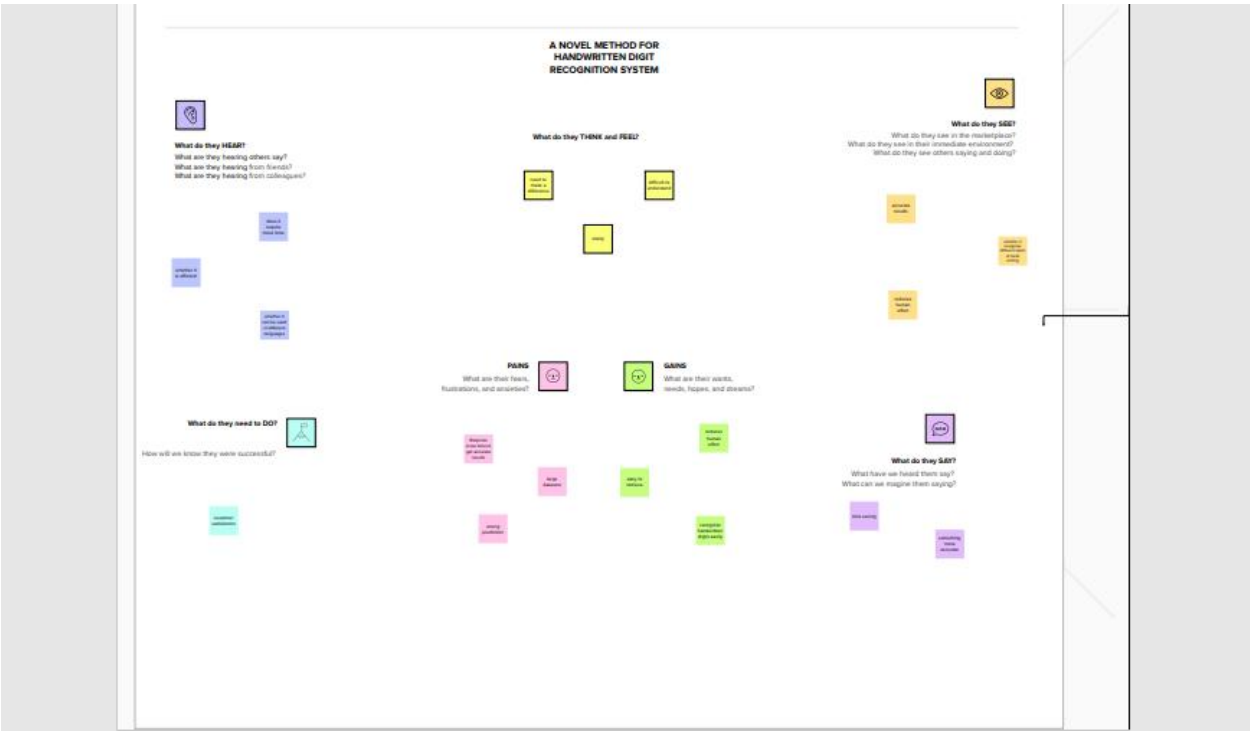
Problem statement(PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	Creator	To Train and test the model	There will be different styles of writing	Hence we have to collect dataset of different people	Dissatisfaction
PS-2	User	To recognize the digit exactly	But the prediction is inaccurate	No optical character recognition	Disappointment

CHAPTER 3

3.IDEATION & PROPOSED SOLUTION

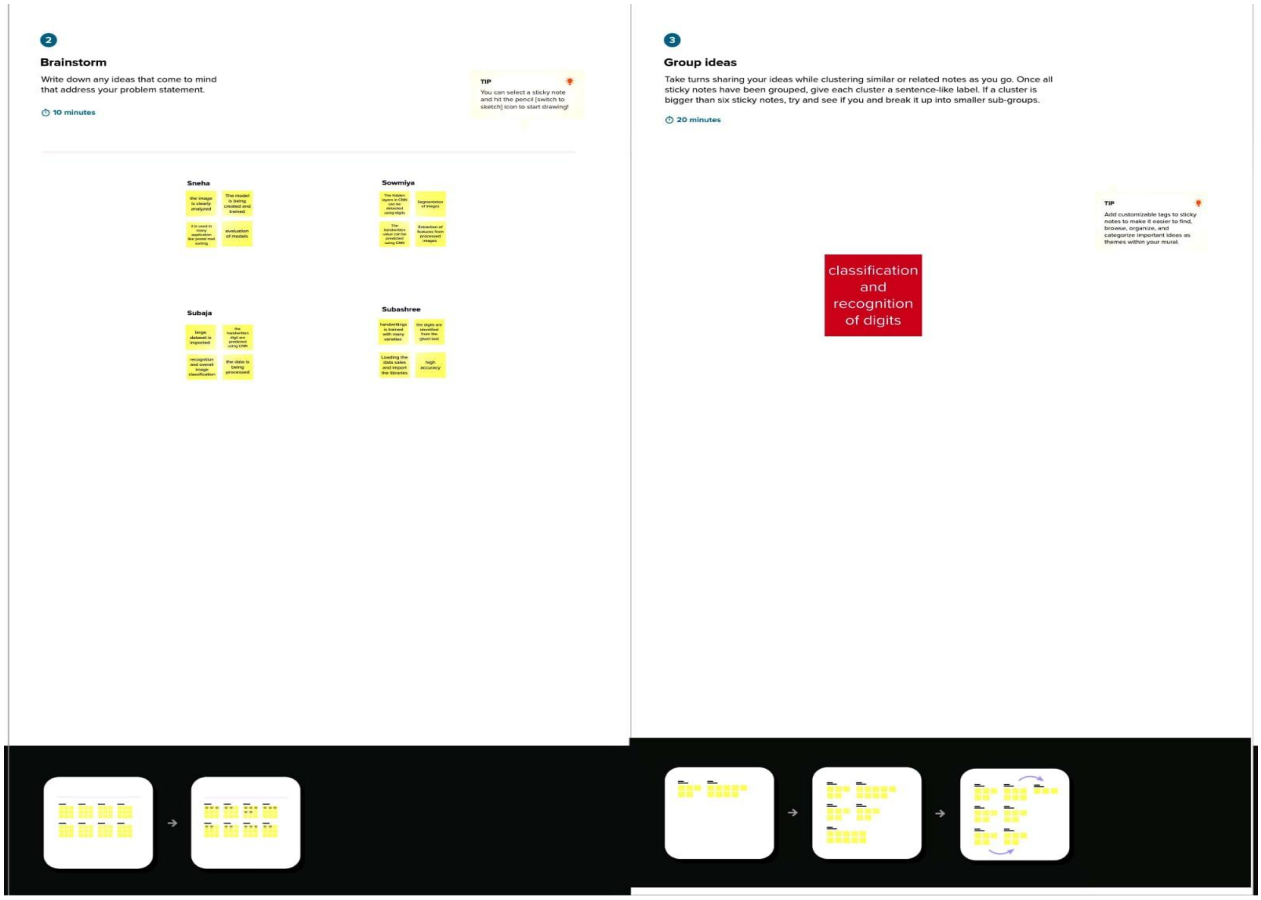
3.1 Empathy map canvas

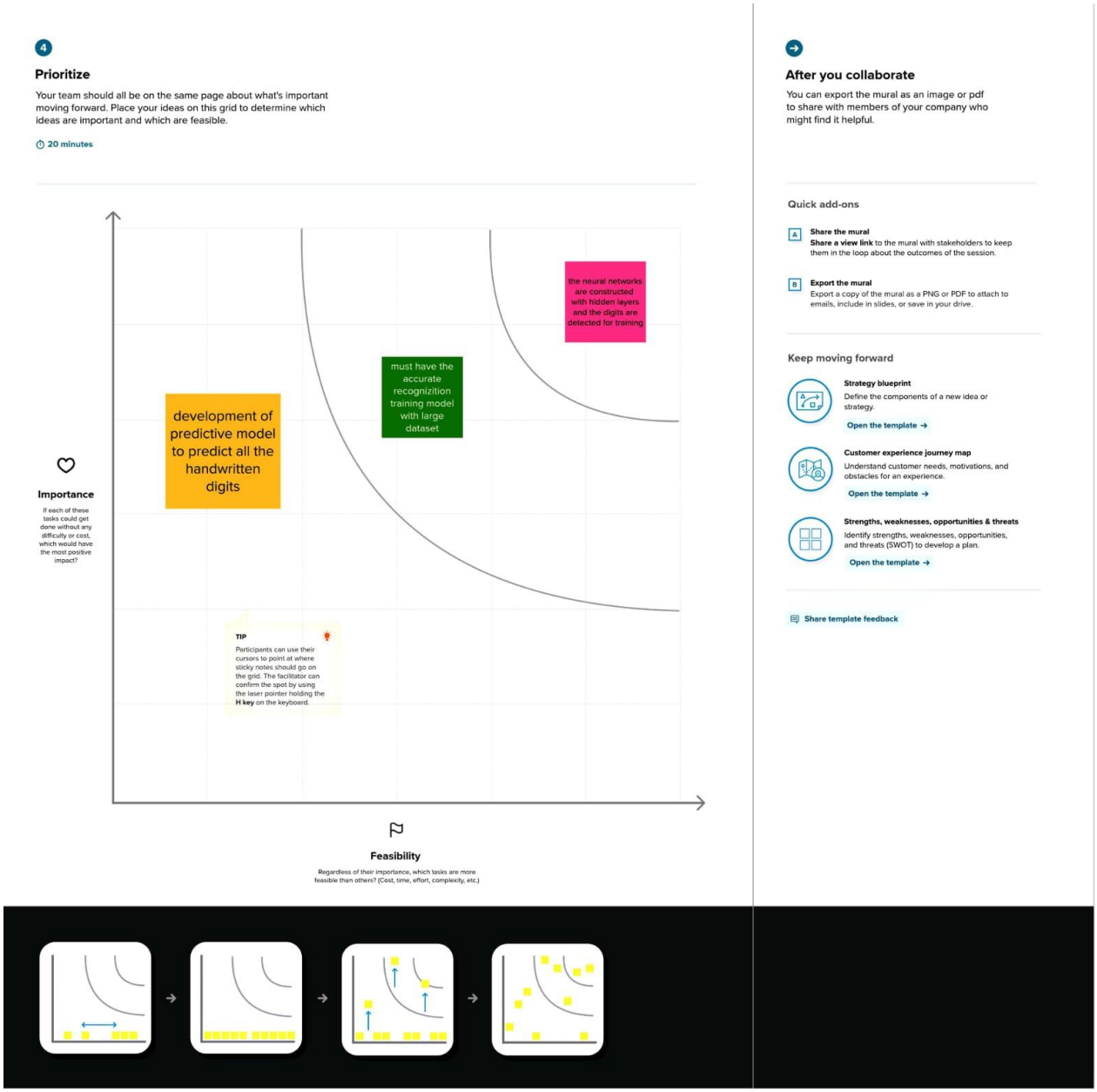
An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviors and attitudes. It is a useful tool to helps teams better understand their users. Creating an effective solution requires understanding the true problem and the person who is experiencing the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with the goals and challenges.



3.2 Ideation & Brainstorming

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.


2
Brainstorm
Write down any ideas that come to mind that address your problem statement.
10 minutes
TIP
You can select a sticky note and hit the pencil (switch to sketch) icon to start drawing!
Sneha
Sowmiya
Subaja
Subashree
3
Group ideas
Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.
20 minutes
TIP
Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.
classification and recognition of digits



3.3 proposed solution

S.No	Parameter	Description
1.	Problem Statement (Problem to be solved)	A Novel Method for Handwritten Digit

		Recognition System
2.	Solution description	With the help of CNN based model we classify digits which is in handwritten format and this model can be trained by using MNIST database which contains 60,000training samples and 10,000 test samples. By doing so we propose solution to the given Problem.
3.	Novelty	Using CNN we classify the image datasets, compared to other methods CNN provides an efficient solution. Here use of ANN algorithm is advisable for voice recognition which helps blind people.
4.	Customer Satisfaction	Use of external dependencies or devices to recognize the digits is not required here. Users can use their mobile phones for this process.

5.	Business Model (Revenue Model)	• Input module • Image processing module • Feature extraction module • Data set training module • Analysis module
6.	Scalability of the Solution	An accuracy of 99.98%, and 99.40% is obtained for the result of the training dataset with 50% noise by using MNIST. We can even improve this model to achieve the better results by training different types of datasets

3.4 Problem solution fit

CHAPTER 4

4. REQUIREMENT ANALYSIS

4.1 Functional requirement

Following are the functional requirements of the proposed solution

FR NO	Functional Requirements (EPIC)	Sub Requirements (Sub Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through Mobile Number Registration through LinkedIn
FR-2	User Confirmation	Confirmation via OTP Confirmation via Email
FR-3	Datasets	MNIST Datasets can be Imported

4.2 Non-functional Requirements:

Following are the non-functional requirements of the proposed solution

FR NO	Non-Functional Requirements	Description
NFR-1	Usability	It has several application which include identification of the Digit Recognition like Number plate recognition, Postal identification, etc.
NFR-2	Security	The uploaded images are not stored so it acts as a secure data stream.
NFR-3	Reliability	It is reliable and accurate of data after training is possible here.
NFR-4	Performance	Each training of datasets results in the improved accuracy thus resulting in easy UI.

NFR-5	Availability	Usage at anytime and anywhere is available.
NFR-6	Scalability	Improve in the efficiency and accommodation with expansion of the Digit Recognition application is achieved with the help of web application.

CHAPTER 5

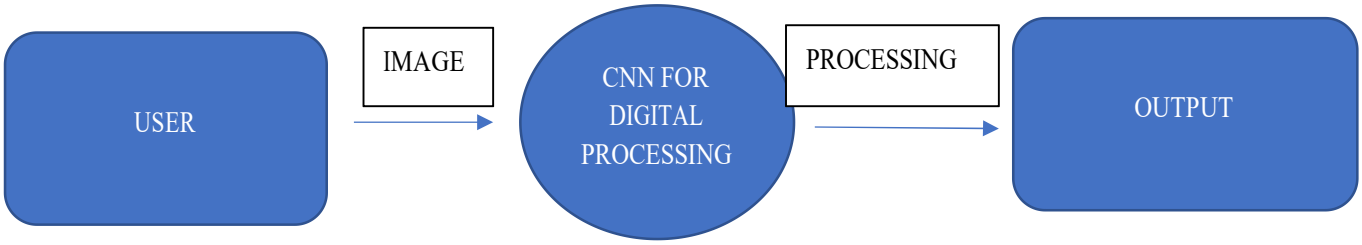
5 PROJECT DESIGN

5.1 Data flow diagrams

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

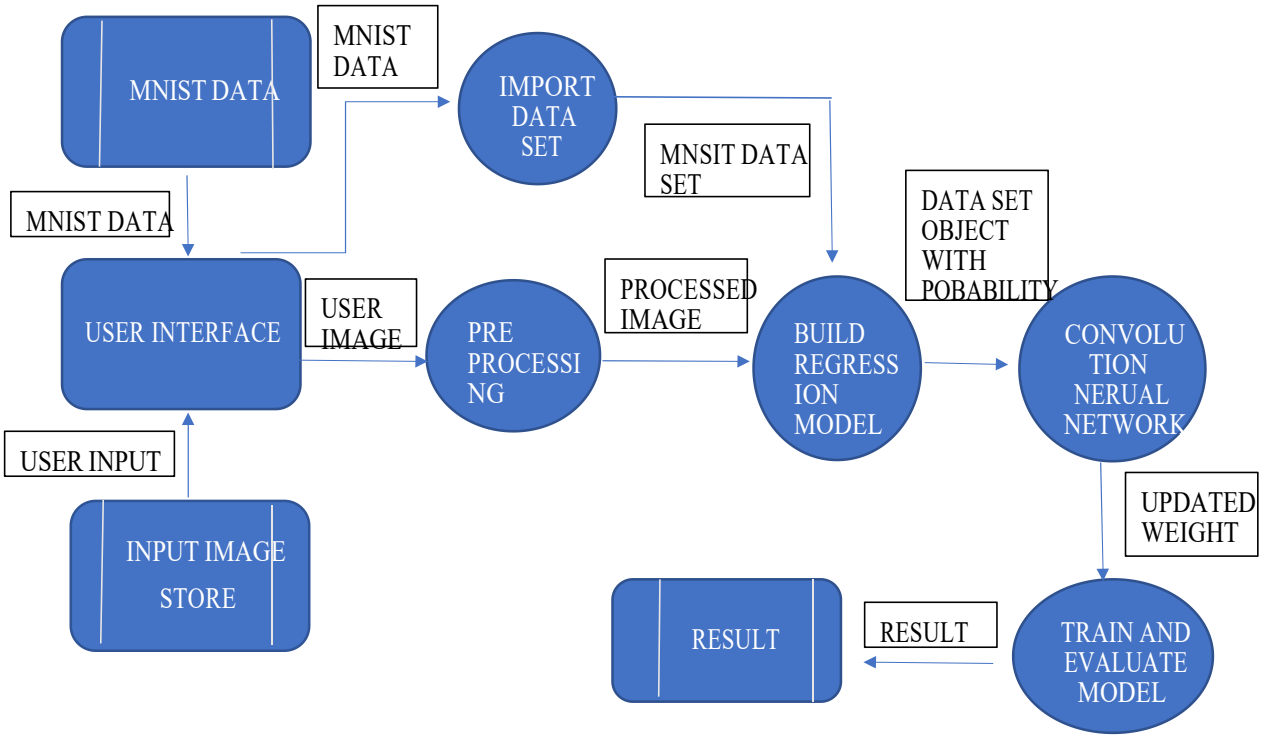
DFD Level-0

The DFD Level-0 consists of two external entities, the UI and the Output, along with a process, representing the CNN for Digit Recognition .Output is obtained after processing.



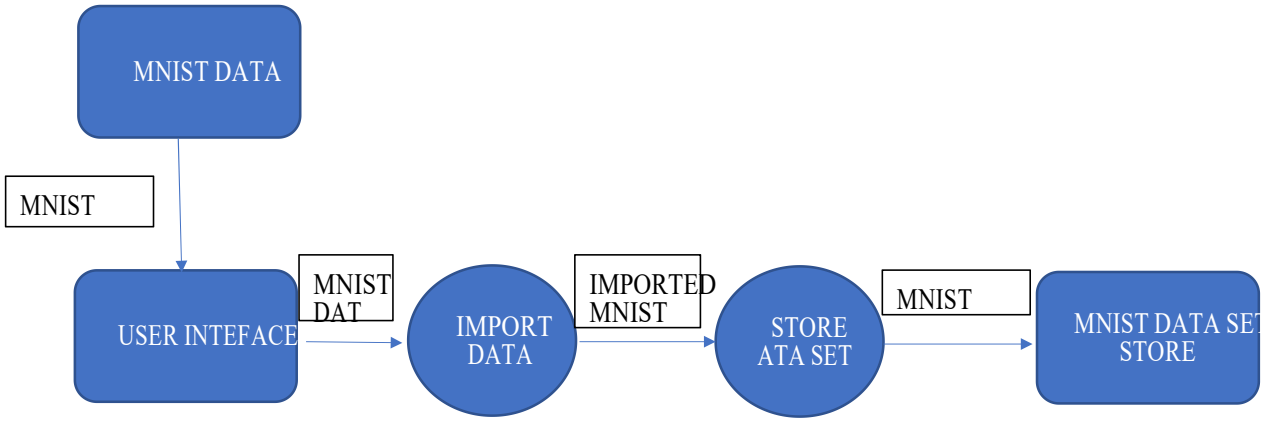
DFD Level-1

The DFD Level-1 consists of 2 external entities, the GUI and the Output, along with five process blocks and 2 data stores MNIST data and the Input image store, representing the internal workings of the CNN for Digit Recognition System. Process block imports MNIST data from library. Process block imports the image and process it and sends it to block where regression model is built. It sends objects with probabilities to CNN where weights are updated and multiple layers are built. Block trains and evaluates the model to generate output.



DFD Level-2

The DFD Level-2 for import data(figure 4) consists of two external data and one entity UI along with three process blocks, representing the three functionalities of the CNN for Digit Recognition System. It imports data from MNIST data store and stores on the system.



!

5.2 Solution and technical architecture

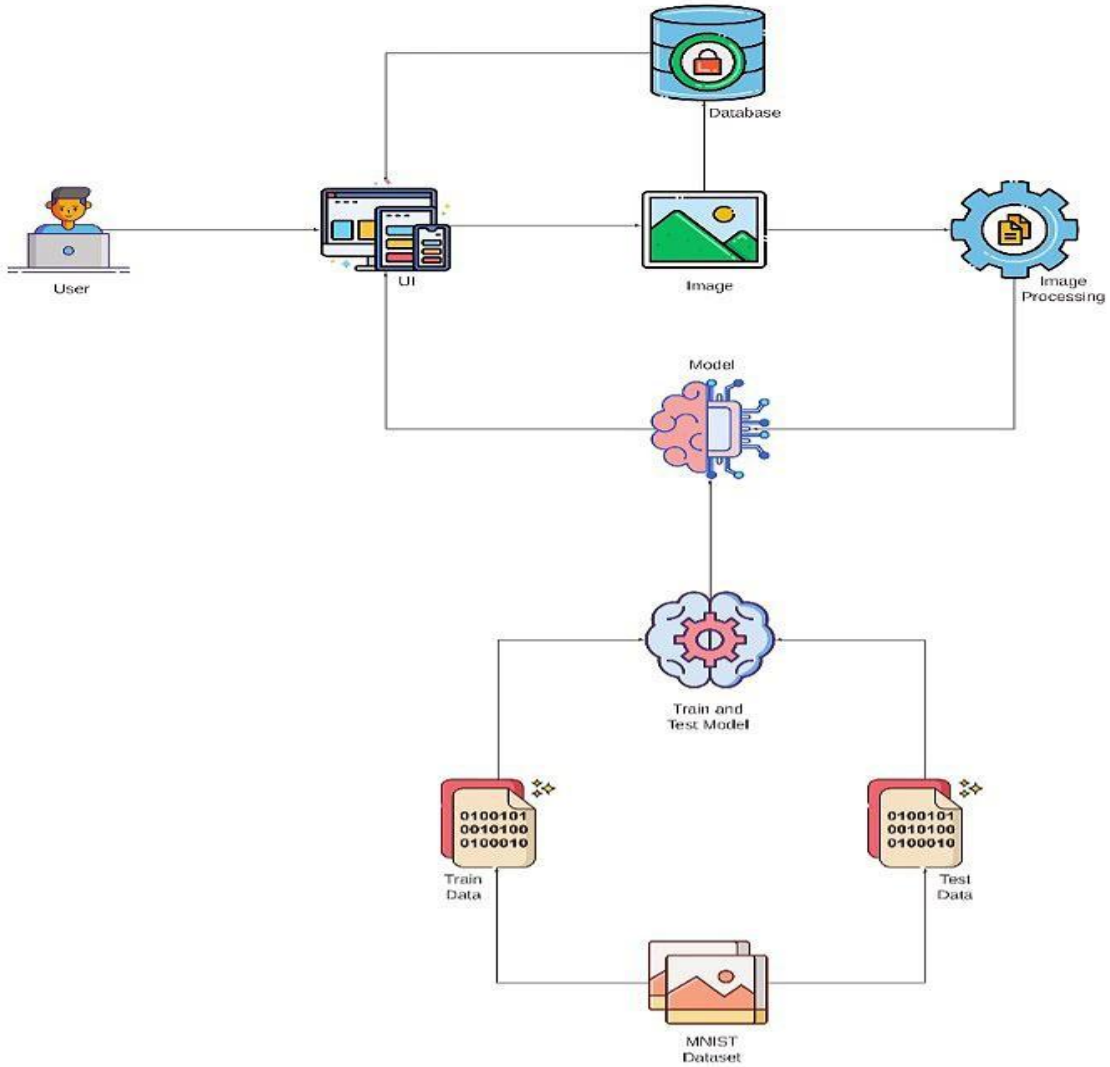


Table-1: Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application	HTML, CSS, JavaScript / Angular Js /etc.
2.	Application Logic-1	Logic for a process in the application	Java / Python
3.	Application Logic-2	Logic for a process in the application	IBM Watson STT service
4.	Database	Data Type, Configurations etc.	MySQL, NoSQL, etc.
5.	Cloud Database	Database Service	IBM DB2 etc.
6.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem

7.	Machine Learning Model	Purpose of Machine Learning Model	Object Recognition Model, etc.
8.	Infrastructure (Server)	Application Deployment on Local System (AI)	Local, Kubernetes, etc.

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Framework from python	Flask, Pytorch.
2.	Security Implementations	Cloud access and Authentication.	IBM Security, SSL Certificate.
3.	Scalable Architecture	Open source container orchestration engine for automating deployment scaling and managing.	Kubernetes.

4.	Availability	To customize and configure using HTML and JSON.	HTML, JSON.
5.	Performance	Caching Helps to improve the performance and Throughput of the application	Browser Cache

5.3 user stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-2
		USN-3	As a user, I can register for the application through gmail or facebook	I can register & access the dashboard with Facebook Login	Medium	Sprint-2
	Login	USN-4	As a user, I can log into the application by entering email & password	I can login to the application	High	Sprint-1
	Dashboard	USN-5	Go to dashboard and refer the content about our project	I can read instructions also and the home page is user-	Low	Sprint-1

				friendly.		
	Upload Image	USN-6	As a user, I can able to input the images of digital documents to the application	As a user, I can able to input the images of digital documents to the application	High	Sprint-3
	Predict	USN-7	As a user I can able to get the recognised digit as output from the images of digital documents or images	I can access the recognized digits from digital document or images	High	Sprint-3
		USN-8	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.	Medium	Sprint-4
Customer (Web user)	Login	USN-9	As a user, I can use the application by entering my email, password.	I can access my account	Medium	Sprint-4
Customer Care Executive	Dashboard	USN-10	upload the image	Recognize and get the output	High	Sprint-1
Administrator	Security	USN-11	updated the features	checking the security	Medium	Sprint-1

CHAPTER 6

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	V	Team Members
Sprint-1	Data Collection	USN-1	As a user, I collect the dataset from various sources of different handwritings.	11	Low	Sneha H Subashree M
Sprint-1	Data Preprocessing	USN-2	As a user, I load the dataset and splitting the data for training and testing.	10	Medium	Sneha H Subaja E
Sprint-2	Model Building	USN-3	As a user, I will get an application with ML model which gives high accuracy of recognized handwritten digit.	5	High	Sneha H Sowmiya B Subashree M
Sprint-2	Add CNN layers	USN-4	Creating the model and adding the input, hidden, and output layers to it.	5	High	Sneha H Subashree M Subaja E
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	Sneha H Sowmiya B
Sprint-2	Train & test the model	USN-6	As a user, let us train our model with our image dataset.	4	Medium	Sneha H Sowmiya B Subaja E

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	Sneha H Subaja E
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.	8	High	Sneha H Subashree M Sowmiya B
Sprint-3		USN-9	As a user, I can know the details of the fundamental usage of the application.	5	Low	Sneha H Sowmiya B Subaja E
Sprint-3		USN-10	As a user, I can see the predicted / recognized digits in the application.	7	Medium	Sneha H Subashree M
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	Sneha H Subaja E Subashree M Sowmiya B
Sprint-4	Cloud Deployment	USN-12	As a user, I can access the web application and make the use of the product from anywhere.	7	High	Sneha H Subashree 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	21	6 Days	24 Oct 2022	29 Oct 2022	21	29 Oct 2022
Sprint-2	18	6 Days	31 Oct 2022	05 Nov 2022	18	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	17	6 Days	14 Nov 2022	19 Nov 2022	17	19 Nov 2022

CHAPTER 7

7. CODING & SOLUTIONING

7.1 Feature 1- FLASK FILE UPLOADING

Handling file upload in Flask is very easy. It needs an HTML form with its enctype attribute set to 'multipart/form-data', posting the file to a URL. The URL handler fetches file from request.files[] object and saves it to the upload folder.

```
import numpy as np
import os
from PIL import Image
from flask import Flask, request, render_template
from werkzeug.utils import secure_filename
from keras.models import load_model
UPLOAD_FOLDER = 'C:/Users/Dell/PycharmProjects/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")
            img = img.resize((28, 28)) # resizing of input image
            im2arr = np.array(img) # converting to image
            im2arr = im2arr.reshape(1, 28, 28, 1)
            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)
```

7.2 FEATURE 2 – UPLOAD IMAGE WITH PREVIEW

A preview refers to a feature that lets you glimpse or view something in part or whole without it being opened. A picture preview would show a small version of the picture and give you a good idea what each picture is without opening each picture it is a useful feature created using JavaScript.

```
<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>

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

7.3 FEATURE 3 – CLEAR IMAGE

This feature can be used to clear the image if we uploaded a wrong image or if we need to change the image. The clear button clears both the image value and the preview of the image in script tag.

<script>

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

CHAPTER 8

8. TESTING

8.1 User Accepting Testing

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the [Handwritten Digit Recognition] project at the time of the release to User Acceptance Testing (UAT)

2. Defect Analysis

This report execute our user scheduling and their approaches.

Task	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
Login	5	1	2	4	12
Index page.html	4	1	5	5	15
Model building	1	0	3	0	4
Execute the model	1	0	0	1	2
Flask(app.py)	1	2	2	2	7
Flask(IBM app.py)	0	0	1	0	1
Deploying the model	0	0	1	1	2
Totals	12	4	12	13	43

3. Test Case Analysis

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

Section	Total Cases	Not Tested	Fail	Pass
Index	1	0	0	pass

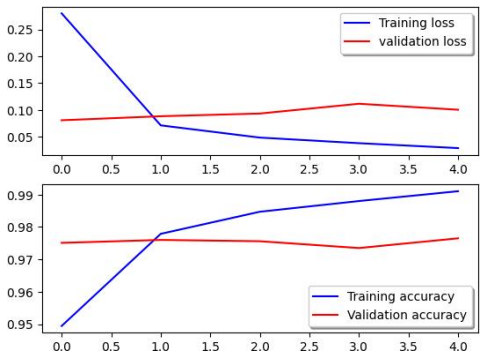
CHAPTER 9

9.RESULTS

9.1 Performance testing

S.No.	Parameter	Values	Screenshot

1.	Model Summary		`metrics = model.evaluate(x_test, y_test, verbose=0)`
			`print("Metrics (Test loss &Test Accuracy) : ")`
			`print(metrics)`
			
			`Metrics (Test loss &Test Accuracy) :`
			`[0.09330402314662933, 0.9800000190734863]`

2.	Accuracy	Training Accuracy - 99% Validation Accuracy - 97%	

CHAPTER 10

10.ADVANTADES AND DISADVANTAGES

10.1 Advantages

1) the system not only produces a classification of the digit but also a rich description of the instantiation parameters which can yield information such as the writing style;

2) the generative models can perform recognition driven segmentation;

3) the method involves a relatively.

10.2 Disadvantages

Plus, sometime characters look very similar, making it hard for a computer to recognize accurately. Joined up handwriting is another challenge for computers. When your letters all connect, it makes it hard for computers to recognize individual characters.

CHAPTER 11
CONCLUSION

This project demonstrated a web application that uses machine learning to recognize 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 numberplates of vehicles, processing bank cheque amounts, numeric entries in forms fifilledup by hand (tax forms) and so on.

CHAPTER 12

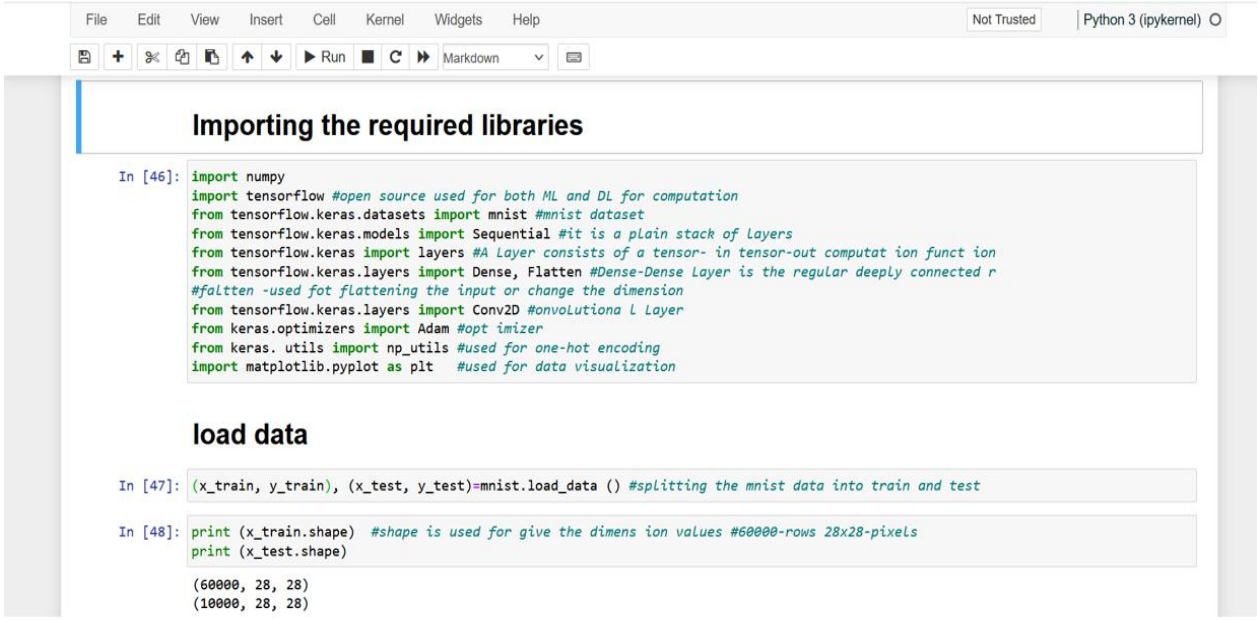
12 Future Scope

We believe that our proposed model can further be applied to other datasets. In contracts as a future work. we find that it is worth taking further actions to improve our model performance in terms of how to perfectly learn and extract the local features in the hidden layers, and how to enhance recognition ability in the fully connected layers to avoid mislabeling problems.

APPENDIX

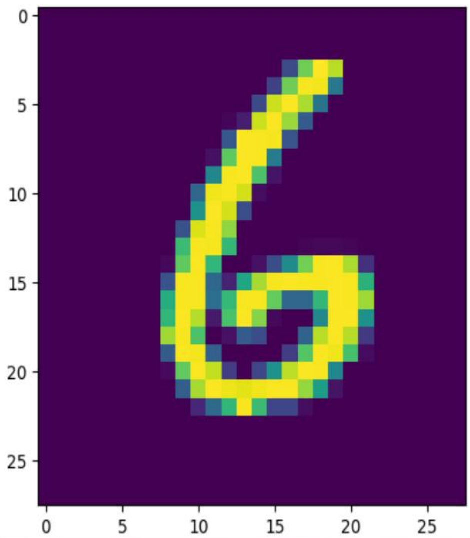
SOURCE CODE

MODEL CREATION:



```
In [49]: x_train[0]

Out[49]: 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,
                  0,   0],
                [  0,   0,   0,   0,   0,   0,   0,  49, 238, 253, 253, 253, 253,
                253, 253, 253, 253, 251,  93,  82,  82,  56,  39,   0,   0,   0,
                  0,   0],
```

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

Out[50]: <matplotlib.image.AxesImage at 0x222c71e0250>
```



Reshaping Dataset

In [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

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

In [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

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

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

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

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

Compiling the model

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

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

Train the model

In [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
```

Out[63]:
```
<keras.callbacks.History at 0x222d90d9db0>
```

Observing the metrics

In [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

In [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]]
```

Save The model

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

CNNPREDICTION:

[66]:
```
from tensorflow.keras.models import load_model
from keras.preprocessing import image
from PIL import Image
import numpy as np
```

[67]:
```
model = load_model("mnistCNN.h5")
```

[79]:
```
import os, types
import pandas as pd
from botocore.client import Config
import ibm_boto3

def __iter__(self): return 0

# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.
# You might want to remove those credentials before you share the notebook.
cos_client = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='KGntsq0dqQGaRo45fB0qk4gb6tCtfW5ky--K7G5VDIUb',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.us-east.cloud-object-storage.appdomain.cloud')

bucket = 'handwrittenforsprint4'
object_key = '55.png'

streaming_body_1 = cos_client.get_object(Bucket=bucket, Key=object_key)['Body']

# Your data file was loaded into a botocore.response.StreamingBody object.
# Please read the documentation of ibm_boto3 and pandas to learn more about the possibilities to load the data.
# ibm_boto3 documentation: https://ibm.github.io/ibm-cos-sdk-python/
# pandas documentation: http://pandas.pydata.org/
```

[80]:
```
img = Image.open(streaming_body_1).convert("L") # convert image to monochrome
img = img.resize( (28,28) ) # resizing of input image
```

[81]:
```
img
```

[81]:

[82]:
```
im2arr = np.array(img) #converting to image
im2arr = im2arr.reshape(1, 28, 28, 1) #reshaping according to our requirement
```

[83]:
```
pred = model.predict(im2arr)
print(pred)
```

```
1/1 [==============================] - 0s 104ms/step
[[1.3385103e-14 7.3480496e-13 4.8976140e-08 6.2034937e-04 1.0703821e-11
  9.8845661e-01 1.8612450e-08 1.0921785e-02 5.2048023e-07 7.1090477e-07]]
```

[84]:
```
print(np.argmax(pred, axis=1)) #printing our Labels
```

```
[5]
```

[ ]:

TRAIN THE MODEL ON IBM:

```
[2]: from ibm_watson_machine_learning import APIClient
     credentials ={
         "url":"https://jp-tok.ml.cloud.ibm.com",
         "apikey":"KGntsq0dqQGaRo45fB0qk4gb6tCtfW5ky--K7G5VDIUb"
     }
     client = APIClient(credentials)
     client

[2]: <ibm_watson_machine_learning.client.APIClient at 0x1fe1bffc8b0>

[ ]: client.spaces.get_details()

[4]: def guid_from_space_name(client,deploy):
       space = client.spaces.get_details()
       return (next(item for item in space['resources'] if item['entity']['name']==deploy)['metadata']['id'])

[5]: space_uid = guid_from_space_name(client,'sprint4dep')
     print("Space UID = " + space_uid)

     Space UID = d701f311-0c80-4787-82a6-3ab67ec9118e

[6]: client.set.default_space(space_uid)

[6]: 'SUCCESS'
```

```
[ ]: client.software_specifications.list(limit=100)

[8]: software_space_uid = client.software_specifications.get_uid_by_name('tensorflow_rt22.1-py3.9')
     software_space_uid

[8]: 'acd9c798-6974-5d2f-a657-ce06e986df4d'

[ ]: model_details = client.repository.store_model(model='handwritten-digit-recognition-model_new.tgz',meta_props={
         client.repository.ModelMetaNames.NAME:"CNN Digit recognition model",
         client.repository.ModelMetaNames.TYPE:"tensorflow_2.7",
         client.repository.ModelMetaNames.SOFTWARE_SPEC_UID:software_space_uid
     })

[ ]: model_details

53]: model_id = client.repository.get_model_id(model_details)
     model_id

53]: 'bf3f45b6-c032-45ce-9cbc-64541fe7bced'

[ ]: client.repository.download(model_id,'DigitRecog_IBM_model.tar.gz')

55]: ls

      Volume in drive C has no label.
      Volume Serial Number is 7C60-CE44

      Directory of C:\Users\hp\Downloads\IBM-Project-20414-1659718946\Project-Development-Phase\Sprint 4\models

     15-Nov-22  06:47 AM    <DIR>          .
     15-Nov-22  06:47 AM    <DIR>          ..
     15-Nov-22  06:47 AM         2,319,342 DigitRecog_IBM_model.tar.gz
     15-Nov-22  06:45 AM         2,319,342 handwritten-digit-recognition-model_new.tgz
     15-Nov-22  06:43 AM         2,475,168 mnistCNN.h5
                    3 File(s)      7,113,852 bytes
                    2 Dir(s)  44,180,271,104 bytes free
```

HOME PAGE(HTML) – index.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',"");
            });
        });


</script>
<body>
  <h1 class="welcome">A Novel Method For Handwritten Digit Recognition System
  <div id="team_id">TEAM ID : PNT2022TMID27693</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>
      Handwritten digit recognition is the process to provide the ability to machines to recognize human handwritten digits.
      It is not an easy task for the machine because handwritten digits are not perfect, vary from person-to-person, and
      can be made with many different flavors..</p>
      <br>
    <p> 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>
```

HOME PAGE(CSS) – style.css

```
#clear_button{
  margin-left: 15px;
  font-weight: bold;
  color: rgb(204, 255, 0);
}

#confidence{
  font-family: "Times New Roman", Times, serif;
  margin-top: 7.5%;
}

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

.welcome{
   text-align: center;
   position: relative;
   color: rgba(240, 255, 240, 0.588);
   background-color: rgb(165, 47, 255);
   padding-top: 1%;
   padding-bottom: 1%;
   font-weight: bold;
   font-family: "Times New Roman", Times, serif;
}

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

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

```
}

#prediction_heading{
  font-family: "Times New Roman", Times, 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: rgba(238, 238, 238, 0.386);
    border: 1px solid rgba(136, 136, 136, 0);
    margin-top: 20px;
    margin-bottom: 20px;
}

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

.heading{
  font-family: "Times New Roman", Times, serif;
  font-weight: 700;
  font-size: 2rem;
  display: inline;
```

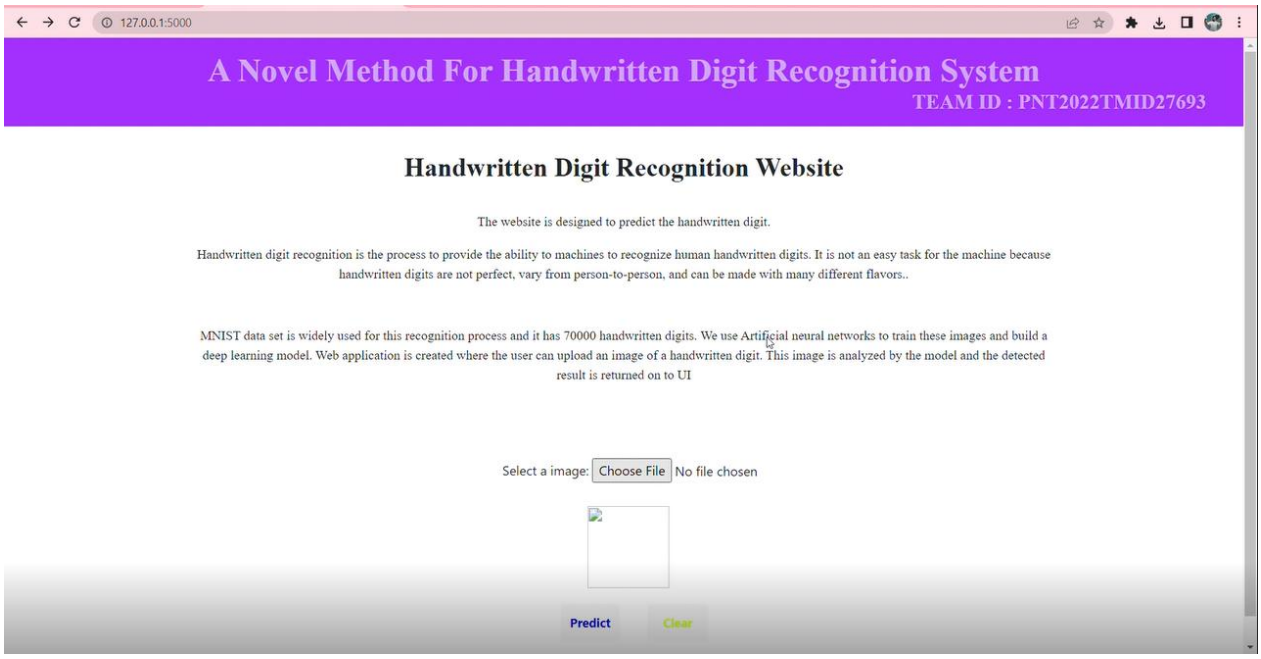
PREDICT PAGE (HTML) – predict.html

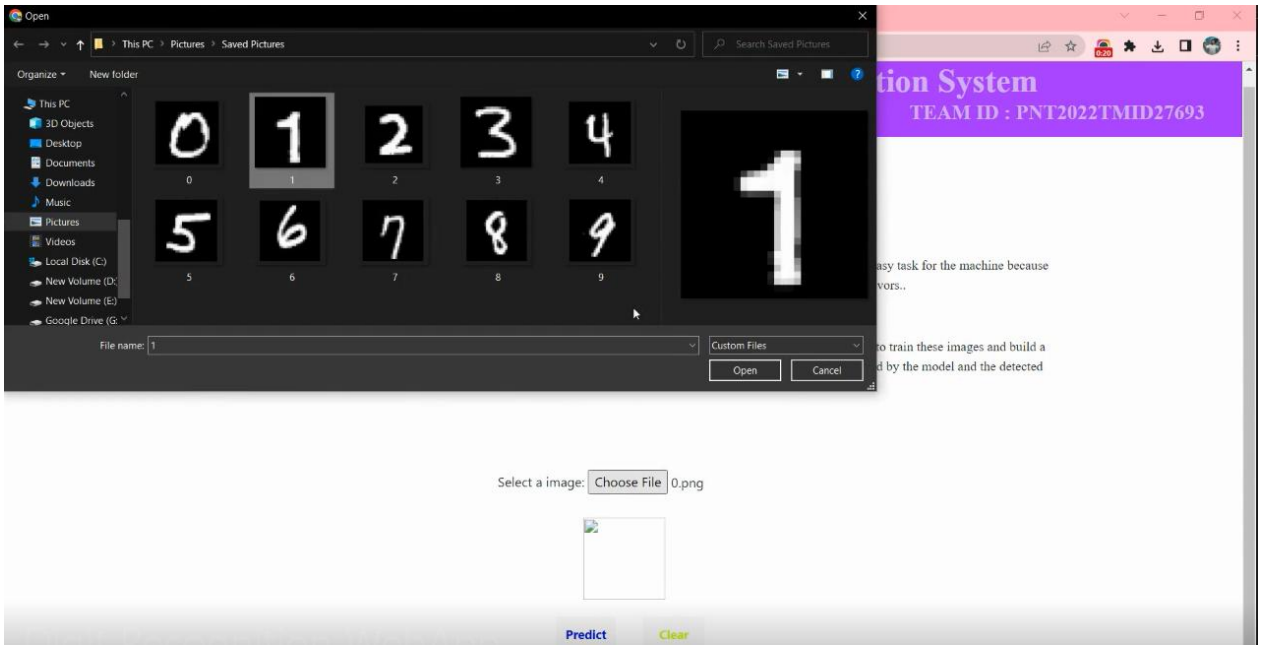
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Prediction</title>
</head>
<style>
    body{
     background-image: url('static/images/index6.jpg');
     background-repeat: no-repeat;
     background-size: cover;
    }
    #rectangle{
     width:400px;
     height:150px;
     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>
```

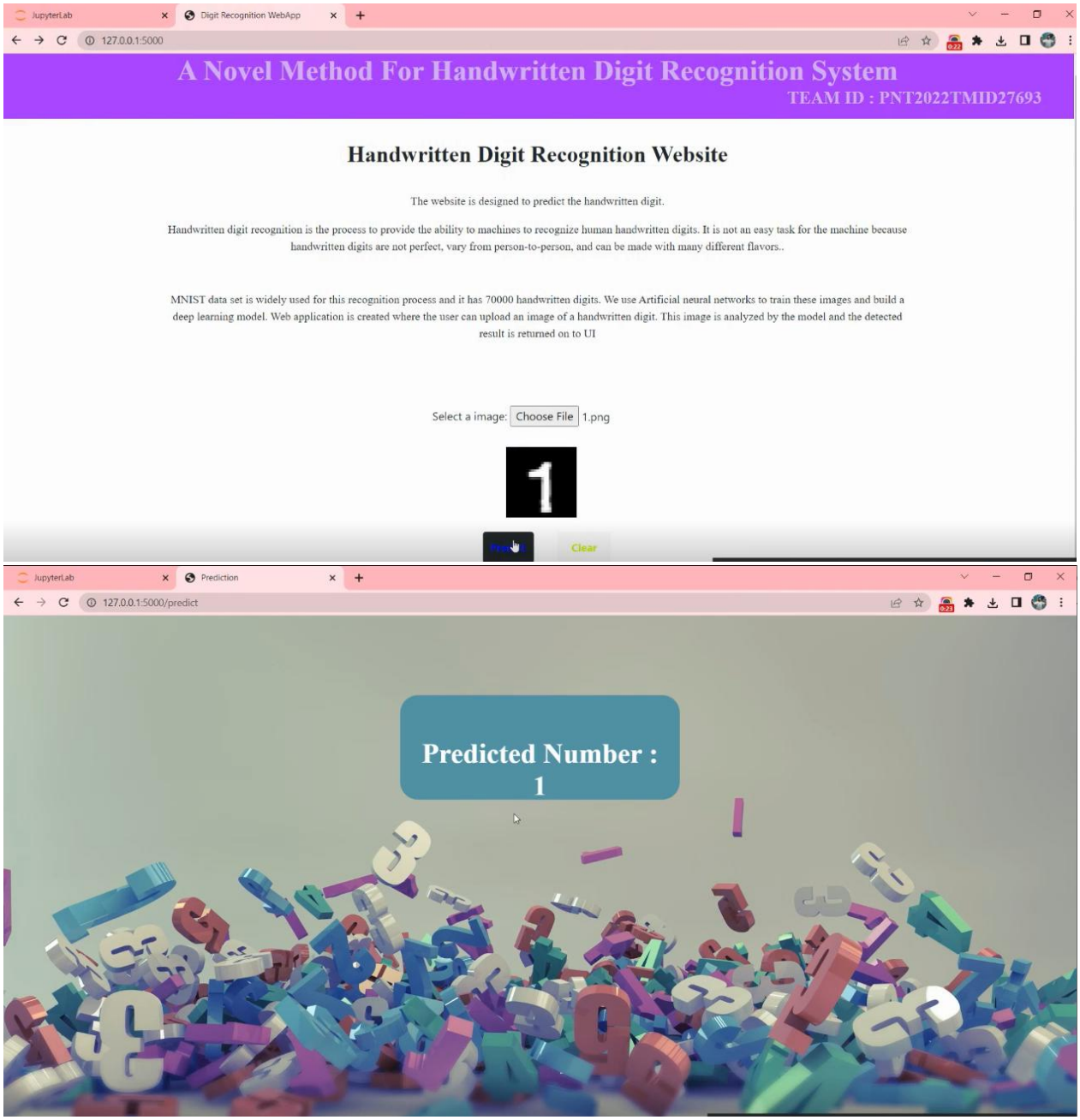
FLASK APP - app.py

```python
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/hp/Downloads/IBM-Project-12154-1659438899/Project Development Phase/Sprint 3/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=True, threaded=False)

```

SCREENSHOTS:







GITHUB LINK:
https://github.com/IBM-EPBL/IBM-Project-12154-1659438899