A Novel Method for Handwritten Digit RecognitionSystem

A PROJECT REPORT

Submitted by

K. BOOMIKA	**61771931009**
H. DHEEPTHIKA	**61771931013**
M. KABINI	**61771931022**
K. MOULISHWARAN	**61771931035**

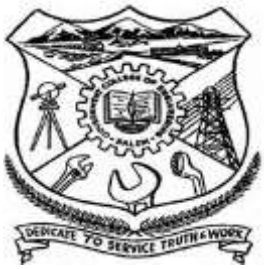
TEAM ID: PNT2022TMID06779

In partial fulfillment for the award of the certificationof

IBM PROJECT

in

Artificial Intelligence Domain



GOVERNMENT COLLEGE OF ENGINEERING, SALEM.
DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING, SALEM 636011.

NOV-2022.

CONTENT

ACKNOWLEDGMENT

An endeavour over a long period can be successful with the advice and support of many well-wishers. We take this opportunity to express our gratitude and appreciation to all of them. We owe our tributes to **Prof.P. Deepti**, Assistant Professor/Electronics and Communication Engineering, Goverment College of Engineering, Salem, for his valuable support and guidance during the period of project implementation. We wish to express our sincere thanks and gratitude to our project guide **Prof.K. Jayanthi**, Electronics and Communication Engineering, Salem for the simulating discussions, in analysing problems associated with our project work and for guiding us throughout the project. Project meetings were highly informative. We express our warm and sincere thanks for the encouragement, untiring guidance and the confidence they had shown in us. We are immensely indebted for their valuable guidance throughout our project. We also thank all the staff members of ECE department for their valuable advices. We also thank supporting staff for providing resources as and when required.

K. BOOMIKA	61771931009
M. KABINI	61771931022
H. DHEEPTHIKA	61771931013
K. MOULISHWARAN	61771931035

1. INTRODUCTION

1.1 Project Overview

Machine learning and deep learning plays an important role in computer technology and artificial intelligence. With the use of deep learning and machine learning, human effort can be reduced in recognizing, learning, predictions and many more areas. This article presents recognizing the handwritten digits (0 to 9) from the famous MNIST dataset, comparing classifiers like KNN, PSVM, NN and convolution neural network on basis of performance, accuracy, time, sensitivity, positive productivity, and specificity with using different parameters with the classifiers. To make machines more intelligent, the developers are diving into machine learning and deep learning techniques. A human learns to perform a task by practicing and repeating it again and again so that it memorizes how to perform the tasks. Then the neurons in his brain automatically trigger and they can quickly perform the task they have learned. Deep learning is also very similar to this. It uses different types of neural network architectures for different types of problems.

For example – object recognition, image and sound classification, object detection, image segmentation, etc. The 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 handwritten digit recognition is the solution to this problem which uses the image of a digit and recognizes the digit present in the image.

1.2 Purpose of Digit Recognition System:

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, numeric entries in forms filled up by hand and so on.

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.

2. LITERATURE SURVEY

2.1 Existing Problem

An early notable attempt in the area of character recognition research is by Grimsdale in 1959. The origin of a great deal of research work in the early sixties was based on an approach known as analysisby-synthesis method suggested by Eden in 1968. The great importance of Eden's work was that he formally proved that all handwritten characters are formed by a finite number of schematic features, a point that was implicitly included in previous works.

2.2 Reference

1. **Ahlawat Savita, Amit Choudhary,** the aim of this paper is to develop a hybrid model of a powerful Convolutional Neural Networks (CNN) and Support Vector Machine (SVM) for recognitionof handwritten digit from MNIST dataset. The proposed hybrid model combines the key properties of both the classifiers. In the proposed hybrid model, CNN works as an automatic feature extractorand SVM works as a binary classifier. The MNIST dataset of handwritten digits is used for training and testing the algorithm adopted in the proposed model. The MNIST dataset consists of handwritten digits images which are diverse and highly distorted. The receptive field of CNN helps in automatically extracting the most distinguishable features from these handwritten digits. The experimental results demonstrate the effectiveness of the proposed framework by achieving a recognition accuracy of 99.28% over MNIST handwritten digits dataset.

2. **Ali Abdullah Yahya, Min Hu,** an enormous number of CNN classification algorithms have been proposed in the literature. Nevertheless, in these algorithms, appropriate filter size selection, data preparation, limitations in datasets, and noise have not been taken into consideration. As a consequence, most of the algorithms have failed to make a noticeable improvement in classification accuracy. To address the shortcomings of these algorithms, our paper presents the following contributions: Firstly, after taking the domain knowledge into consideration, the size ofthe effective receptive field (ERF) is calculated. Calculating the size of the ERF helps us to select a typical filter size which leads to enhancing the classification accuracy of our CNN. our CNN algorithm achieves state-of-the-art results in handwritten digit recognition, with a recognition accuracy of 99.98%, and 99.40% with 50% noise.

2.3 Problem Statement Definition

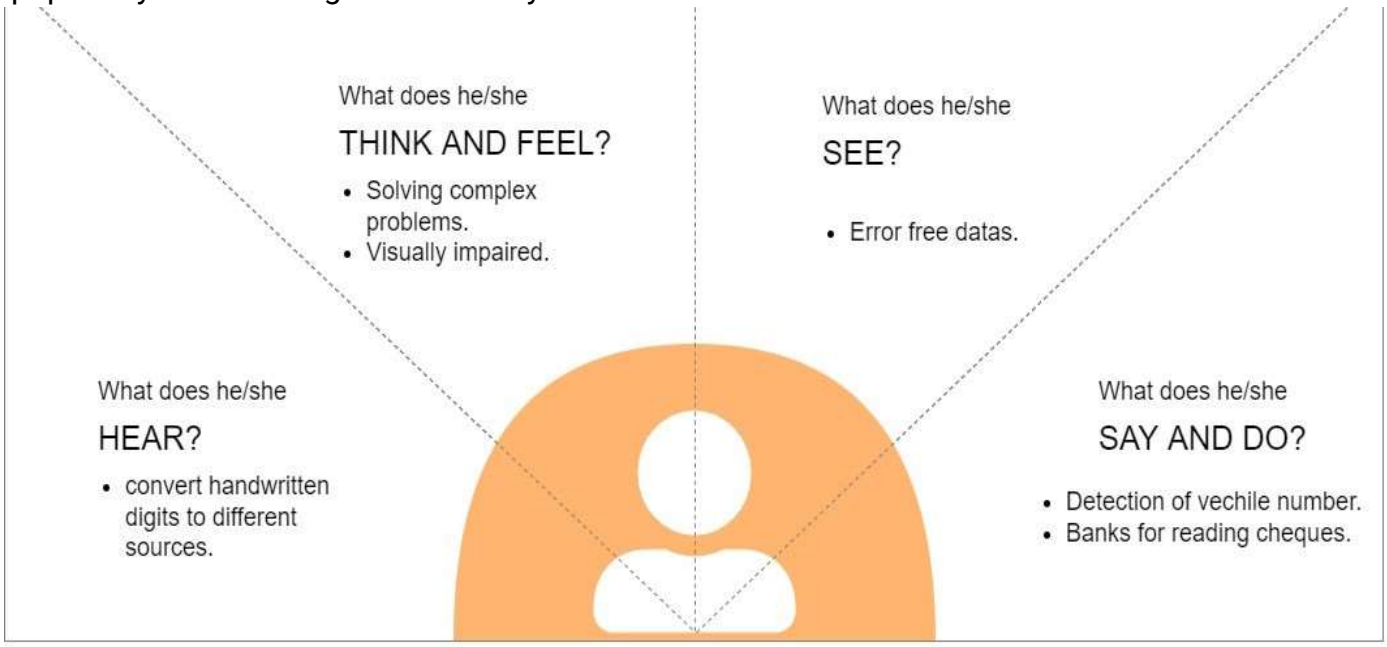
The goal of this project is to create a model that will be able to recognize and determine the handwritten digits from its image by using the concepts of Convolution Neural Network. Though the goal is to create a model which can recognize the digits, it can be extended to letters and an individual's handwriting. The major goal of the proposed system is understanding Convolutional Neural Network, and applying it to the handwritten recognition system.

The handwritten digit recognition is the capability of computer applications to recognize the human handwritten digits. It is a hard task for the machine because handwritten digits are not perfect and can be made with many different shapes and sizes. The handwritten digit recognition system is a way to tackle this problem which uses the image of a digit and recognizes the digit present in the image. Convolutional Neural Network model created using PyTorch library over the MNIST dataset to recognize handwritten digits. Handwritten Digit Recognition is the capability of a computer to fete the mortal handwritten integers from different sources like images, papers, touch defenses, etc, and classify. Them into predefined classes. 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 peoples as it. is not an Optic 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 base of their delicacy, crimes, and testing- training time corroborated by plots and maps that have been constructed using matplotlib for visualization.

3.IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

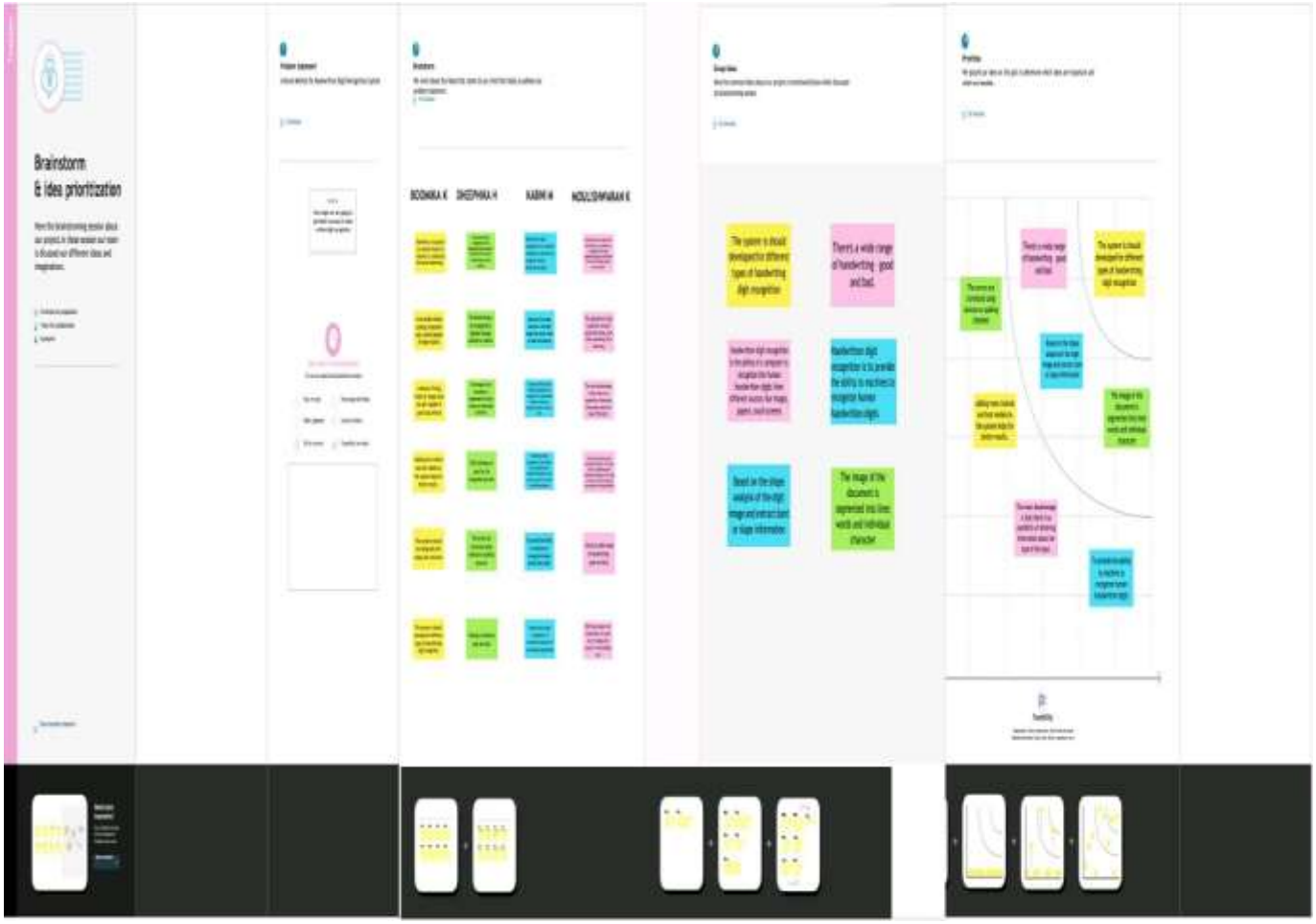
An empathy map is a **collaborative tool teams can use to gain a deeper insight into their customers.** Much like a user persona, an empathy map can represent a group of users, such as a customer segment. The empathy map was originally created by Dave Gray and has gained much popularity within the agile community.



PAIN	GAIN
• Font styles are more to recognize. • Similarities between digits.	• Banking and postal operations easier and error free. • Gain accuracy.

3.2 Ideation & Brainstorming

Ideation refers to the whole creative process of coming up with and communicating new ideas. Brainstorming is a **group problem-solving method that involves the spontaneous contribution** of creative ideas and solutions. This technique requires intensive, freewheeling discussion in which every member of the group is encouraged to think aloud and suggest as many ideas as possible based on their diverse knowledge.



3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	To recognize the handwritten digits from a wide variety of sources like emails, papers, images, letters etc. It can solve more complex problems and makes humans' jobs easier.
2.	Idea / Solution description	This system is built by using (CNN) Convolutional Neural Network. By using this system, we can capture the image of handwritten digits and can predict the digits.
3.	Novelty / Uniqueness	Here users can upload the handwritten digits from anywhere and it gives accurate results.
4.	Social Impact / Customer Satisfaction	The feasibility of implementing this idea is moderate neither easy nor tough because the system needs to satisfy the basic requirements of the customer and should give accurate results.
5.	Business Model (Revenue Model)	BY using this website, the users can predict and analyze the handwritten digits of the user. The website can be developed at minimum cost with high performance and interactive user interface.
6.	Scalability of the Solution	The solution can be made scalable. The people can easily understand the digits. This system can also be integrated with future technologies

3.4.Problem Solution Fit

Define CS, fit into CC

1. CUSTOMER SEGMENT(S)

- Customers are those who work with handwritten numbers in places like banks, schools, colleges, railroads, etc.

6. CUSTOMER CONSTRAINTS

- Lack of reliable internet connections, unavailability of gadgets like mobile phones and computers, inaccessibility of appropriate cameras.
- Because handwritten numbers are not always accurate and might have a wide variety of tastes, it is a difficult work for the computer.
- This issue can be solved by using an image of a digit to identify the digit that is present in the image, which is done through handwritten digit recognition.

5. AVAILABLE SOLUTIONS

- Although there are current alternatives to this approach, they are not very precise, robust, or rotation- and variation-invariant.
- The ability of a computer to honor the mortal handwritten characters from many sources, including as photographs, papers, and touch input.

Explore AS, differentiate

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

- It is really challenging to comprehend and analyze the handwritten numbers.
- More training data required.
- Hard to recognize digits, dim lighting, weak eyesight.

9. PROBLEM ROOT CAUSE

- Hand-written digits are in varying fonts and sizes; thus, they are becoming increasingly difficult to ascertain due to various factors such as weakening eyesight, time constraints, etc.

7. BEHAVIOUR

- Finding the best software that more quickly and accurately identifies digits.
- Customer wants reliable internet connections and high-quality cameras.

Focus on J&P, tap into BE, understand

SOLUTION ARCHITECTURE

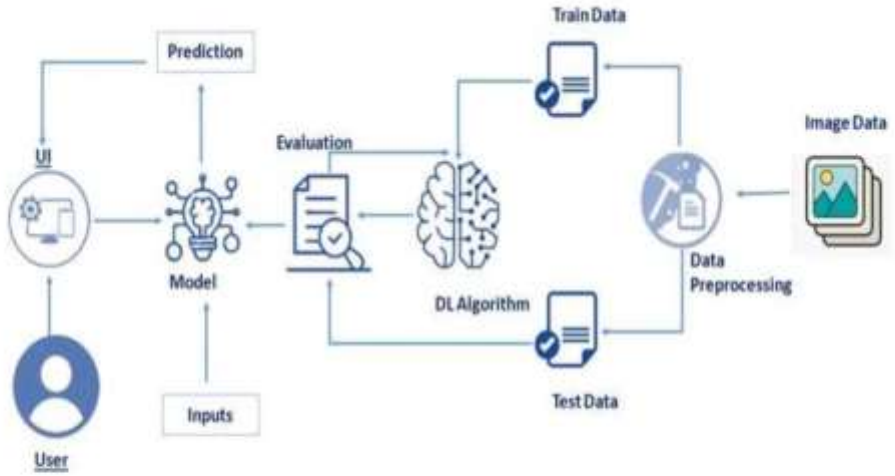


Figure 1: Architecture and data flow of the Handwritten Digit Recognition application

4.Requirements Analysis

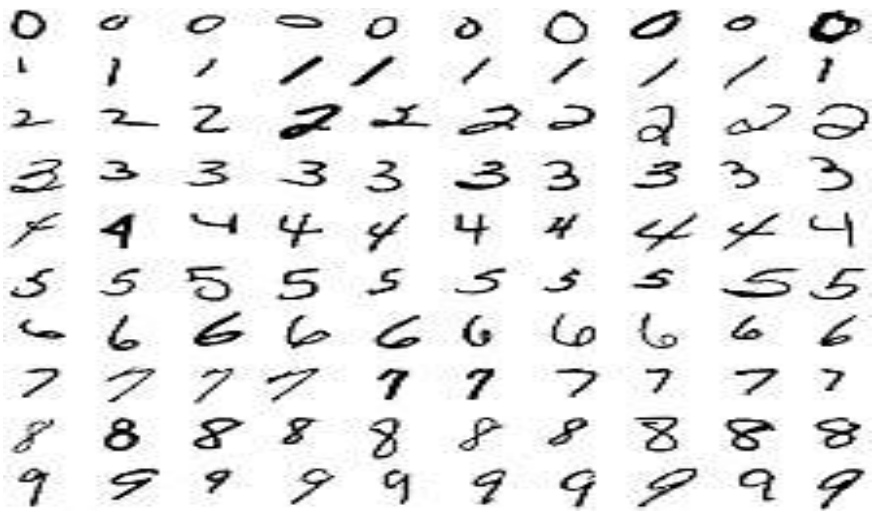
4.1 Functional Requirements:

1. **Image Data:** The 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.

2. **Hosting** (also known as Web site hosting, Web hosting, and Webhosting) is the business of **housing, serving, and maintaining files for one or more Web sites**. More important than the computer space that is provided for Web site files is the fast connection to the Internet.

3. **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.

4. **MNIST Dataset:** The MNIST dataset is an acronym that stands for the **Modified National Institute Standards and Technology dataset.** It is a dataset of 60,000 small square 28×28-pixel grayscale images of handwritten single digits between 0 and 9.



5.Cloud: The cloud provides a number of IT services such as servers,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

Usability:

Handwritten character recognition is one of the practically important issues in pattern recognition applications. The applications of digit recognition include in postal mail sorting, bank check processing, form data entry, etc.

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.

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.

Accuracy:

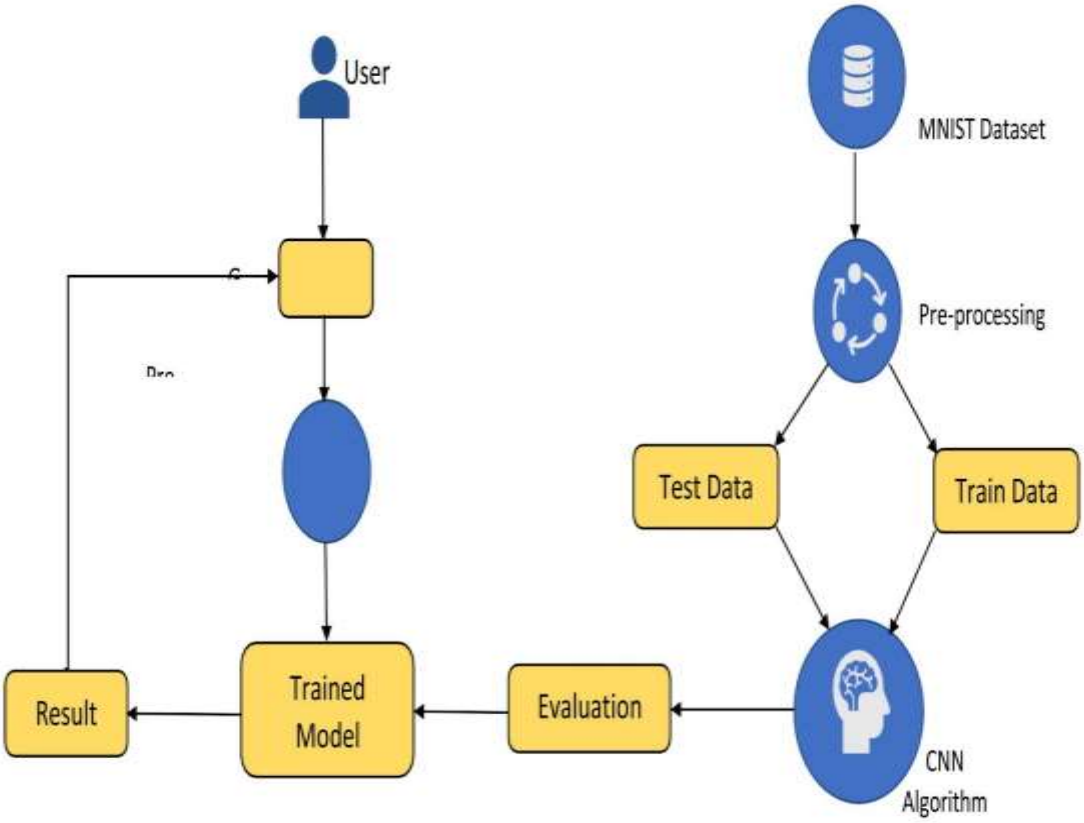
MNIST - Classification of digits (accuracy=98%).

The MNIST dataset is an image dataset of handwritten digits. It has has 60,000 training images and 10,000 test images, each of which are grayscale 28 x 28 sized images.

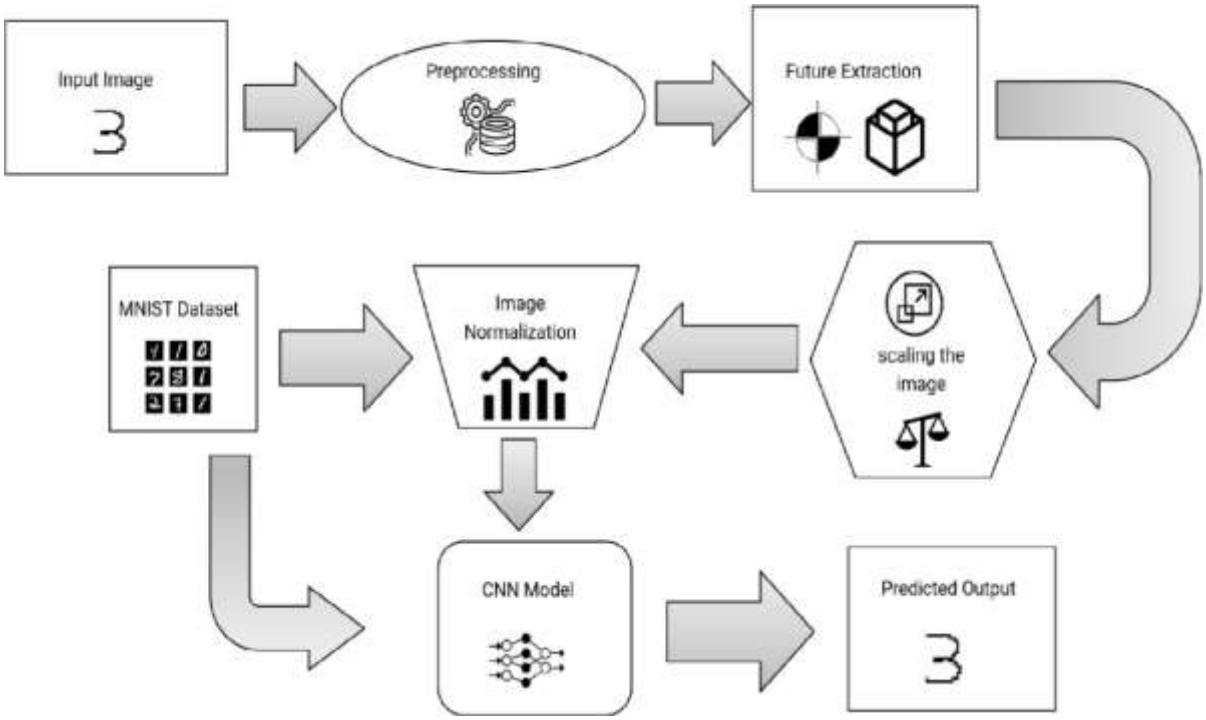
5. PROJECT DESIGN

5.1 Data Flow Diagram

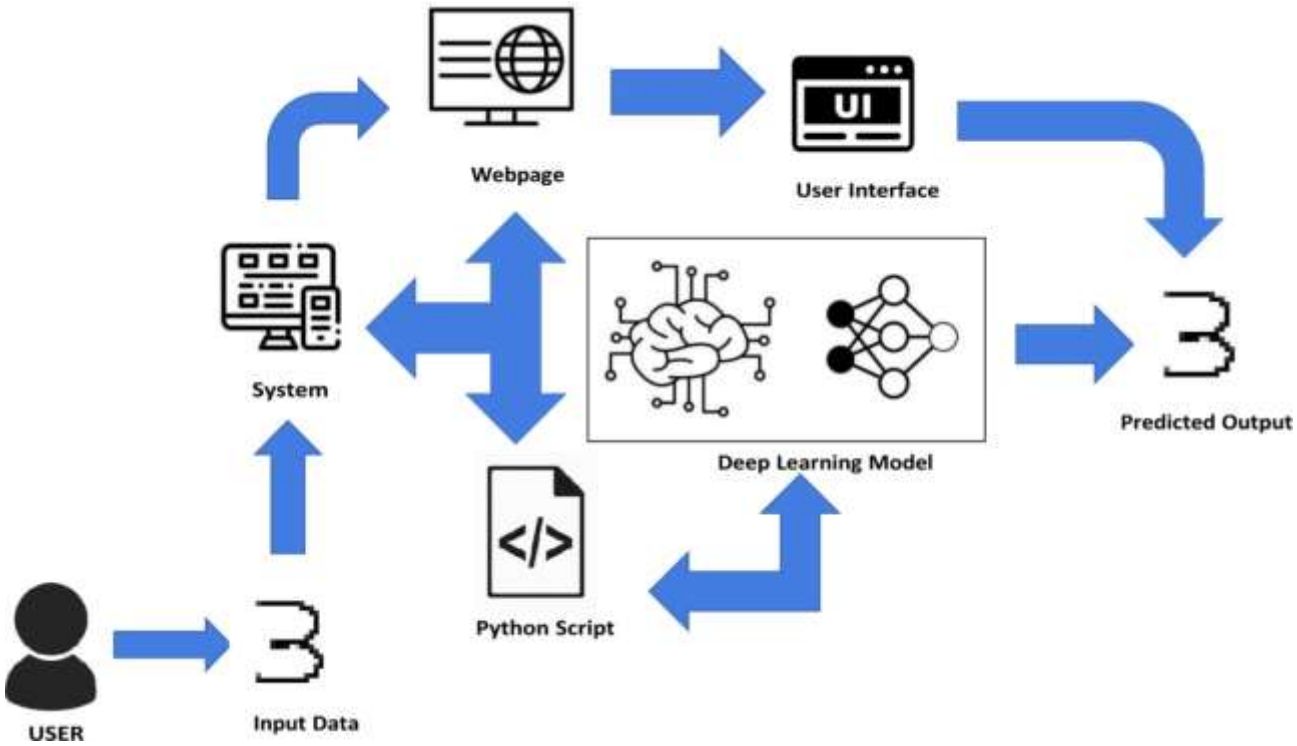
Data Flow Diagrams:



5.2 Solution & Technical Architecture



Interactive Technology Architecture



5.3 User Stories

1. Custome

i.Home

User 1. As a user, I can view the guide and awareness to use this application.
Acceptence Criteria - I can view the awareness to use this application and its limitations.

User 2. As a user, I'm allowed to view the guided video to use the interface of this application.
Acceptance Criteria - I can gain knowledge to use this application by a practical method.

User 3. As a user, I can read the instructions to use this application.
Acceptance Criteria - can read instructions also to use it in a user-friendly method.

ii.Home

User 4. As a user, in this prediction page I get to choose the image.
Acceptance Criteria - I can choose the image from our local system and predict the output.

ii. Predict

User 5. As a user, I'm Allowed to upload and choose the image to be uploaded.
Acceptance Criteria - I can upload and choose the image from the system storage and also in any virtual storage.

User 6. As a user, I will train and test the input to get the maximum accuracy of output.
Acceptance Criteria - I can able to train and test the application until it gets maximum accuracy of the result.

User 7. As a user, I can access the MNIST data set.
Acceptance Criteria - I can access the MNIST data set to produce the accurate result.

User 8. As a user, I can view the guide to use the web application.
Acceptance Criteria - I can view the awareness of this application and its limitations.

6. Project Planning & Scheduling

6.1 Sprint Planning & Estimation

S. No	Milestones	Activities	Timeline
1.	Empathy map	Prepared empathy map canvas to capture the user gains and pains.	10 September 2022
2.	Literature survey	Literature survey on the handwritten digit recognition system and information gathering.	20 September 2022
3.	Ideation Phase	Ideas are listed and top 3 ideas are prioritized based on the feasibility and importance.	22 September 2022
4.	Proposed Solution	Proposed solution document is prepared which includes Novelty, feasibility of idea, social impact, scalability of solution, etc.	26 September 2022
5.	Problem Solution fit	Includes customer segments and customer constraints, the problem root cause and jobs to be done.	5 October 2022
6.	Solution architecture	From data collection to digit recognition by the web application are represented in architectural diagram.	5 October 2022
7.	Customer Journey	Prepare Customer Journey maps to understand user interactions and experiences with the application	10 October 2022
8.	Functional Requirements	Functional requirements and non functional requirements alike scalability and accuracy are described.	18 October 2022

9.	Data flow diagram and user stories	Data flow diagram and user stories are prepared and four sprint phases are described.	18 October 2022
10.	Technology Architecture	Technical flow graphs are created and the functions of technical stacks are defined.	19 October 2022

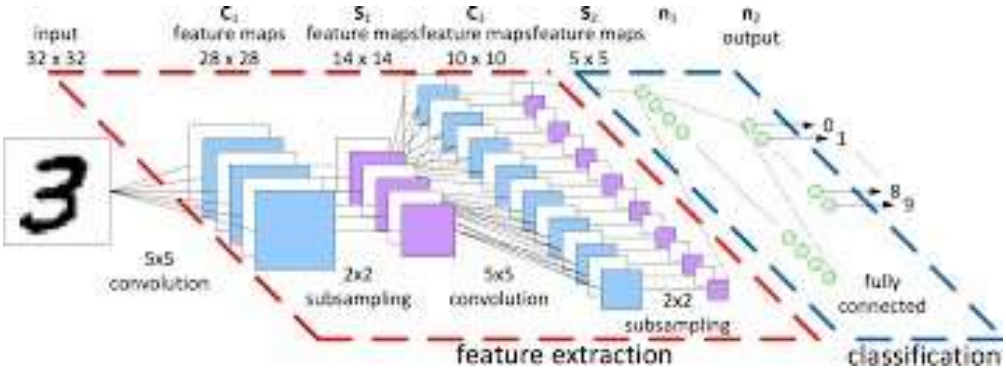
6.2 Sprint Delivery Schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Dashboard	USN-1	As a user, they can see the information regarding the prediction of handwritten digit recognition.	2	High	Boomika K Dheepthika H Kabini M Moulishwaran K
Sprint-1	Launch	USN-2	On clicking the launch button, it will redirect the user to a page where the images to be predicted can be uploaded.	2	High	Boomika K Dheepthika H Kabini M Moulishwaran K
Sprint-2	Upload	USN-3	Users can select the image from the local storage.	2	High	Boomika K Dheepthika H
Sprint-3	Predict	USN-4	Once the image is uploaded, it will predict the respective image.	2	High	Kabini M Moulishwaran K
Sprint-4	Display	USN-5	The predicted image will be displayed with the accuracy chart.	2	High	Boomika K Dheepthika H Kabini M Moulishwaran K

7.CODING & SOLUTIONING

7.1 Feature [1]

i). Using CNN Model in our Project: CNN is basically a model known to be **Convolutional Neural Network** and in recent times it has gained a lot of popularity because of its usefulness. CNN uses multilayer perceptrons to do computational works.
ii). CNN uses relatively little pre-processing compared to other image classification algorithms. This means the network learns through filters that in traditional algorithms were hand-engineered. So, for the image processing tasks CNNs are the best-suited option.



7.2 Feature [2]

ii). Using Flask application in our Project: Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools.

8.TESTING

8.1 Test Cases

Testing is defined as an activity to check whether the actual results match the expected results and to ensure that the software system is defect free. It involves the execution of a software component or system component to evaluate one or more properties of interest. Software testing also helps to identify errors, gaps, or missing requirements in contrary to the actual requirements.

i). Unit Testing:

When the testing happens for some individual group or some related units then that type of testing is called as Unit Testing. It is often done by programmer to test the part of the program he or she has implemented.

Unit Testing is successful means all the modules has been successfully tested and it can proceed further.

ii). Functional Testing:

This type of testing is tested because to check the functional components or the functionality required from the system is gained or not. It actually falls under the testing of the Black Box testing of Software Engineering. This part includes the feeding of the inputs in the system or the project and to check if that system or the project is getting the same value or not as expected if not then calculate the error as wanted and check for more. Functional Testing of this project mainly involves below things. All of these are tested successfully and errors are also calculated.

i) Verifying the input image.
ii) Verifying the work flow.
iii) Correct recognition and calculate the error.

iii). Integration Testing:

In a total project or the system, many groups of components are getting added or summed up in the purpose of the project query. Integration testing is about to check interaction between various modules of the project or the system. This module also includes the hardware and the software requirements of the project.

All the individual modules are integrated and tested together. All the best and extreme cases that the modules are interacting or not are successfully checked and passed, errors are calculated for the deep learning platforms.

iv). System Testing:

This type of testing is actually meant for the system or the project and also the platform and the integrated software's and tools, technologies are also tested. The idea or purpose behind the system testing is to check all the requirements that will be provided by the system.

This application of the project along with the tools and technologies has been tested in both windows and Linux. It passed successfully.

TEST CASE ANALYSIS

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

USER ACCEPTANCE TESTING REPORT

PURPOSE OF THE 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).

DEFECT ANALYSIS

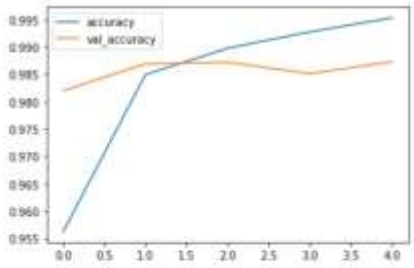
Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Total
By Design	1	0	1	0	2
Duplicate	0	0	0	0	0
External	0	0	2	0	2
Fixed	4	1	0	1	6
Not Reproduced	0	0	0	1	1
Skipped	0	0	0	1	1
Won't Fix	1	0	1	0	2
Total	6	1	4	3	14

9.RESULTS

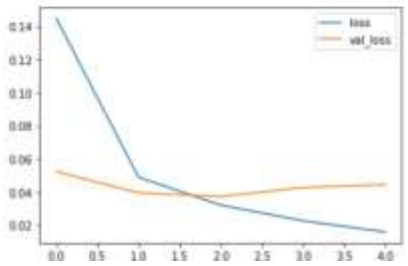
9.1 Performance Metrics:

i). Model Metrics: Our model performs 98% of accuracy when train and testing session.

```
losses[['accuracy','val_accuracy']].plot()
```




```
losses[['loss','val_loss']].plot()
```




```
print(model.metrics_names)
print(model.evaluate(x_test,y_cat_test,verbose=0))
```

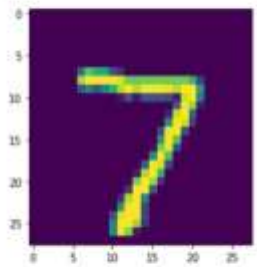
```
['loss', 'accuracy']
[0.044522497802972794, 0.9872999787330627]
```

ii). Overall Application Performance

Predicting a given image

```
my_number = x_test[0]
```

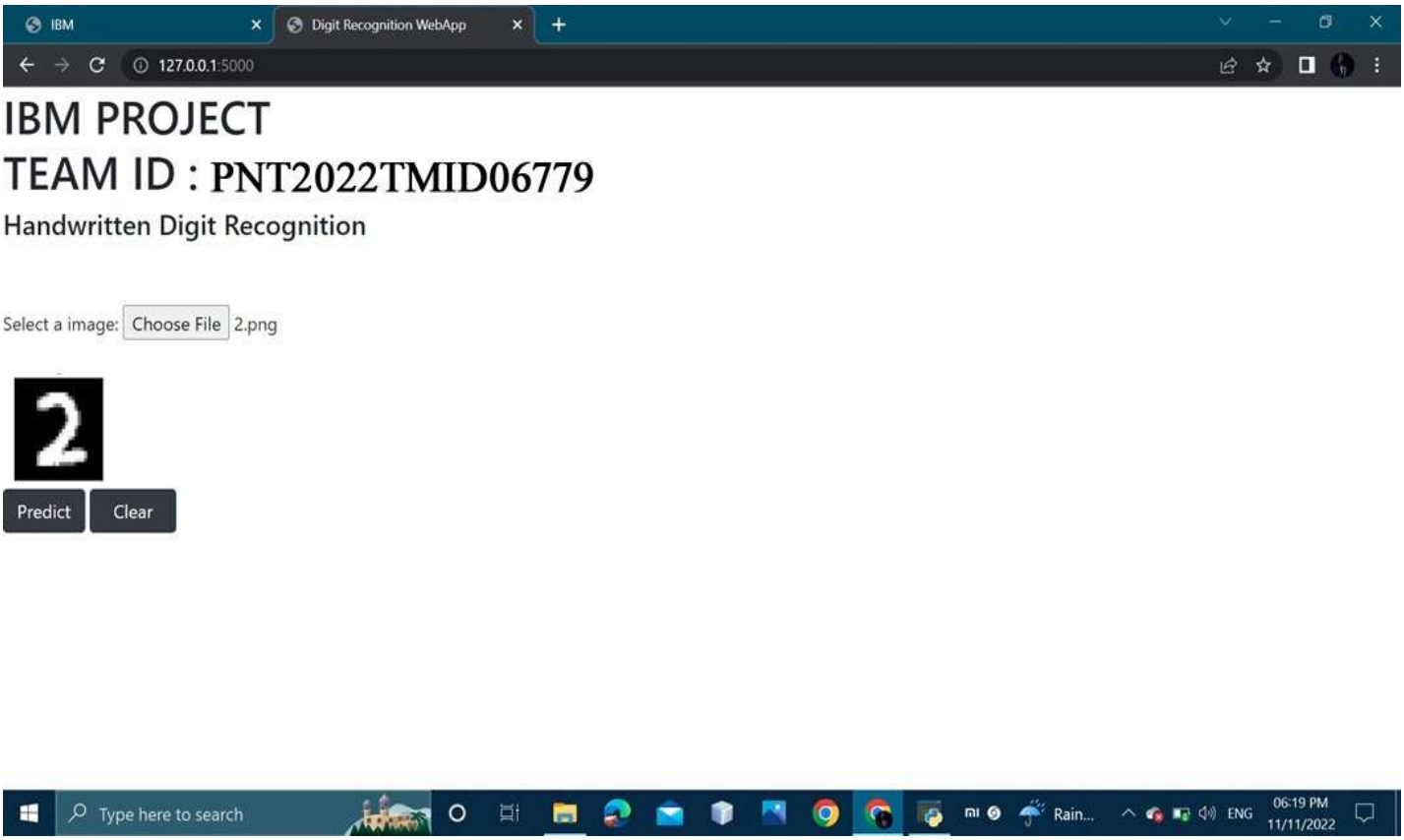
```
plt.imshow(my_number.reshape(28,28))
```



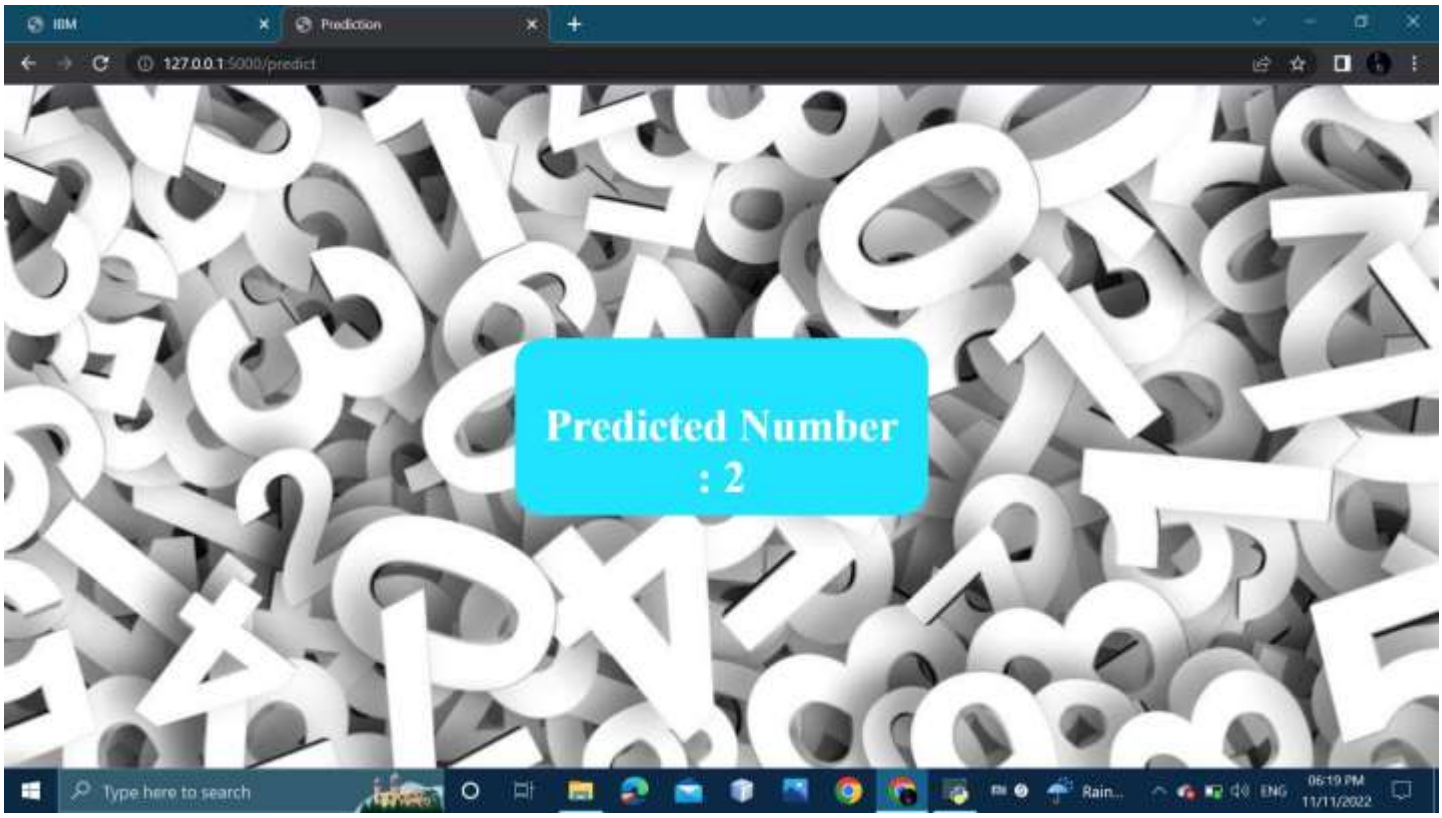
```
# SHAPE --> (num_images,width,height,color_channels)
model.predict(my_number.reshape(1,28,28,1))
```

```
1/1 [==============================] - 0s 42ms/step
array([[1.1048161e-12, 2.5010433e-10, 8.5665883e-09, 6.9659154e-07,
        2.0384635e-12, 6.1936345e-11, 3.0275941e-15, 9.9999928e-01,
        2.5742997e-10, 1.9984461e-09]], dtype=float32)
```

i). Index of the Application



ii). Prediction Page



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

- The Handwritten Digit Recognition using Deep learning methods has been implemented. CNN have been trained and tested on the same data in order acquire the comparison between the classifiers. Utilizing 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%.

12.FUTURE WORK

The proposed system takes 28x28 pixel sized images as input. The same system with further modifications and improvements in the dataset and the model can be used to build Handwritten Character Recognition System which recognizes human handwritten characters and predicts the output.

13 APPENDIX

SOURCE CODE

MODEL CREATION

Importing the required libraries

```python
import numpy
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
```

Python

Loading the data

```python
(X_train,y_train),(X_test,y_test)=mnist.load_data()
```

Python

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz
11490434/11490434 [==============================] - 0s 0us/step
```

Analysing the data

```python
X_train[0]
```

Python

```
Output exceeds the size limit. Open the full output data in a text editor
array([[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   3,
         18,  18,  18, 126, 136, 175,  26, 166, 255, 247, 127,   0,   0,
          0,   0],
```

```
        0,   0]], dtype=uint8)
```

```
y_train[0]
```

Python

```
5
```

```
import matplotlib.pyplot as plt
plt.imshow(X_train[0])
```

Python

```
<matplotlib.image.AxesImage at 0x7fbe944b9150>
```

Reshaping the data

```
X_train=X_train.reshape(60000,28,28,1).astype('float32')
X_test=X_test.reshape(10000,28,28,1).astype('float32')
```

Python

Applying one hot 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)
```

Python

```
y_train[0]
```

Python

```
array([0., 0., 0., 0., 0., 1., 0., 0., 0., 0.], dtype=float32)
```

Adding CNN layers

```
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'))
```

Python

Compiling the model

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

Python

Train the model

```
model.fit(X_train,y_train,validation_data=(X_test,y_test),epochs=5,batch_size=32)
```

Python

```
Epoch 1/5
1875/1875 [==============================] - 195s 103ms/step - loss: 0.2126 - accuracy: 0.9543 - val_loss: 0.0867 - val_accuracy: 0.9730
```

Observing the metrics

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

Python

```
Metrics(Test loss & Test Accuracy):
[0.11479571041059494, 0.9763000011444092]
```

Test the model

```
prediction=model.predict(X_test[:4])
print(prediction)
```

Python

```
1/1 [==============================] - 0s 109ms/step
[[4.21976367e-19 2.76640418e-21 1.07767846e-13 4.71178166e-11
  2.10180035e-22 1.09333558e-23 3.61171152e-26 1.00000000e+00
  6.55367672e-18 5.30832922e-15]
 [1.43194623e-10 1.67714037e-11 9.99998331e-01 1.29229049e-12
  2.77214423e-15 6.86406363e-20 1.72249804e-06 5.78033124e-16
  6.75309364e-10 1.71233192e-21]
```

Test the model

```
prediction=model.predict(X_test[:4])
print(prediction)
```

Python

```
1/1 [==============================] - 0s 109ms/step
[[4.21976367e-19 2.76640418e-21 1.07767846e-13 4.71178166e-11
  2.10180035e-22 1.09333558e-23 3.61171152e-26 1.00000000e+00
  6.55367672e-18 5.30832922e-15]
 [1.43194623e-10 1.67714037e-11 9.99998331e-01 1.29229049e-12
  2.77214423e-15 6.86406363e-20 1.72249804e-06 5.78033124e-16
  6.75309364e-10 1.71233192e-21]
 [1.41170807e-09 9.99988927e-01 9.63070237e-09 3.00579769e-11
  4.58667927e-07 1.44993351e-08 1.61555591e-09 6.39673292e-09
  6.12086751e-07 4.38722626e-12]
 [1.00000000e+00 1.69138509e-15 1.90748825e-13 5.08622956e-16
  4.55195829e-16 3.16392881e-14 1.27083393e-08 5.42493669e-14
  6.74359917e-12 2.46580225e-11]]
```

```
import numpy as np
print(np.argmax(prediction,axis=1))
print(y_test[:4])
```

Test with saved model

```
from keras.datasets import mnist
from matplotlib import pyplot
(X_train,y_train),(X_test,y_test)=mnist.load_data()
print('X_train: ' +str(X_train.shape))
print('y_train: ' +str(y_train.shape))
print('X_test: ' +str(X_test.shape))
print('y_test: ' +str(y_test.shape))
from matplotlib import pyplot
for i in range(9):
  pyplot.subplot(330+1+i)
  pyplot.imshow(X_train[i],cmap=pyplot.get_cmap('gray'))
  pyplot.show()
```

Python

```
X_train:(60000, 28, 28)
y_train:(60000,)
X_test:(10000, 28, 28)
y_test:(10000,)
```

FLASK APP

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

UPLOAD_FOLDER = 'F:\ibm\IBM-Project-50222-1660900453-main\Application Building\data'


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

model = load_model("./models/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))
```

```
@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)
```

HOME PAGE (HTML)

```
<html>

<head>
  <title>HDR</title>

  <meta name="viewport" content="width=device-width">

  <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">

  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" integrity="sha384-ggOyR0iXCbMQv3Xipma34M
  <link rel="stylesheet" type= "text/css" href= "{{ url_for('static',filename='css/style.css') }}">

  <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+8abtTE1Pi6j1
  <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js" integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clH
  <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js" integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf
  <script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@latest"></script>
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.1/dist/css/bootstrap.min.css">
  <script src="https://cdn.jsdelivr.net/npm/jquery@3.6.0/dist/jquery.slim.min.js"></script>
  <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js"></script>
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.1/dist/js/bootstrap.bundle.min.js"></script>

</head>
<style>
    body{
     background-image: url('static/images/bc1.jpg');
     background-repeat: no-repeat;
```

```
<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>
<div class="container p-7 my-7 bg-dark text-light">
    <h1>HandWritten Digit Recognition System</h1>
</div>
        <section id="content">

            <div class="leftside">
            <form action="/predict" method="POST" enctype="multipart/form-data">
            <label><h3>Select a image:</h3></label>
            <input id="image" type="file" name="image" accept="image/png, image/jpeg" onchange="preview()"><br><br>
              <img id="frame" width="100px" height="100px"/>
              <div class="buttons_div">
                <button type="submit" class="btn btn-dark">Predict</button>
                <button type="button" class="btn btn-dark">  Clear  </button>
              </div>
            </form>
            </div>
```

```
            <input id="image" type="file" name="image" accept="image/png, image/jpeg" onchange="preview()"><br><br>
              <img id="frame" width="100px" height="100px"/>
              <div class="buttons_div">
                <button type="submit" class="btn btn-dark">Predict</button>
                <button type="button" class="btn btn-dark">  Clear  </button>
              </div>
            </form>
            </div>
        </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" 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>
            <button type="submit" class="btn btn-light">Predict</button>
            <button type="button" class="btn btn-light">  Clear  </button>
          </div>
        </form>
        </div>
    </section>-->

</body>

</html>
```

PREDICT PAGE

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

<style>
    body{
     background-image: url('static/images/bc2.jpg');
     background-repeat: no-repeat;
     background-size: cover;
    }

    #rectangle{
     width:400px;
     height:150px;
     background-color: #000000;
     border-radius: 15px;
     position:absolute;
     box-shadow: 0px 0px 10px 5px white;
     top:25%;
     left:50%;
     transform:translate(-50%,-50%);
    }

    #head{
  text-align: center;
  font-size: 30px;
  margin: 0 auto;
  padding: 3% 5%;
```

GITHUB LINK

https://github.com/IBM-EPBL/IBM-Project-25604-1659968485

DEMO VIDEO LINK

https://drive.google.com/file/d/1T_F4bcffOYVVN18R5QVpGyiGw1L-kjM6/view?usp=drivesdk