A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

IBM PROJECT REPORT

Submitted by

Dineshraja A(913119104021)

Nithees Kumar M(913119104063)

Raghul S (913119104074)

Shiva Kumar A (913119104095)

in partial fulfillment for the award of the degree Of

BACHELOR OF ENGINEERING IN COMPUTER SCIENCE ENGINEERING VELAMMAL COLLEGE OF ENGINEERING AND TECHNOLOGY MADURAI

ANNA UNIVERSITY : CHENNAI 600 025

JUNE 2022

Project Report

A Novel Method for Handwritten digit Recognition system

1. INTRODUCTION

1.1 Project Overview :

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 digitised to reduce human effort. Hence, there comes a need for handwritten digit recognition in many real-time applications. The 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 analysed by the model and the detected result is returned on to the UI.

1.2 Purpose :

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 almost mimic the human brain and are a key ingredient in the image processing field.

2.LITERATURE SURVEY

2.1 Existing Problem :

Humans can see and visually sense the World around them by using there eyes and brains.Computer vision works on enabling computers to see and process images in the same way that human vision does.Seval algorithms developed in the area of the computer vision to recognize images.The goal of our work will be to create a model that will be able to identify and determine the Handwritten digit from its image with better accuracy.

2.2 References :

[1]. M. Wu and Z. Zhang, Handwritten Digit Classification using the MNIST Dataset, 2010.

[2]. A. Dutta and A. Dutta, Handwritten digit recognition using deep learning, International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), vol. 6, no. 7, July 2017

[3]. Al Maadeed, Somaya, and Abdelaali Hassaine, Automatic prediction of age, gender, and nationality in offline handwriting. EURASIP Journal on Image and Video Processing, no. 1 2014

[4]. Gaurav Jain, Jason Ko, Handwritten DigitsRecognition, Project Report, University of Toronto, 11/21/2008.

[5]. Hamid, Norhidayu Abdul, and NilamNur Amir Sjarif,Handwritten recognition using SVM, KNN and neural network, arXiv preprint arXiv:1702.00723 (2017).

[6]. R.G.Mihalyi, Handwritten digit classification using support vector machines, 2011.

[7]. Z. Dan, C. Xu, The Recognition of Handwritten Digits Based on BP Neural Networks and the Implementation on Android, In: 3rd International Conference on Intelligent System Design and Engineering Applications, pp. 1498-1509, 2013.

2.3 Problem Statement Definition :

Handwritten Digit Recognition is the capability of a computer to fete the mortal handwritten integers from different sources like images, papers, touch defences, etc, and classify. them into 10 predefined classes (0-9). This has been a Content of bottomless- exploration in the field of deep literacy. Number recognition has numerous operations like number plate recognition, postal correspondence sorting, bank check processing, etc . In Handwritten number recognition, we face numerous challenges because of different styles of jotting of different people as it is not an Optical character recognition. This exploration provides a comprehensive comparison between different machine literacy and deep literacy algorithms for the purpose of handwritten number recognition. For this, we've used Support Vector Machine, Multilayer Perceptron, and Convolutional . Neural Network. The comparison between these algorithms is carried out on the basis of their delicacy, crimes, and testing- training time corroborated by plots and maps that have been constructed using matplotlib for visualisation.

3.IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas :

An empathy map canvas is a more in-depth version of the original empathy map, which helps identify and describe the user’s needs and pain points. And this is valuable information for improving the user experience.

An empathy map canvas helps brands provide a better experience for users by helping teams understand the perspectives and mindset of their customers. Using a template to create an empathy map canvas reduces the preparation time and standardises the process so you create empathy map canvases of similar quality.

THINK AND FEEL	Whether it will take more time to recognize the digits	Whether it will recognize other than English	I wish there was an option to identify numeral in image
HEAR	The handwritten digit recognition system is to convert handwritten digits into machine readable format	It is the process to provide the ability to machines to recognize human handwritten digits	Human effect can be reduced to seeing handwritten digits
SEE	Using handwritten digit recognition can Identify the zip codes on mall for postal mail sorting	handwritten digit recognition has a great Importance such as online handwriting recognition	The effective and reliable approaches for recognition for handwritten digits and make bank operation easy and error free
SAY AND DO	I can identify someone's handwriting	I can recognize the handwritten digits in an accuracy rate	I can identify any digits in any handwritten notes or books

PAIN	Not always accurate	Unique style of writing	Poor images of text
GAIN	Electronic data storage	Easier data retrieval	Historical preservation

3.2 Ideation & Brainstorming :

Ideation is often closely related to the practice of brainstorming, a specific technique that is utilised to generate new ideas. A principal difference between ideation and brainstorming is that ideation is commonly more thought of as being an individual pursuit, while brainstorming is almost always a group activity.

Ideas :

Dineshraja A	Features based on shapes analysis of digits	Neural network is used to train and identify the images	It detects the handwritten digits 90% accurately
Raghul S	As it is based on Al and programing improves,they are often correct	Ability of the computer to recognize the handwritten digits from different source	Widely used in online digit recognition

Nithees Kumar M	Convolution neural network from machine learning more accurate	Handwritten digit recognition converts the text digit into electrical form	Major application postal mail sorting
Shiva Kumar A	As it is in the form of machine code it requires less space for storing	It is recognise handwritten digits in a faster way	Can recognize the digits in different language

Group ideas :

As it is based on Al programming improves they are often correct.It detects the handwritten digits 90% accurately.Widely used in online digit recognition.Handwritten digit recognition converts the text digit into electrical form.Major application postmail sorting.As it is in the form of machine code it requires less space for storing.It is also used for the automatic processing such as bank check process.The Al will be train and testing the handwritten numbers from O to 9.It contains 60,000 handwritten digits for training and testing.It recognize handwritten digits in a faster way.It reduces the human efforts.

3.3 Proposed Solution Template:

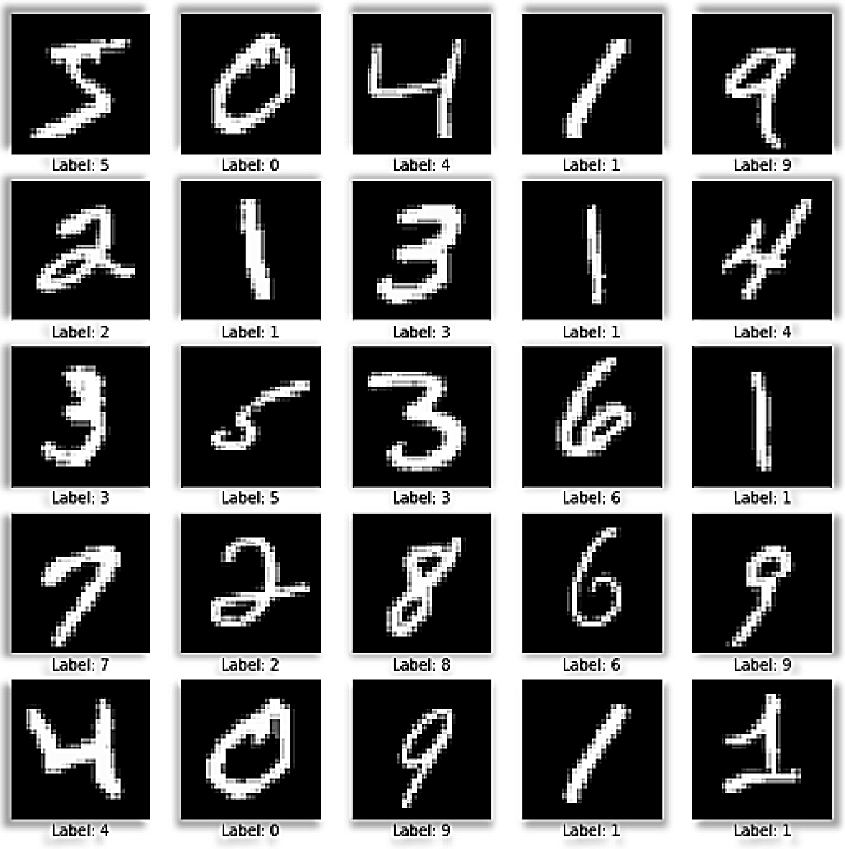
PARAMETER	DESCRIPTION
Problem Statement (Problem to be solved)	Hard task for the machine to be able to recognize the handwritten digits because handwritten digits are not perfect and can be made with many different writing styles.
Idea / Solution description	A handwriting digit recognition system is to convert handwritten digits into machine readable formats. It is the process to provide the ability to machines to recognize human handwritten digits
Novelty / Uniqueness	It recognizes the digits based on the analysis of shape and the thickness of the numerical image which is very effective and good accuracy.
Social Impact / Customer Satisfaction	It is used in the detection of vehicle numbers, banks for reading cheques, post offices for arranging letters, and many other tasks. It is time consuming and the fastest method.

Business Model (Revenue Model)	objective of this is to ensure effective and reliable approaches for recognition of handwritten digits in online digits recognition such as banking operations easier and error free
Scalability of the Solution	Handwritten digit recognition becomes vital scope and it is appealing many researchers because of it using in variety of machine learning and computer vision applications

3.4 Problem Solution fit :

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 digitised to reduce human effort. Hence, there comes a need for handwritten digit recognition in many real-time applications. The 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 analysed by the model and the detected result is returned on to UI The MNIST Handwritten Digit Recognition Dataset contains 60,000 training and 10,000 testing labelled handwritten digit pictures. Each picture is 28 pixels in height and 28 pixels wide, for a total of 784

(28×28) pixels. Each pixel has a single pixel value associated with it. It indicates how bright or dark that pixel is (larger numbers indicate darker pixels). This pixel value is an integer ranging from 0 to 255.



PROCEDURE :

1.Install the latest TensorFlow library.

2.Prepare the dataset for the model.

3.Develop Single Layer Perceptron model for classifying the handwritten digits.

4.Plot the change in accuracy per epochs. .Evaluate the model on the testing data.

5.Analyse the model summary.

6.Add hidden layer to the model to make it Multi-Layer Perceptron. Add Dropout to prevent overfitting and check its effect on accuracy. Increasing the number of Hidden Layer neuron and check its effect on accuracy.

7.Use different optimizers and check its effect on accuracy. Increase the hidden layers and check its effect on accuracy.

8.Manipulate the batch size and epochs and check its effect on accuracy.

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.

Handwritten digit recognition using MNIST dataset is a major project made with the help of Neural Network. It basically detects the scanned images of handwritten digits.

Approach:

We will approach this project by using a three-layered Neural Network.

The input layer:It distributes the features of our examples to the next layer for calculation of activations of the next layer.

The hidden layer: They are made of hidden units called activations providing nonlinear ties for the network. A number of hidden layers can vary according to our requirements.

The output layer: The nodes here are called output units. It provides us with the final prediction of the Neural Network on the basis of which final predictions can be made.

A neural network is a model inspired by how the brain works. It consists of multiple layers having many activations, this activation resembles neurons of our brain. A neural network tries to learn a set of parameters in a set of data which could help to recognize the underlying relationships. Neural networks can adapt to changing input; so the network generates the best possible result without needing to redesign the output criteria.

WORKING

- Neural Networks receive an input and transform it through a series of hidden layers.
- Each hidden layer is made up of a set of neurons, where each neuron is fully connected to all neurons in the previous layer.
- Neurons in a single layer function completely independently. The last fully connected layer is called the "output layer".

Convolution Layer: The Convolutional layer is the core building block of a CNN.

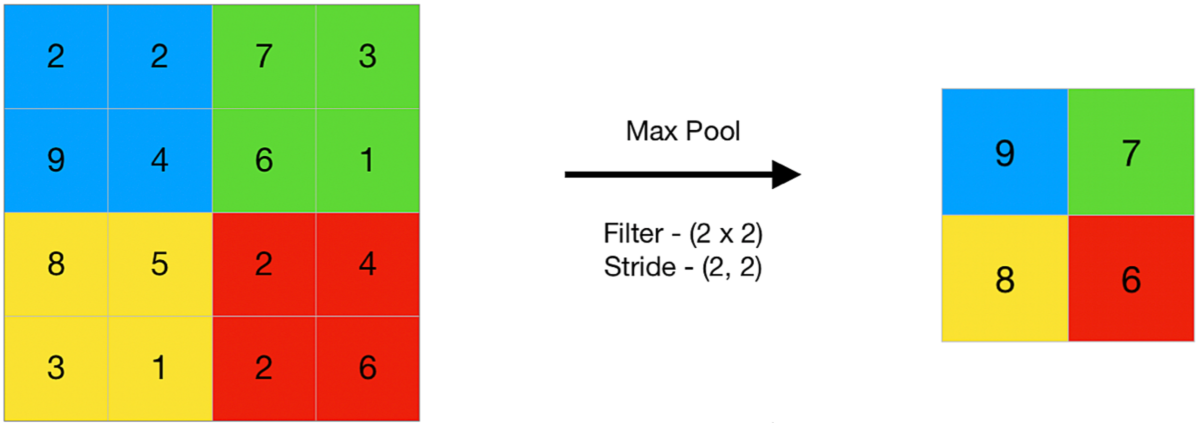
The layer's parameters consist of a set of learnable filters (or kernels), which have a small receptive field, but extend through the full depth of the input volume.

During the forward pass, each filter is convolved across the width and height of the input volume, computing the dot product between the entries of the filter and the input and producing a 2-dimensional activation map of that filter.
As a result, the network learns filters that activate when they see some specific type of feature at some spatial position in the input..

Feature Extraction:
All neurons in a feature share the same weights .In this way all neurons detect the same feature at different positions in the input image. Reduce the number of free parameters.
Subsampling Layer: Subsampling, or down sampling, refers to reducing the overall size of a signal .The subsampling layers reduce the spatial resolution of each feature map. Reduce the effect of noises and shift or distortion invariance is achieved.

Pooling layer: It is common to periodically insert a Pooling layer in-between successive Conv layer in a Convent architecture. Its function is to progressively reduce the spatial size of the representation to reduce the number of parameters and computation in the network, and hence to also control overfitting. The Pooling Layer operates independently on every depth slice of the input and resizes it spatially, using the MAX operation.

TensorFlow: TensorFlow is an open-source machine learning library for research and production. TensorFlow offers APIs for beginners and experts to develop for desktop, mobile, web, and cloud. See the sections below to get started. By scanning the numerical digit and convert into png format using python3 command in terminal we can get text output and sound output.



2	2	7	3
9	4	6	1
8	5	2	4
3	1	2	6

9	7
8	6

Results :

As with any work or project taken up in the field of machine learning and image processing, we are not considering our results to be perfect. Machine learning is a constantly evolving field and there is always room for improvement in your methodology; there is always going to be another new approach that gives better results for the same problem. The application has been tested using three models: Multi-Layer Perceptron (MLP), Convolution Neural Network (CNN). With each model we get a different accuracy of the classifier which shows which one is better.

4.REQUIREMENT ANALYSIS

4.1 Functional requirement :

The system should support the three stages of the writing process, these are planning, translation (writing), and review. Within these stages it should provide ideas for planning, allow for fast and accurate transcription, and allow for the easy movement, alteration and deletion of characters, words and phrases. It should include some spelling support and should incorporate file-handling facilities. The recognition component should be able to work even when children write slowly, it should be able to deal with 'wobbly' writing, and should be able to recognise common misconstructions of characters.
Data requirements :The system needs to be able to cope with multiple users, each user may have multiple documents and each document may have many files associated with it. These may be text files as well as ink files, and the text and ink files will be related. Each text or ink file may have updated versions following an editing process

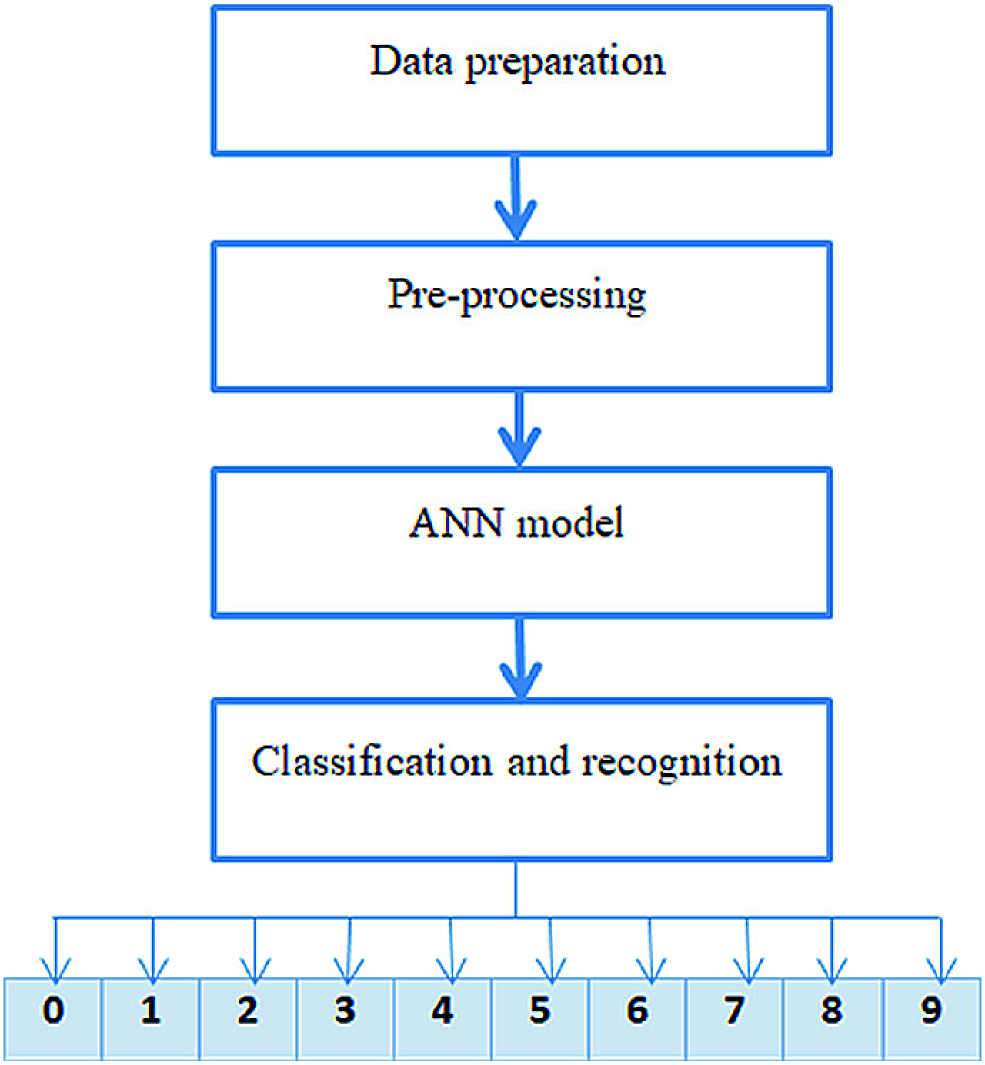
4.2 Non-Functional requirements :

For use in the classroom – the system needs to be robust, easy to learn and have on line help. It should not need an adult to make it work. It should be designed to work on a standard PC with a tablet and pen. The interface may be used in a noisy environment, or in a quiet environment – this implies that any sound output needs to be non-essential and easily turned off.Children are likely to be working collaboratively so large font sizes on the screen are necessary.Users will be novices at the start but will quickly acquire competence.The primary users will be children of normal educational ability and without any motor dysfunction in their upper limbs.They will have reasonable vision and it is expected that the children will be able to read, but not with confidence.For this reason the words and language need to be kept simple and spoken output should be an option on the help screens.Children cannot be expected to be able to spell well, nor to be able to write cursively (although both will result in a better experience at the interface!).They can be expected to be able to hold and manipulate a pen, and to be able to construct even sized, legible alphabetic characters in upper and lower case.Secondary users are adult helpers. These can be expected to be literate, and to be able to use a mouse driven GUI interface. They cannot be expected to be familiar with the handwriting recognition processes nor with the file handling of the application.

5.PROJECT DESIGN

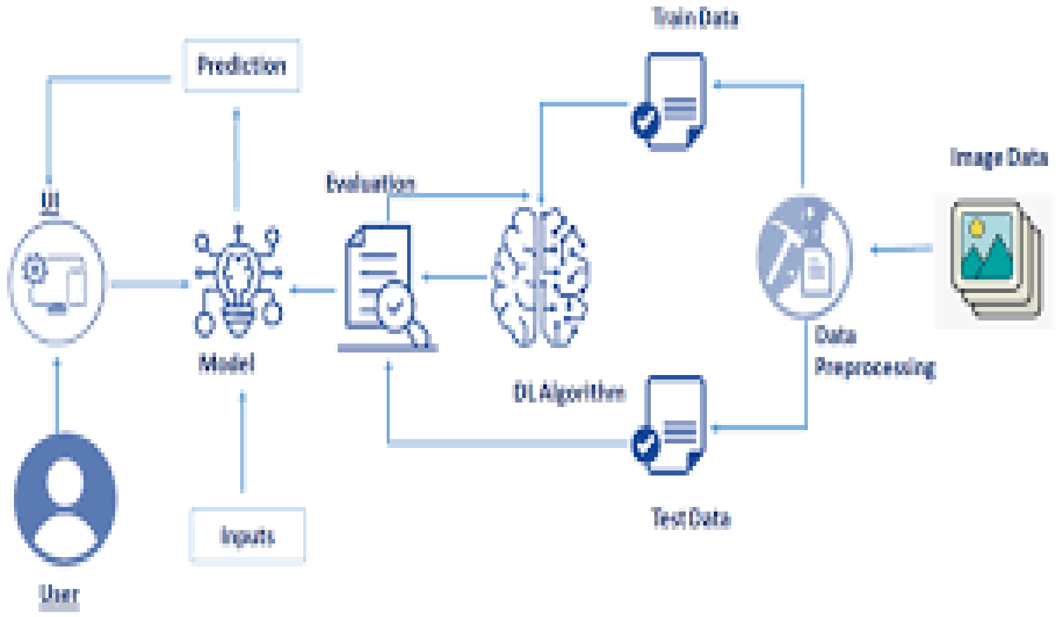
5.1 Data Flow Diagram :

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. Example: DFD Level 0 (Industry Standard)



5.2 Solution & Technical Architecture :

Given that everyone in the world has their own writing style, handwriting detection is one of the most intriguing research projects now underway. It is the computer's capacity to automatically recognise and understand handwritten figures or letters. Because of advances in science and technology, everything is being digitalized in order to reduce human effort. As a result, handwritten digit identification is required in many real-time applications. The MNIST data collection, which contains 70000 handwritten digits, is commonly employed in this recognition process. To train these photos and create a deep learning model, we use artificial neural networks. A web application is developed that allows the user to upload an image of a handwritten digit.



5.3 User Stories :

Use the below template to create product backlog and sprint schedule

S.NO	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
1	Data Collection	USN-1	As a user, I can collect the dataset from various resources with different handwritings	10	Low	Dineshraja A
2	Data Pre-processing	USN-2	As a user, I can load the dataset, handling the missing data, scaling and split data into train and test	10	Medium	Raghul S
3	Model building	USN-3	As a user, I will get an application with ML model which provides higher accuracy of recognized handwritten digit	5	High	Nithees kumar M
4	Add CNN layers	USN-4	Creating a model and adding the input, hidden, and output layers to it.	5	High	Shiva Kumar A
5	Compiling the model	USN-5	With both the training data defined and model defined, it’s time to configure the learning process	2	High	Raghul S

6	Train and test the model	USN-6	As a user, let us train our model with our image dataset.	6	Medium	Nithees Kumar M
7	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	Shiva Kumar A
8	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	Dineshraja A
9		USN-9	As a user, I can know the details of the fundamental usage of the application.	5	Low	Nithees Kumar M
10		USN-10	As a user, I can see the predicted/ recognized digits in the application.	5	Medium	Shiva Kumar A
11	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	Dineshraja A
12	Cloud Deployment	USN-12	As a user ,I can access the web application and make the use of the product from	10	High	Raghul S

			anywhere			

6.PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation:

Estimation is done by the entire team during Sprint Planning Meeting. The objective of the Estimation would be to consider the User Stories for the Sprint by Priority and by the Ability of the team to deliver during the Time Box of the Sprint.

6.2 Sprint Delivery Schedule:

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	Dineshraja A
Sprint-1	Data Pre-processing	USN-2	As a user, I can load the dataset, handling the missing data, scaling and split data into train and test	10	Medium	Raghul S

Sprint-2	Model building	USN-3	As a user, I will get an application with ML model which provides higher accuracy of recognized handwritten digit	5	High	Raghul S
Sprint-2	Add CNN layers	USN-4	Creating a model and adding the input, hidden, and output layers to it.	5	High	Raghul S
Sprint-2	Compiling the model	USN-5	With both the training data defined and model defined, it’s time to configure	2	High	Raghul S

			the learning process			
Sprint-2	Train and test the model	USN-6	As a user, let us train our model with our image dataset.	6	Medium	Dineshraja A
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	Dineshraja A

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	Nithees Kumar M
Sprint-3		USN-9	As a user, I can know the details of the fundamental usage of the application.	5	Low	Nithees Kumar M
Sprint-3		USN-10	As a user, I can see the predicted/ recognized digits in the application.	5	Medium	Nithees Kumar A

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	Shiva Kumar A
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	Shiva Kumar A

7.CODING & SOLUTIONING

7.1 Feature 1 :

```
import imghdr

from flask import Flask, render_template, request

from PIL import Image

import numpy as np

from tensorflow.keras.models import load_model

import tensorflow as tfapp = Flask(__name__)

@app.route('/')

def home():

    return render_template('index.html')

@app.route('/about')

def about():

    return render_template('index.html')

@app.route('/upload')

def upload():

    return render_template('upload.html')

@app.route('/predict',methods=['POST'])
```

```
def upload_image_file():
    model = load_model("models/mnistCNN.h5")
    if request.method == 'POST':
        img = Image.open(request.files['img']).convert('L')
        img = img.resize((28,28))
        im2arr = np.array(img)
        im2arr = im2arr.reshape(1,28,28,1)
        # predict = model.predict(im2arr)
        predict = model.predict([im2arr])[0]
        predicted =  np.argmax(predict)
        acc = max(predict)
        print(predicted,acc)

    return
render_template('result.html',prediction=predicted,Accuracy=str(int(acc*100))+'%
')
if __name__ == '__main__':
    app.run(host='127.0.0.1', port=8000, debug=True)
```

Handwritten_digit_recognition.ipynb:

```
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
from tensorflow.keras.layers import Conv2D
from keras.optimizers import Adam
from keras.utils import np_utils
#loading data
(X_train, Y_train), (X_test, Y_test) = mnist.load_data()
print(X_train.shape)
print(X_test.shape)
X_train[0]
Y_train[0]
import matplotlib.pyplot as plt
plt.imshow(X_train[0])
#reshaping dataset
X_train = X_train.reshape(60000,28,28,1).astype('float32')
X_test = X_test.reshape(10000,28,28,1).astype('float32')
#onehot encoding
number_of_classes = 10
Y_train = np_utils.to_categorical(Y_train,number_of_classes)
Y_test = np_utils.to_categorical(Y_test,number_of_classes)
Y_train[0]
```

```
#Creating Model
model = Sequential()
model.add(Conv2D(64,(3,3),input_shape=(28,28,1),activation='relu'))
model.add(Conv2D(32,(3,3),activation='relu'))
model.add(Flatten())
model.add(Dense(number_of_classes,activation='softmax'))
model.compile(loss='categorical_crossentropy',optimizer='Adam',metrics=['accuracy'])
#Model fit
model.fit(X_train,Y_train,
validation_data=(X_test,Y_test),epochs=5,batch_size=32)
#Observing metrics
metrics = model.evaluate(X_test,Y_test,verbose=0)
print("Accuracy : ",metrics)
#Predicting Output
prediction = model.predict(X_test[:4])
print("Prediction : ",prediction)
print(np.argmax(prediction,axis=1))
print(Y_test[:4])
#Observing Metrics
metrics = model.evaluate(X_test,Y_test,verbose=0)
print("Metrics : ",metrics)
```

```
from tensorflow.keras.models import load_model
from PIL import Image
import numpy as np
model = load_model("models/mnistCNN.h5")
filename = 'num'
img = Image.open(filename +'.png').convert("L")
img = img.resize((28,28))
im2arr = np.array(img)
im2arr = im2arr.reshape(1,28,28,1)
Y_pred = model.predict(im2arr)
print(Y_pred)
```

7.2 Feature 2 :

templates

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <title>Handwritten Digit Recognition</title>
    <meta name="description" content="PR & Team built the Handwritten Digit
Recognition using Artificial Intelligence which supported by Nalaiya Thiran
Initiative">
```

```
    <meta name="author" content="PR & Team">
    <meta name="keywords" content="Digit Recognition using Artificial Intelligence">
    <meta name="viewport" content="width=device-width, minimum-scale=1.0">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <link href="https://fonts.googleapis.com/css?family=Open+Sans:300,400,600,700|Varela" rel="stylesheet">
    <link rel="apple-touch-icon" sizes="144x144" href="static/img/apple-touch-icon.png">
    <link rel="icon" type="image/png" sizes="32x32" href="static/img/favicon-32x32.png">
    <link rel="icon" type="image/png" sizes="16x16" href="static/img/favicon-16x16.png">
    <link rel="icon" sizes="16x16" href="assets/img/favicon.ico">
    <link rel="manifest" href="static/img/manifest.json">
    <link rel="mask-icon" href="static/img/safari-pinned-tab.svg" color="#5bbad5">
    <meta name="theme-color" content="#ffffff">
    <link rel="stylesheet" href="static/css/bootstrap.min.css" />
    <link rel="stylesheet" href="static/css/font-awesome.min.css" />
    <link rel="stylesheet" href="static/css/style.css">
```

```
</head>
<body>
    <div id="top" class="hero background-overlay">
        <div class="hero-content">
            <h1>A Novel Method for Handwritten Digit Recognition System</h1>
            <p class="hero-job"><span>Using AI</span></p>
            <p class="hero-job-desc"><a style="color:white;" href"#">PR &amp;
Team</a> </p>
        </div>
        <div class="hero-arrow page-scroll home-arrow-down">
            <a class="" href="upload">
                <i class="fa fa-angle-double-down" aria-hidden="true"></i>
            </a>
        </div>
    </div>
</body>
</html>
```

result.html

```
<!doctype html>
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <title>Handwritten Digit Recognition</title>
    <meta name="description" content="PR & Team built the Handwritten Digit Recognition using Artificial Intelligence which supported by Nalaiya Thiran Initiative">
    <meta name="author" content="PR & Team">
    <meta name="keywords" content="Digit Recognition using Artificial Intelligence">
    <meta name="viewport" content="width=device-width, minimum-scale=1.0">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <link href="https://fonts.googleapis.com/css?family=Open+Sans:300,400,600,700|Varela" rel="stylesheet">
    <link rel="apple-touch-icon" sizes="144x144" href="static/img/apple-touch-icon.png">
    <link rel="icon" type="image/png" sizes="32x32" href="static/img/favicon-32x32.png">
    <link rel="icon" type="image/png" sizes="16x16" href="static/img/favicon-16x16.png">
    <link rel="icon" sizes="16x16" href="assets/img/favicon.ico">
    <link rel="manifest" href="static/img/manifest.json">
    <link rel="mask-icon" href="static/img/safari-pinned-tab.svg"
```

```
color="#5bbad5">
    <meta name="theme-color" content="#ffffff">
    <link rel="stylesheet" href="static/css/bootstrap.min.css" />
    <link rel="stylesheet" href="static/css/font-awesome.min.css" />
    <link rel="stylesheet" href="static/css/style.css">
</head>
<style>
    .center {
        margin-bottom: 25%;
        vertical-align: middle;
        text-align: center;
        /* border: 5px solid green; */
      }
</style>
<body>
    <div class="center">
        <h1> {{ prediction }}</h1>
        <h3>Accuracy: {{Accuracy}}</h3>
    </div>

</body>
</html>
```

upload.html

```html
<!DOCTYPE html>
<html lang="en">
 <head>
   <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <title>Handwritten Digit Recognition</title>
   <meta name="description" content="PR & Team built the Handwritten Digit Recognition using Artificial Intelligence which supported by Nalaiya Thiran Initiative">
   <meta name="author" content="PR & Team">
   <meta name="keywords" content="Digit Recognition using Artificial Intelligence">
   <meta name="viewport" content="width=device-width, minimum-scale=1.0">
   <meta name="apple-mobile-web-app-capable" content="yes">
   <link href="https://fonts.googleapis.com/css?family=Open+Sans:300,400,600,700|Varela" rel="stylesheet">
   <link rel="apple-touch-icon" sizes="144x144" href="static/img/apple-touch-icon.png">
   <link rel="icon" type="image/png" sizes="32x32" href="static/img/favicon-32x32.png">
```

```
    <link rel="icon" type="image/png" sizes="16x16" href="static/img/favicon-16x16.png">
    <link rel="icon" sizes="16x16" href="assets/img/favicon.ico">
    <link rel="manifest" href="static/img/manifest.json">
    <link rel="mask-icon" href="static/img/safari-pinned-tab.svg" color="#5bbad5">
    <meta name="theme-color" content="#ffffff">
</head>
  <body style='align-self: center;'>
    <h1>Upload and Predict</h1>
    <form method="POST" enctype="multipart/form-data" action="/predict" method="POST">
        <input type="file" id="myFile" name="img" accept=".png">
        <input type="submit" value="Predict">
    </form>
  </body>
</html>
```

7.3Database Schema :

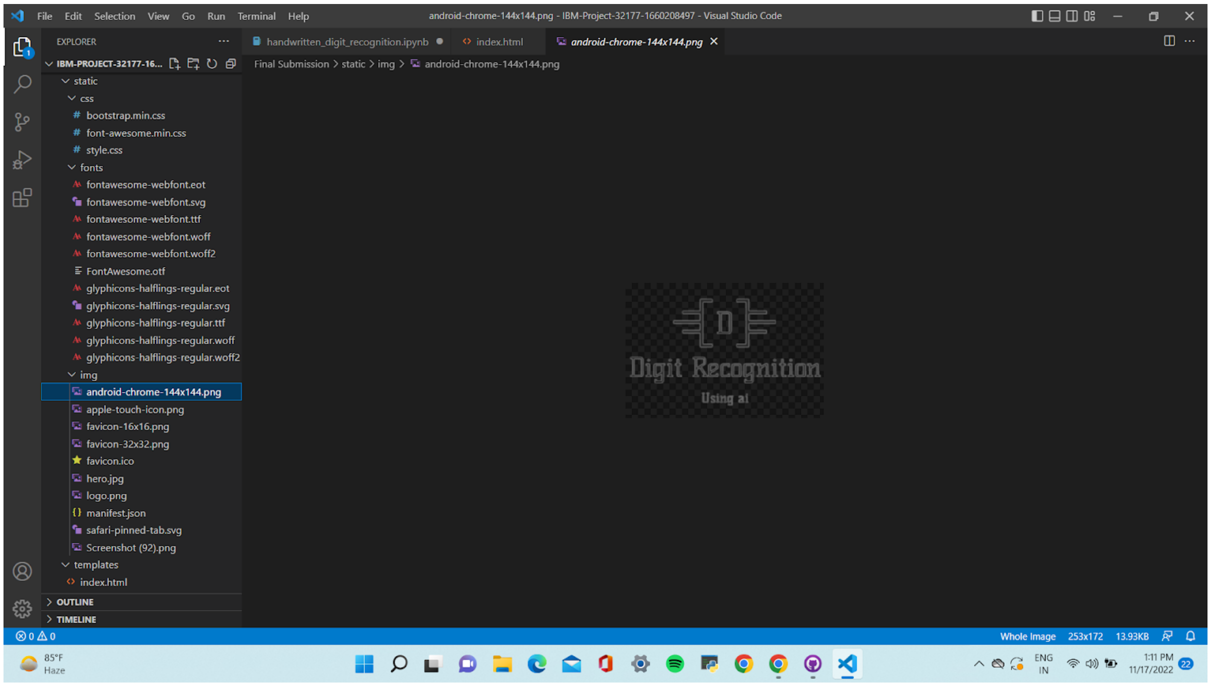
The **MNIST database** (*Modified National Institute of Standards and Technology database*) is a large database of handwritten digits that is commonly used for training various image processing systems. The database is also widely used for training and testing in the field of machine learning. It was created by "re-mixing" the samples from NIST's original datasets.The creators felt that since NIST's training dataset was taken from American Census Bureau employees, while the testing dataset was taken from American high school students, it was not well-suited for machine learning experiments.Furthermore, the black and white images from NIST were normalized to fit into a 28x28 pixel bounding box and anti-aliased, which introduced grayscale levels.

The MNIST database contains 60,000 training images and 10,000 testing images.Half of the training set and half of the test set were taken from NIST's training dataset, while the other half of the training set and the other half of the test set were taken from NIST's testing dataset.The original creators of the database keep a list of some of the methods tested on it. In their original paper, they use a support-vector machine to get an error rate of 0.8%.Extended MNIST (EMNIST) is a newer dataset developed and released by NIST to be the (final) successor to MNIST. MNIST included images only of handwritten digits.

EMNIST includes all the images from NIST Special Database 19, which is a large database of handwritten uppercase and lower case letters as well as digits. The images in EMNIST were converted into the same 28x28 pixel format, by the same process, as were the MNIST images. Accordingly, tools which work with the older, smaller, MNIST dataset will likely work unmodified with EMNIST.

8.TESTING

8.1 Test Cases :

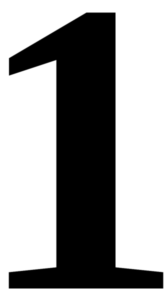


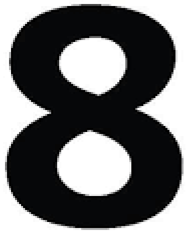
android-chrome-144x144.png - IBM-Project-32177-1660208497 - Visual Studio Code

Final Submission > static > img > android-chrome-144x144.png

```
IBM-PROJECT-32177-16...
  static
    css
      bootstrap.min.css
      font-awesome.min.css
      style.css
    fonts
      fontawesome-webfont.eot
      fontawesome-webfont.svg
      fontawesome-webfont.ttf
      fontawesome-webfont.woff
      fontawesome-webfont.woff2
      FontAwesome.otf
      glyphicons-halflings-regular.eot
      glyphicons-halflings-regular.svg
      glyphicons-halflings-regular.ttf
      glyphicons-halflings-regular.woff
      glyphicons-halflings-regular.woff2
    img
      android-chrome-144x144.png
      apple-touch-icon.png
      favicon-16x16.png
      favicon-32x32.png
      favicon.ico
      hero.jpg
      logo.png
      manifest.json
      safari-pinned-tab.svg
      Screenshot (92).png
  templates
    index.html
```

8.2 User Acceptance Testing :

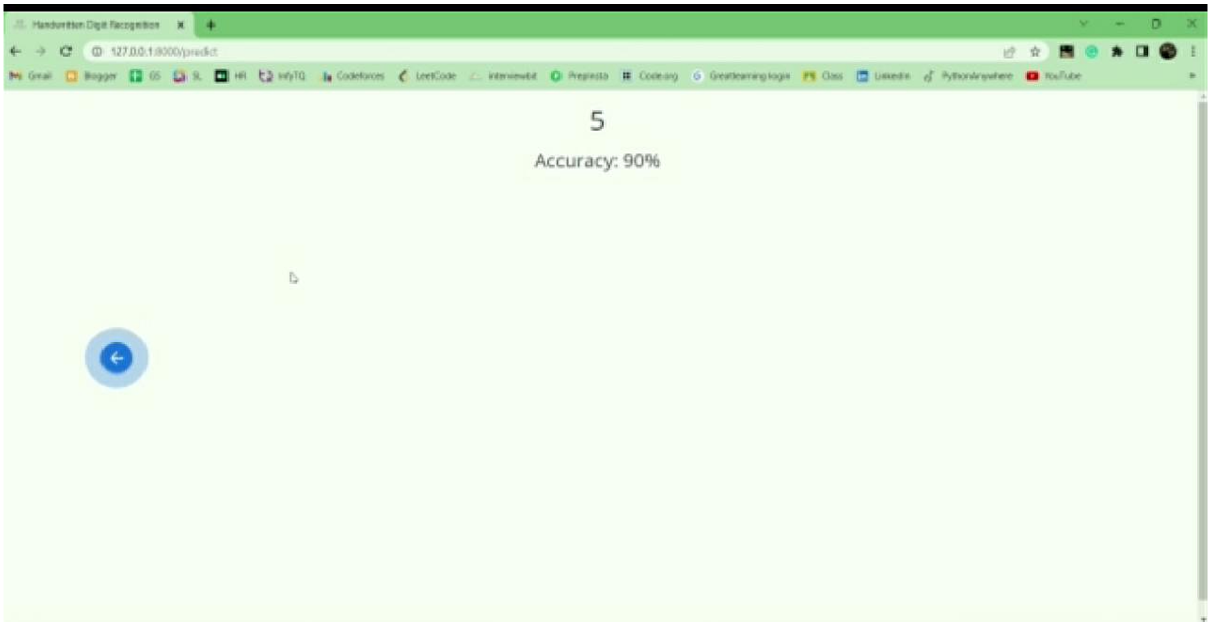




9.RESULTS

9.1 Performance Metrics :

The performance of handwriting recognition systems is typically measured in terms of “recognition rate”. Many academic competitions work this way. However, additional external requirements may shift the view of recognition quality: Processing time and acceptable error rate may be limited, lexica may be missing, but are needed to unambiguously define result correctness. These aspects will be discussed in detail, and appropriate metrics will be proposed. A single-valued combination of these metrics may then be defined for specific application areas. It can be used in order to choose between recognition approaches or systems, and to optimize system parameters automatically.



5

Accuracy: 90%

11.CONCLUSION

The Handwritten Digit Recognition using Deep learning methods has been implemented. The most widely used Machine learning algorithms, KNN, SVM, RFC and CNN have been trained and tested on the same data in order acquire the comparison between the classifiers. Utilising these deep learning techniques, a high amount of accuracy can be obtained. Compared to other research methods, this method focuses on which classifier works better by improving the accuracy of classification models by more than 99%. Using Keras as backend and Tensorflow as the software, a CNN model is able to give accuracy of about 98.72%. In this initial experiment, CNN gives an accuracy of 98.72%, while KNN gives an accuracy of 96.67%, while RFC and SVM are not that outstanding.

12.FUTURE SCOPE

The future development of the applications based on algorithms of deep and machine learning is practically boundless. In the future, we can work on a denser or hybrid algorithm than the current set of algorithms with more manifold data to achieve the solutions to many problems. In future, the application of these algorithms lies from the public to high-level authorities, as from the differentiation of the algorithms above and with future development we can attain high-level functioning applications which can be used in the classified or government agencies as well as for the common people, we can use these algorithms in hospitals application for detailed medical diagnosis, treatment and monitoring the patients, we can use it in surveillances system to keep tracks of the suspicious activity under the system, in fingerprint and retinal scanners, database filtering applications, Equipment checking for national forces and many more problems of both major and minor category. The advancement in this field can help us create an environment of safety, awareness and comfort by using these algorithms in day-to-day application and high-level application (i.e., corporate level or Government level). Application-based on artificial intelligence and deep learning is the future of the technological world because of their absolute accuracy and advantages over many major problems.

13.APPENDIX

Source Code**:**

app.py

```
import imghdr

from flask import Flask, render_template, request

from PIL import Image

import numpy as np

from tensorflow.keras.models import load_model

import tensorflow as tfapp = Flask(__name__)

@app.route('/')

def home():

    return render_template('index.html')

@app.route('/about')

def about():
    return render_template('index.html')
@app.route('/upload')
def upload():
    return render_template('upload.html')
@app.route('/predict',methods=['POST'])
def upload_image_file():
    model = load_model("models/mnistCNN.h5")
    if request.method == 'POST':
        img = Image.open(request.files['img']).convert('L')
        img = img.resize((28,28))
        im2arr = np.array(img)
        im2arr = im2arr.reshape(1,28,28,1)
        # predict = model.predict(im2arr)
        predict = model.predict([im2arr])[0]
        predicted =  np.argmax(predict)
        acc = max(predict)
        print(predicted,acc)

    return
render_template('result.html',prediction=predicted,Accuracy=str(int(acc*100))+'%
')
if __name__ == '__main__':
    app.run(host='127.0.0.1', port=8000, debug=True)
```

Handwritten_digit_recognition.ipynb:

```
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
from tensorflow.keras.layers import Conv2D
from keras.optimizers import Adam
from keras.utils import np_utils
#loading data
(X_train, Y_train), (X_test, Y_test) = mnist.load_data()
print(X_train.shape)
print(X_test.shape)
X_train[0]
Y_train[0]
import matplotlib.pyplot as plt
plt.imshow(X_train[0])
#reshaping dataset
X_train = X_train.reshape(60000,28,28,1).astype('float32')
X_test = X_test.reshape(10000,28,28,1).astype('float32')
```

```
#onehot encoding
number_of_classes = 10
Y_train = np_utils.to_categorical(Y_train,number_of_classes)
Y_test = np_utils.to_categorical(Y_test,number_of_classes)
Y_train[0]
#Creating Model
model = Sequential()
model.add(Conv2D(64,(3,3),input_shape=(28,28,1),activation='relu'))
model.add(Conv2D(32,(3,3),activation='relu'))
model.add(Flatten())
model.add(Dense(number_of_classes,activation='softmax'))
model.compile(loss='categorical_crossentropy',optimizer='Adam',metrics=['accuracy'])
#Model fit
model.fit(X_train,Y_train,
validation_data=(X_test,Y_test),epochs=5,batch_size=32)
#Observing metrics
metrics = model.evaluate(X_test,Y_test,verbose=0)
print("Accuracy : ",metrics)
#Predicting Output
prediction = model.predict(X_test[:4])
print("Prediction : ",prediction)
print(np.argmax(prediction,axis=1))
print(Y_test[:4])
#Observing Metrics
metrics = model.evaluate(X_test,Y_test,verbose=0)
print("Metrics : ",metrics)
from tensorflow.keras.models import load_model
from PIL import Image
import numpy as np
model = load_model("models/mnistCNN.h5")
filename = 'num'
img = Image.open(filename +'.png').convert("L")
img = img.resize((28,28))
im2arr = np.array(img)
im2arr = im2arr.reshape(1,28,28,1)
Y_pred = model.predict(im2arr)
print(Y_pred)
```

templates

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <title>Handwritten Digit Recognition</title>
    <meta name="description" content="PR & Team built the Handwritten Digit Recognition using Artificial Intelligence which supported by Nalaiya Thiran Initiative">
    <meta name="author" content="PR & Team">
    <meta name="keywords" content="Digit Recognition using Artificial Intelligence">
    <meta name="viewport" content="width=device-width, minimum-scale=1.0">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <link href="https://fonts.googleapis.com/css?family=Open+Sans:300,400,600,700|Varela" rel="stylesheet">
    <link rel="apple-touch-icon" sizes="144x144" href="static/img/apple-touch-icon.png">
    <link rel="icon" type="image/png" sizes="32x32" href="static/img/favicon-32x32.png">
    <link rel="icon" type="image/png" sizes="16x16" href="static/img/favicon-16x16.png">
    <link rel="icon" sizes="16x16" href="assets/img/favicon.ico">
    <link rel="manifest" href="static/img/manifest.json">
    <link rel="mask-icon" href="static/img/safari-pinned-tab.svg"
```

```
color="#5bbad5">
    <meta name="theme-color" content="#ffffff">
    <link rel="stylesheet" href="static/css/bootstrap.min.css" />
    <link rel="stylesheet" href="static/css/font-awesome.min.css" />
    <link rel="stylesheet" href="static/css/style.css">
</head>
<body>
    <div id="top" class="hero background-overlay">
        <div class="hero-content">
            <h1>A Novel Method for Handwritten Digit Recognition System</h1>
            <p class="hero-job"><span>Using AI</span></p>
            <p class="hero-job-desc"><a style="color:white;" href"#">PR &amp;
Team</a> </p>
        </div>
        <div class="hero-arrow page-scroll home-arrow-down">
            <a class="" href="upload">
                <i class="fa fa-angle-double-down" aria-hidden="true"></i>
            </a>
        </div>
    </div>
</body>
</html>
```

result.html

```
<!doctype html>
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <title>Handwritten Digit Recognition</title>
    <meta name="description" content="PR & Team built the Handwritten Digit Recognition using Artificial Intelligence which supported by Nalaiya Thiran Initiative">
    <meta name="author" content="PR & Team">
    <meta name="keywords" content="Digit Recognition using Artificial Intelligence">
    <meta name="viewport" content="width=device-width, minimum-scale=1.0">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <link href="https://fonts.googleapis.com/css?family=Open+Sans:300,400,600,700|Varela" rel="stylesheet">
    <link rel="apple-touch-icon" sizes="144x144" href="static/img/apple-touch-icon.png">
    <link rel="icon" type="image/png" sizes="32x32" href="static/img/favicon-32x32.png">
    <link rel="icon" type="image/png" sizes="16x16" href="static/img/favicon-16x16.png">
    <link rel="icon" sizes="16x16" href="assets/img/favicon.ico">
    <link rel="manifest" href="static/img/manifest.json">
    <link rel="mask-icon" href="static/img/safari-pinned-tab.svg" color="#5bbad5">
    <meta name="theme-color" content="#ffffff">
    <link rel="stylesheet" href="static/css/bootstrap.min.css" />
    <link rel="stylesheet" href="static/css/font-awesome.min.css" />
    <link rel="stylesheet" href="static/css/style.css">
</head>
<style>
    .center {
        margin-bottom: 25%;
        vertical-align: middle;
        text-align: center;
        /* border: 5px solid green; */
      }
</style>
<body>
    <div class="center">
        <h1> {{ prediction }}</h1>
    <h3>Accuracy: {{Accuracy}}</h3>
  </div>


</body>
</html>
```

upload.html

```
<!DOCTYPE html>
<html lang="en">
 <head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <title>Handwritten Digit Recognition</title>
  <meta name="description" content="PR & Team built the Handwritten Digit
Recognition using Artificial Intelligence which supported by Nalaiya Thiran
Initiative">
  <meta name="author" content="PR & Team">
  <meta name="keywords" content="Digit Recognition using Artificial
Intelligence">
  <meta name="viewport" content="width=device-width, minimum-scale=1.0">
  <meta name="apple-mobile-web-app-capable" content="yes">
  <link
href="https://fonts.googleapis.com/css?family=Open+Sans:300,400,600,700|Varel
```

```
a" rel="stylesheet">
    <link rel="apple-touch-icon" sizes="144x144" href="static/img/apple-touch-icon.png">
    <link rel="icon" type="image/png" sizes="32x32" href="static/img/favicon-32x32.png">
    <link rel="icon" type="image/png" sizes="16x16" href="static/img/favicon-16x16.png">
    <link rel="icon" sizes="16x16" href="assets/img/favicon.ico">
    <link rel="manifest" href="static/img/manifest.json">
    <link rel="mask-icon" href="static/img/safari-pinned-tab.svg" color="#5bbad5">
    <meta name="theme-color" content="#ffffff">
</head>
  <body style='align-self: center;'>
    <h1>Upload and Predict</h1>
    <form method="POST" enctype="multipart/form-data" action="/predict" method="POST">
        <input type="file" id="myFile" name="img" accept=".png">
        <input type="submit" value="Predict">
    </form>
  </body>
</html>
```

GitHub & Project Demo Link

GitHub Link:
https://github.com/IBM-EPBL/IBM-Project-22985-1659863822

Project Demo Link:
https://drive.google.com/file/d/1R35h6qb5wGQUNBKxutSZvsQRwX9lDodf/view?usp=share_link