A Novel Method for Handwritten Digit Recognition System.

A PROJECT REPORT

Submitted by

SANDHOSHSIVAN M	**(721819104023)**
VIKRAM	**(721819104033)**
MAHALINGAM K	**(721819104011)**
PAVUNKUMAR S	**(721819104023)**

in the partial fulfilment of the requirements for the award of a degree of

NAME OF THE DEGREE

IN

BRANCH OF STUDY

RATHINAM TECHNICAL CAMPUS, COIMBATORE.

ANNA UNIVERSITY: CHENNAI 600 025

NOVEMBER & 2022

Project Report Format

1. **INTRODUCTION**
   - 1.1 Project Overview
   - 1.2 Purpose
2. **LITERATURE SURVEY**
   - 2.1 Existing problem
   - 2.2 References
   - 2.3 Problem Statement Definition
3. **IDEATION & PROPOSED SOLUTION**
   - 3.1 Empathy Map Canvas
   - 3.2 Ideation & Brainstorming
   - 3.3 Proposed Solution
   - 3.4 Problem Solution fit
4. **REQUIREMENT ANALYSIS**
   - 4.1 Functional requirement
   - 4.2 Non-Functional requirements
5. **PROJECT DESIGN**
   - 5.1 Data Flow Diagrams
   - 5.2 Solution & Technical Architecture
   - 5.3 User Stories
6. **PROJECT PLANNING & SCHEDULING**
   - 6.1 Sprint Planning & Estimation
   - 6.2 Sprint Delivery Schedule
   - 6.3 Reports from JIRA
7. **CODING & SOLUTIONING (Explain the features added in the project along with code)**
   - 7.1 Feature 1
   - 7.2 Feature 2
   - 7.3 Database Schema (if Applicable)
8. **TESTING**
   - 8.1 Test Cases
   - 8.2 User Acceptance Testing
9. **RESULTS**
   - 9.1 Performance Metrics
10. **ADVANTAGES & DISADVANTAGES**
11. **CONCLUSION**
12. **FUTURE SCOPE**
13. **APPENDIX**

    Source Code

    GitHub & Project Demo Link

1.INTRODUCTION

Project Name:

A Novel Method for Handwritten Digit Recognition System.

Domain:

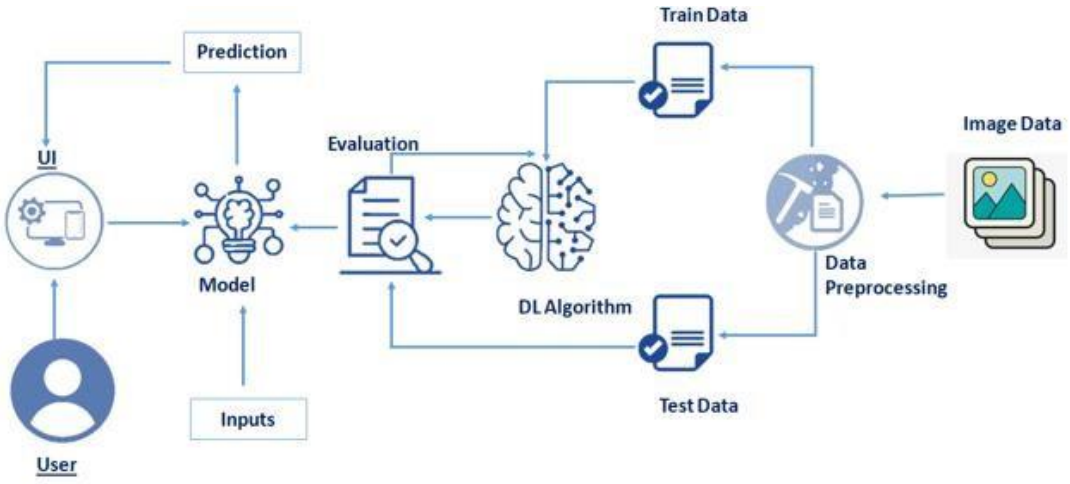
Artificial Intelligence.

Description:

Handwriting recognition is one of the compelling Research works going on because every individual in this world has their own style of writing. It's the capability of computer to identify and understand handwritten digits or character automatically. Because of the progress in this field of science and technology. Everything is being digitalized to reduce human effort. Hence, there comes need for handwritten digit-recognition in many real time applications. We use Artificial Intelligence 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. The image is analyzed by the model and the detected result is returned on to UI.

The handwritten digit recognition is the capability of computer applications to recognize the human handwritten digits. It is the hard task for the machine because handwritten digits are not perfect and can made many different shapes and sizes. The handwritten digit recognition is a way to tackle this problem which cause the images of digit and recognizes the digit present in the image.

The handwritten digit recognition is a way to tackle this problem which cause the images of digit and recognizes the digit present in the image. It 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. A Convolutional Neural Network or CNN is a Deep Learning algorithm which is very effective in handling image classification tasks. The famous MNIST data set is widely used for this recognition process and it has 70000 handwritten digits. It is a widely used and deeply understood dataset and, for the most part, is "solved." The MNIST dataset was compiled with images of digits from various scanned documents and then normalized in size.



Technical Architecture

2.LITERATURE SURVEY

2.1. Existing problem:

1. Ensemble neural networks that combined with ensemble decision tree.

2. Digit recognition using single layer neural Network with principal component analysis.

3. Recognition using Simple Neural network and back propagation**.**

2.2. References:

1. L. Bottou, C. Cortes, "Comparison of Classifier methods a case study in handwritten digit recognition", Pattern Recognition, 1994. Vol. 2 Conference B; Image Processing, Proceedings of the 12th IAPR International. Conference IEEE, 06 August 2002.

2."Skeletonization Algorithm for Binary Images - ScienceDirect." Available: https://www.sciencedirect.com/science/article/pii/S2 212017313004027. [Accessed: 15-Jan2019].

3. Y.-K. Chen and J.-F. Wang, "Segmentation of single-or multipletouching handwritten numeral string using background and foreground analysis," IEEE Trans. Pattern Anal. Mach. Intell., no. 11, pp. 1304– 1317, 2000.

4. Vandana Pujari and Kishor Mane "Signature Matching with Automated Cheque System",2013 International Conference on Intelligent Systems and Signal Processing (ISSP), Volume 10, No 1, Feb 2002.

5. Mathias M.Adankon, Mohamed Cheriet, "Model selection for the LSSVM. Application to handwriting recognition", Pattern Recognition, vol.42, pp.3264-3270, (2009).

2.3. Problem Statement Definition:

Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes mefeel
PS-1	Handwritten input from sources such as paper documents	The SVM classifier is the most accuratein terms of accuracy	automatically infer rules for recognizing handwritten digits	functional model is to maintain easiness while connecting the layers	machine learning and computer vision applications
PS-2	There are a number of ways and algorithms to recognize handwritten digits	The handwritten digit recognition is the ability of computers	These features are based on Shape analysis of the digit image	The applications of digit Recognition include in postal mail sorting,	ability of a computerto receive and interpret intelligible handwritten inputfrom sources

Following are the constraints faced when computers approach to recognize handwritten digits:

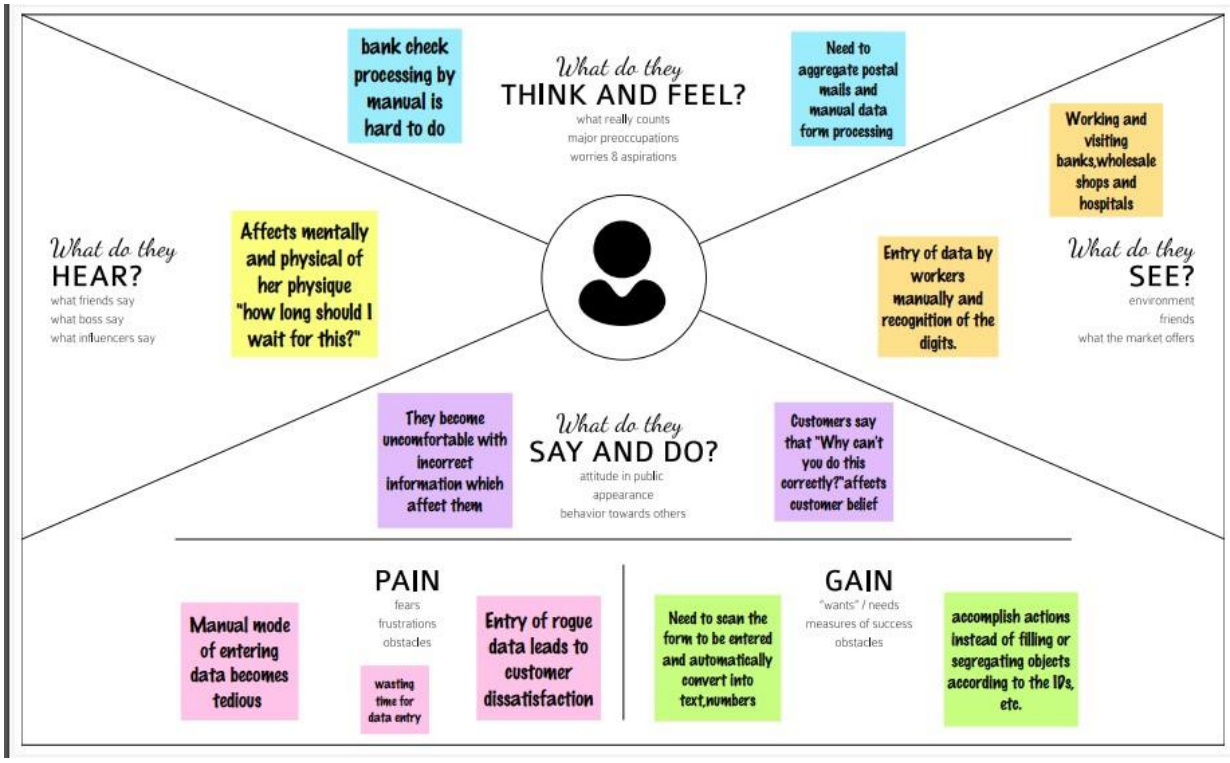
1. The Handwritten digits are not always of the same size, width, orientation and justified to margins as they differ from writing of person to person.

2. The similarity between digits such as 1 and 7, 5 and 6, 3 and 8, 2 and 7 etc. So, classifying between these numbers is also a major problem for computers.

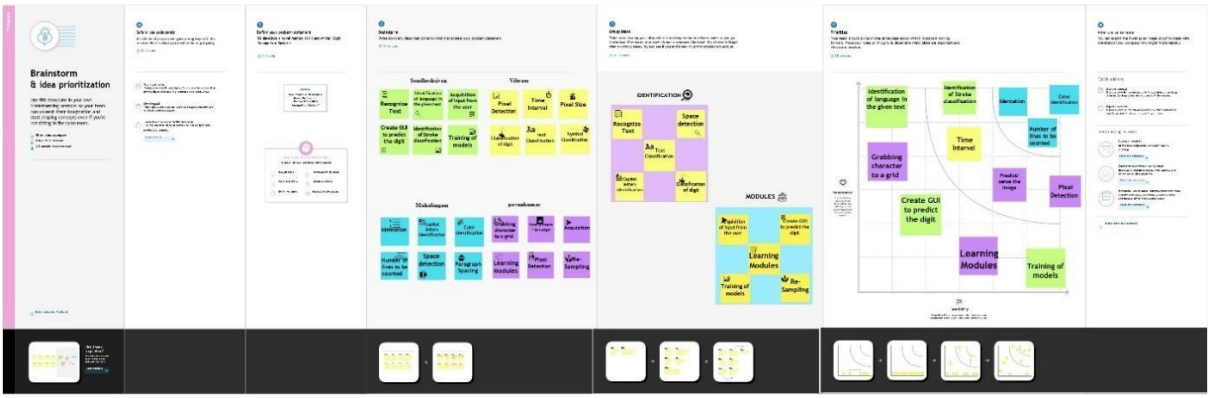
3. The uniqueness and variety in the handwriting of different individuals also influence the formation and appearance of the dig.

3.IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas:



3.2 Ideation & Brainstorming:



3.3 Proposed Solution:

Problem Statement (Problem to be solved):

Handwriting recognition is one of the compelling research works going on because every individual in this world has their own style of writing. It is the capability of the computer to identify and understand handwritten digits or

characters automatically. Because of the progress in the field of science and technology, everything is being digitalized to reduce human effort. Hence, there comes a need for handwritten digit recognition in many real-time applications. MNIST data set is widely used for this recognition process and it has 70000 handwritten digits. We use Artificial neural networks to train these images and build a deep learning model. Web application is created where the user can upload an image of a handwritten digit. this image is analyzed by the model and the detected result is returned on to UI.

Idea / Solution description:

HANDWRITTEN digit recognition isthe 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.

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 pybrains, Keras, Theano, Tensorflow by Google, TFLearn 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.

Novelty / Uniqueness:

The first layer of the architecture is the User layer. User layer will comprise of the people who interacts with the app and for the required results. The next three layers is the frontend architecture of the application. The application will be developed using Bootstrap which is the open source platform for HTML, CSS and JavaScript. The application is deployed in the localhost which is shown on the

browser. Through the app, the user will be able to upload pictures of the handwritten digits and convert it into the digitalized form. The one in between the database and view layer is the business layer which is the logical calculations on the basis of the request from the client side. It also host service interface. The backend layer consists of two datasets: Training Data and Test Data. The MNIST database has been used for that which is already divided into training set of 60,000 examples and test of 10,000 examples.

Social Impact/ Customer Satisfaction:

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.

Scalability of the Solution:

An implementation of Handwritten Digit Recognition using Deep Learning has been implemented in this paper. Additionally, some of the most widely used Machine Learning algorithms i.e. CNN using Tensor flow have been trained and tested on the same data to draw a comparison also why we require deep learning methods in critical applications like Handwritten Digit Recognition. In this project, we have shown that using Deep Learning techniques, a very high amount of accuracy can be achieved. Using the Convolutional Neural Network with Keras and Theano as backend, I am able to get an accuracy of 95.72%. Every tool has its own complexity and accuracy. Although, we see that the complexity of the code and the process is bit more as compared to normal Machine Learning algorithms but looking at the accuracy achieved, it can be said that it is worth it. Also, the current implementation is done only using the CPU. Thus we settled on classifying a given handwritten digit image as the required digit using three different algorithms and consequently testing its accuracy. In future we are planning to further explore the topic to recognize people’s handwriting.

3.4 Problem Solution fit:

SOLUTION:

MNIST Dataset Description

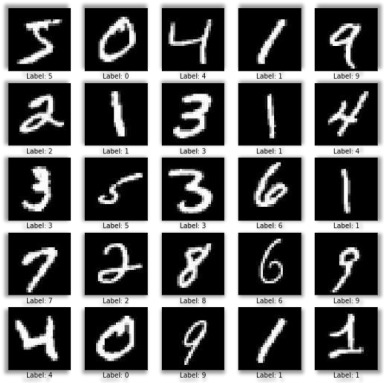
Handwriting recognition is one of the compelling research works going on because every individual in this world has their own style of writing. It is the capability of the computer to identify and understand handwritten digits or

characters automatically. Because of the progress in the field of science and technology, everything is being digitalized to reduce

human effort. Hence, there comes a need for handwritten digit recognition in many real-time applications. MNIST data set is widely used for this recognition process and it has 70000 handwritten digits. We use Artificial neural networks to train these images and build a deep learning model. Web application is created where theuser can uploadan image of a handwritten digit. this image is analyzed by the modeland the detected result is returned on to UI.

The MNIST Handwritten Digit Recognition Dataset contains 60,000 trainingand 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 indicates darker pixel). This pixel value is an integer ranging from 0 to 255.



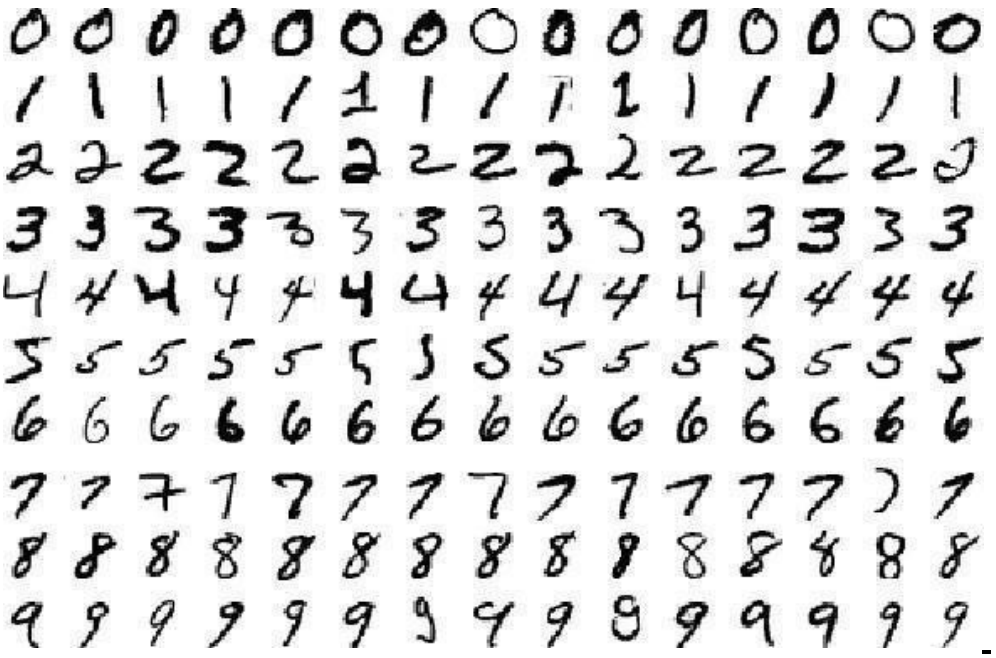
Procedure:

- Install the latest TensorFlow library.
- Prepare the dataset for the model.

- Develop Single Layer Perceptron model for classifying the handwritten digits.
- Plot the change in accuracy per epochs.
- Evaluate the model on the testing data.
- Analyze the model summary.
- 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.
- Use different optimizers and check its effect on accuracy.
- Increase the hidden layers and check its effect on accuracy.
- 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. Theartificial neural networks can all most mimic the human brain and are a key ingredient in image processing field. Handwritten digit recognition using MNISTdataset is a major project made with the help of Neural Network. It basically detectsthe scanned images of handwritten digits

We have taken this a step further where our handwritten digit recognition system not only detects scanned images of handwritten digits but also allows writing digits on the screen with the help of an integrated GUI for recognition.



Approach:

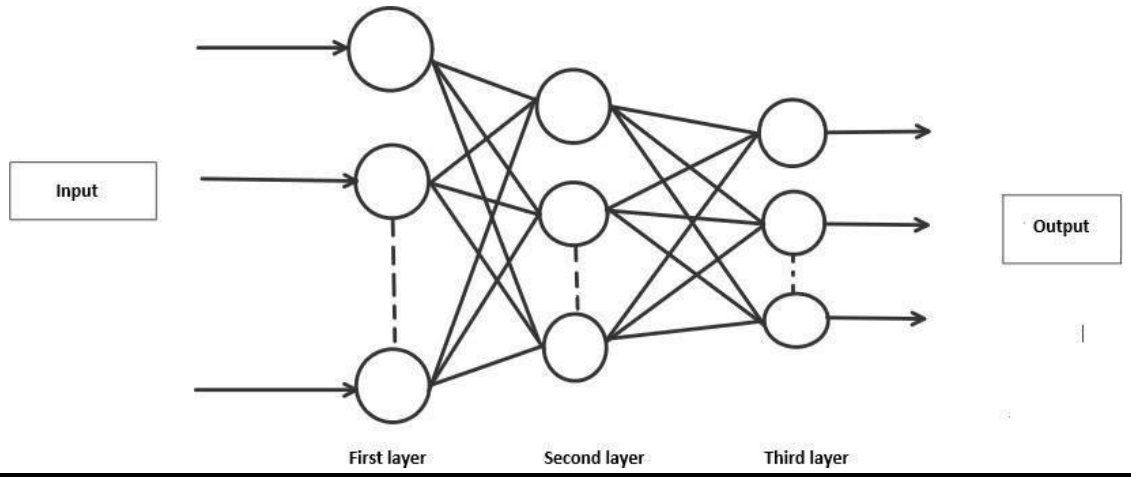
We will approach this project by using a three-layered Neural Network.

- 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.

METHODOLOGY:

We have implemented a Neural Network with 1 hidden layer having 100 activation units (excluding bias units). The data is loaded from a .mat file, features(X) and labels(y) were extracted. Then features are divided by 255 to rescale them into a range of [0,1] to avoid overflow during computation. Data is split up into 60,000 training and 10,000 testing examples. Feedforward is performed with the training set for calculating the hypothesis and then backpropagation is done in order to reduce the error between the layers. The regularization parameter lambda is set to 0.1 to address the problem of overfitting. Optimizer is run for 70 iterations to find the best fit model.



ALGORITHM:

Forward Propagation Architecture: It is a small workflow of how CNN module will extract the features and classify the image based on it. The architecture shows the input layer, hidden layers and output layer of the network. There are many layers involved in the feature extraction phase of the network which involves convolution and sub sample.

EXPLANATION OF THE PROPOSED SYSTEM:

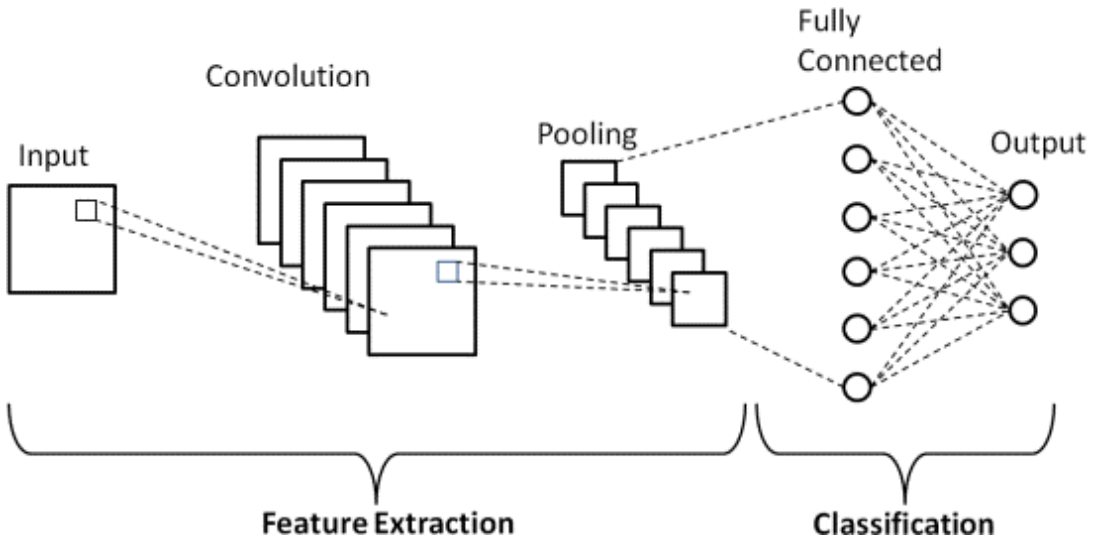
• The first layer of the architecture is the User layer. User layer will comprise of the people who interacts with the app and for the required results.

• The next three layers is the frontend architecture of the application. The application will be developed using which is the open-source platform for HTML, CSS and JavaScript. The application is deployed in the localhost which is shown on the browser. Through the app, the user will be able to upload pictures of the handwritten digits and convert it into the digitalized form.

• The one in between the database and view layer is the business layer which is the logical calculations on the basis of the request from the client side. It also has the service interface.

• The backend layer consists of two datasets: Training Data and Test Data. The MNIST database has been used for that which is already divided into training set of 60,000 examples and test of 10,000 examples.

• The training algorithm used is Convolution Neural Network. This will prepare the trained model which will be used to classify the digits present in the test data. Thus, we can classify the digits present in the images as: Class 0,1,2,3,4,5,6,7,8,9.



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:

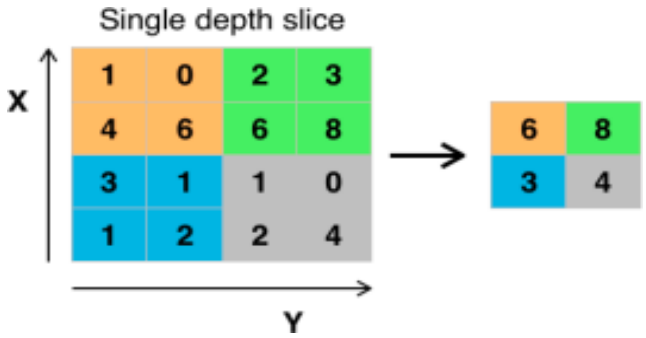
Subsampling, or down sampling, refers to reducing the overall size of a signal. The subsampling layers reduce the spatial resolution of each feature map. Reduce the effect of noises and shift or distortion invariance is achieved.

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.



Pooling layer

4.REQUIREMENT ANALYSIS

4.1. Functional requirement:

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	The product essentially converts handwritten digits to digital form.	The user is first asked to draw a number on the canvas, and the model that is built is then utilised to compare the data and provide an output in digitalized form.
FR-2	Recognizing the handwritten digit and displaying.	Recognizing the handwritten digit and displaying.
FR-3	Import dataset file directly to the program from a command that will download the dataset from its website. Save the dataset file in the same directory as the program	Installing packages and applications.
FR-4	Build a Neural Network with a number of nodes in the input layer equal to the number of pixels in the arrays	Nil
FR-5	Activating the Neural Network	Packages – tensor flow

4.2. Non-Functional requirements:

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	System design should be easily understood and user friendly to users. Furthermore, users of all skill levelsof users should be able to navigate it without problems.
NFR-2	**Security**	The system should automatically be able to authenticate all users with their unique usernameand password
NFR-3	**Performance**	Should reduce the delay in information When hundreds of requests are given.
NFR-4	**Availability**	Information is restricted to each users limited access

5.PROJECT DESIGN

5.1. Data Flow Diagrams:

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:

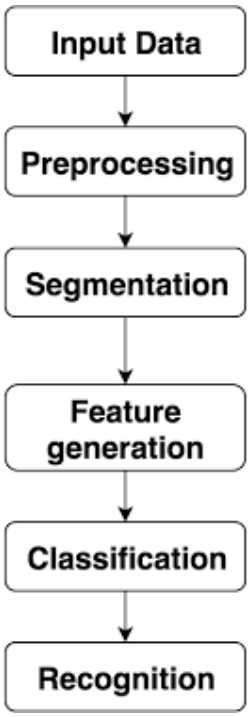
The DFD Level-1 consists of 2 external entities, the GUI and the Output, along with five process blocks and 2 data stores MNIST data and the Input image store, representing the internal workings of the CNN for Digit Recognition System. Process block imports MNIST data from library. Process block imports 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.

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 & Technical Architecture:



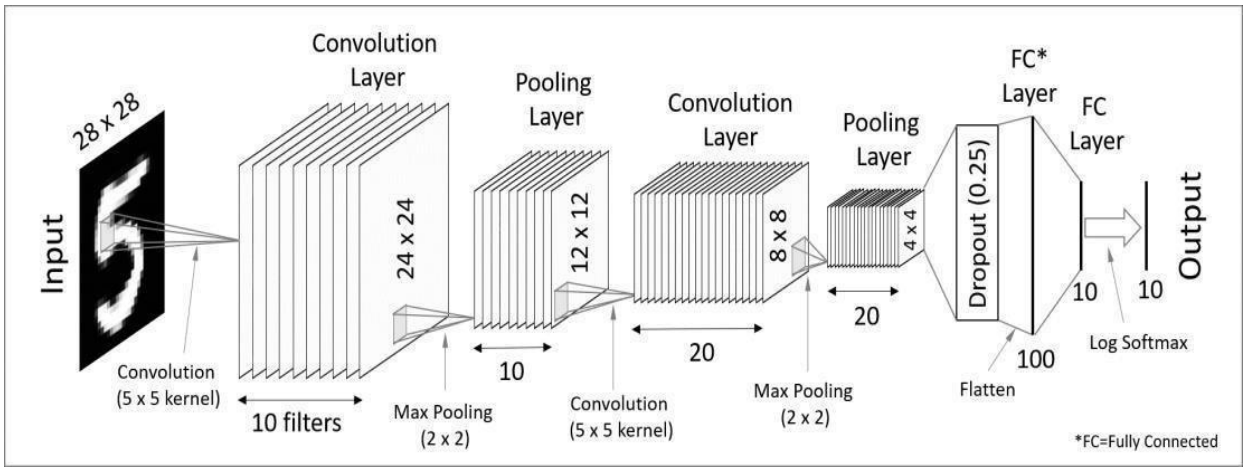


Table-1 : Components & Technologies:

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

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used	Technology of Opensource framework
2.	Security Implementations	List all the security / access controls implemented,use of firewalls etc.	SHA-256, Encryptions, IAM Controls,OWASP
3.	Scalable Architecture	Justify the scalability of architecture (3 – tier,Micro-services)	3 – tier, Micro-services
4.	Availability	Justify the availability of application (e.g. use of load balancers, distributed servers etc.)	Distributed servers, IBM cloud
5.	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc.	Number of requests per sec, use of Cache, use of CDN's

5.3.User Stories:

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer	Registration	USN-1	As a user, I can	I can access	High	Sprint-1
(Mobile user)			register for the application by entering my email,password, and confirming my password.	my account / dashboard		
	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
	Predict	USN-7	As a user I can able to get the recognized digit as output from the images of digital documents or images	I can access the recognizeddigits from digital document or images	High	Sprint-3

Customer Care Executive	Dashboard	USN-10	upload the image	Recognize andget the output	High	Sprint-1
Administrator	Security	USN-11	updated the features	checking the security	Medium	Sprint-1

6.PROJECT PLANNING & SCHEDULING

6.1. Sprint Planning & Estimation:

Sprint	Functional Requirement	Task
Sprint-1	Image Data	As a User need to collect the Image Data of Handly WrittenImages to train the model.
Sprint-2	Dash Board or Website	We using Python Flask Framework to create a dynamicWebpage to host our model (UI).
Sprint-3	Classifier Model	Using CNN Model for Image Classification.
Sprint-4	Cloud	Hosting the Organized application in Cloud platform.

6.2. Sprint Delivery Schedule:

Sprint	Duration	Sprint Start Date	Sprint End Date
Sprint-1	6 Days	25 Oct 2022	29 Oct 2022
Sprint-2	6 Days	31 Oct 2022	05 Nov 2022
Sprint-3	6 Days	07 Nov 2022	12 Nov 2022
Sprint-4	6 Days	14 Nov 2022	19 Nov 2022

7.CODING & SOLUTIONING

App.py:

```
import numpy as np
import os
from PIL import Image
from flask import Flask, request, render_template, url_for
from werkzeug.utils import secure_filename, redirect
#from gevent.pywsgi import WSGIServer
from keras.models import load_model
from keras.preprocessing import image
from flask import send_from_directory
UPLOAD_FOLDER = 'C:/Users/navie/Desktop/IBM Project/Project Development Phase/Sprint
    3/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))
    upload_img = os.path.join(UPLOAD_FOLDER, filepath)
    img = Image.open(upload_img).convert("L")  # convert image to monochrome
    img = img.resize((28, 28))  # resizing of input image
    im2arr = np.array(img)  # converting to image
    im2arr = im2arr.reshape(1, 28, 28, 1)  # reshaping according to our requirement
    pred = model.predict(im2arr)
    num = np.argmax(pred, axis=1) # printing our Labels
    return render_template('predict.html', num=str(num[0]))
if__name__== '__main__':
  app.run(debug=True, threaded=False)
```

Index.html:

```
<html>
<head>
  <title>Handwritten Digit Recognition</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-
     ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
     crossorigin="anonymous">
  <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+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>
  <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;
    background-size: cover;
   }
</style>
<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>IBM Guided project</h1>
    <div class="container p-3 my-3 bg-dark text-white">
      <p style = "text-align:center"> A Novel Method for Handwritten Digit Recognition
    System</p>
      <p style = "text-align:center"> Team ID - PNT2022TMID43750</P>
    </div>
    <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-light">Predict</button>
          <button type="button" class="btn btn-light">  Clear  </button>
        </div>
      </form>
      </div>
    </section>
  </body>
</html>
```

Predict.html:

```
<!DOCTYPE html>

<html lang="en">

<head>

   <meta charset="UTF-8">

   <title>Prediction</title>

</head>

<style>

   body{

    background-image: url('static/images/bc1.jpg');

    background-repeat: no-repeat;

    background-size: cover;

   }
```

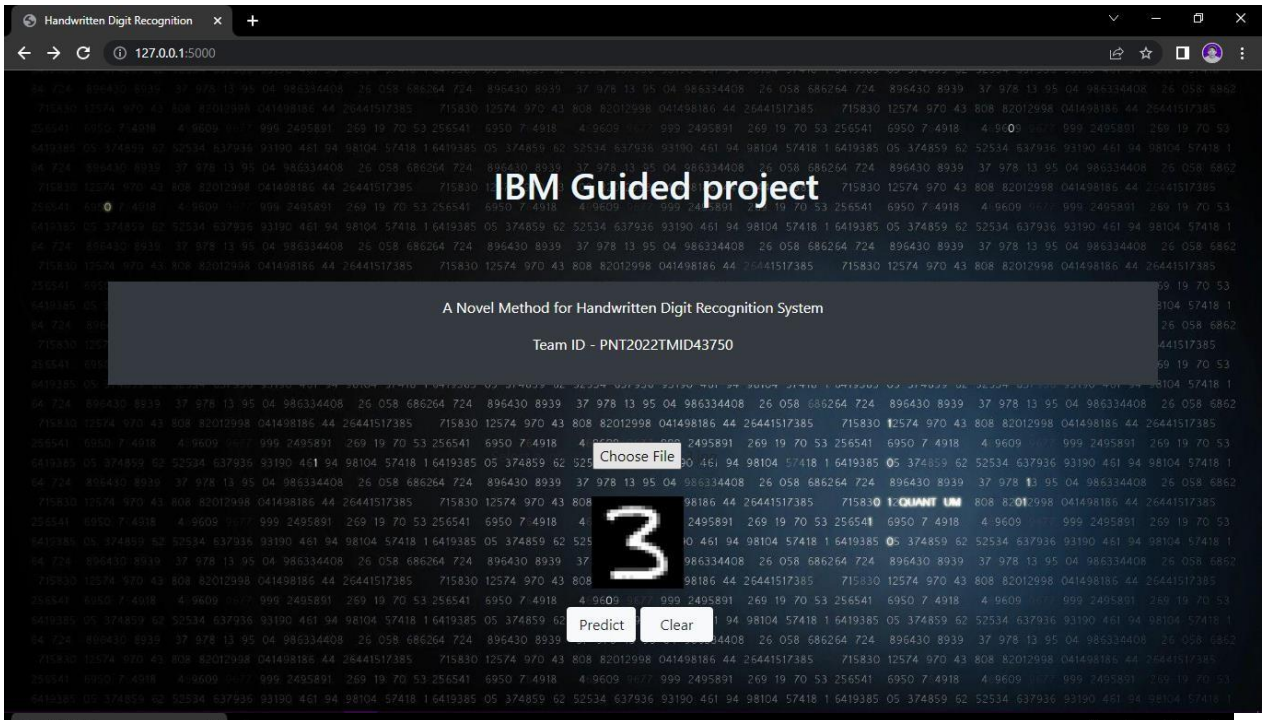
```
 #rectangle{
     width:400px;
     height:150px;
     background-color: #000000;
     border-radius: 15px;
     position:absolute;
     box-shadow: 0px 0px 10px 5px white;
     top:25%;
     left:50%;
     transform:translate(-50%,-50%);
    }
    #head{
  text-align: center;
  font-size: 30px;
  margin: 0 auto;
  padding: 3% 5%;
  font-family: Arial, Helvetica, sans-serif;
  color: white;
    }
    #num{
        font-size: 50px;
    }
</style>
<body>
    <div id="rectangle">
        <h1 id="head">Predicted Number : <br><center id="num">{{num}}</center></h1>
```

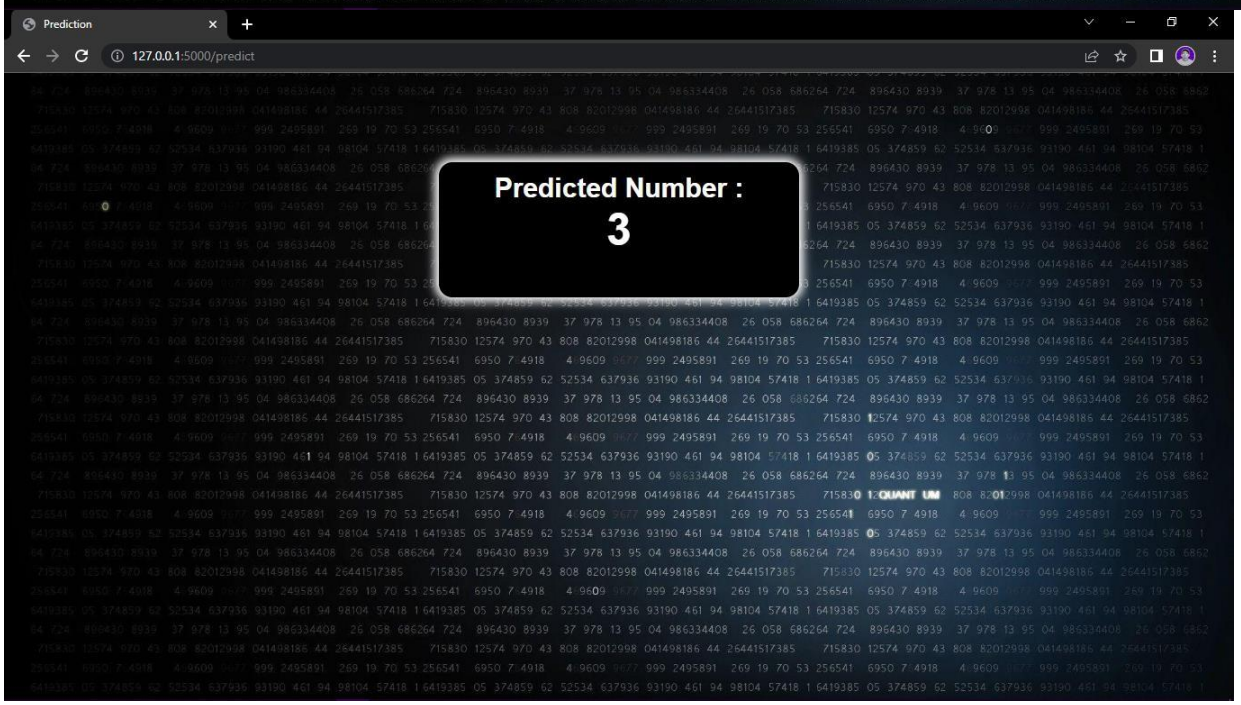
</body>

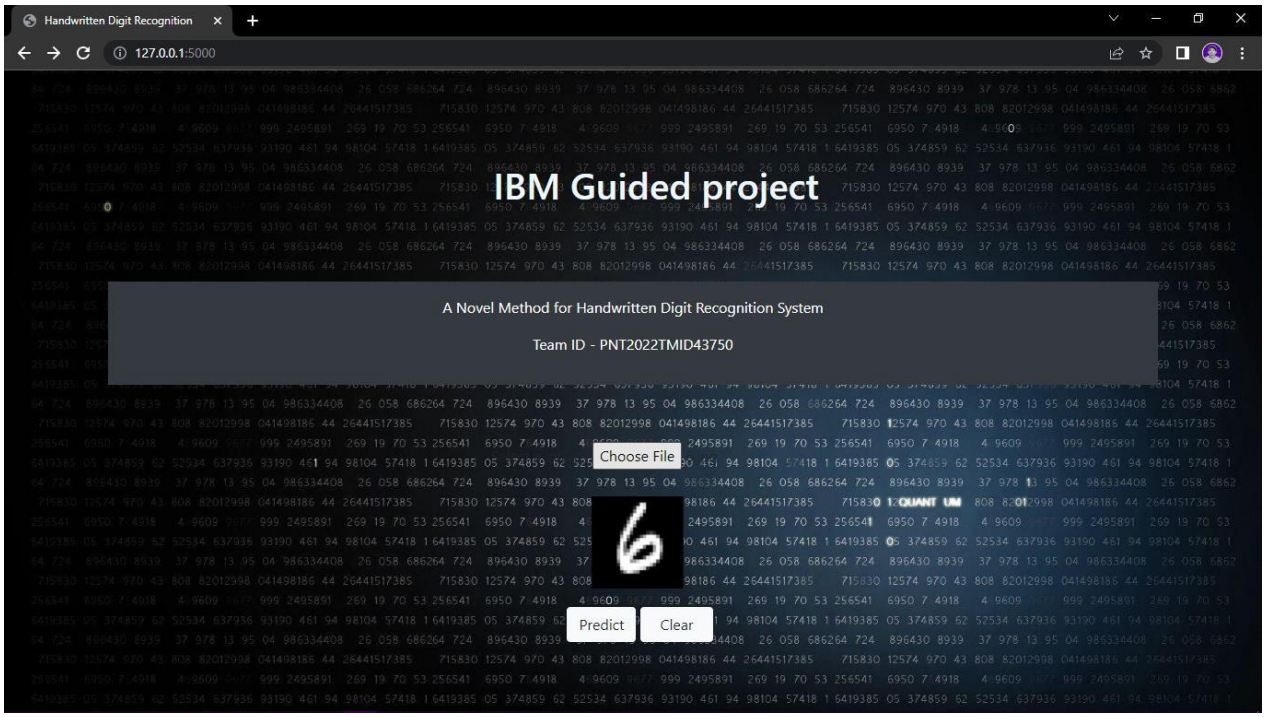
</html>

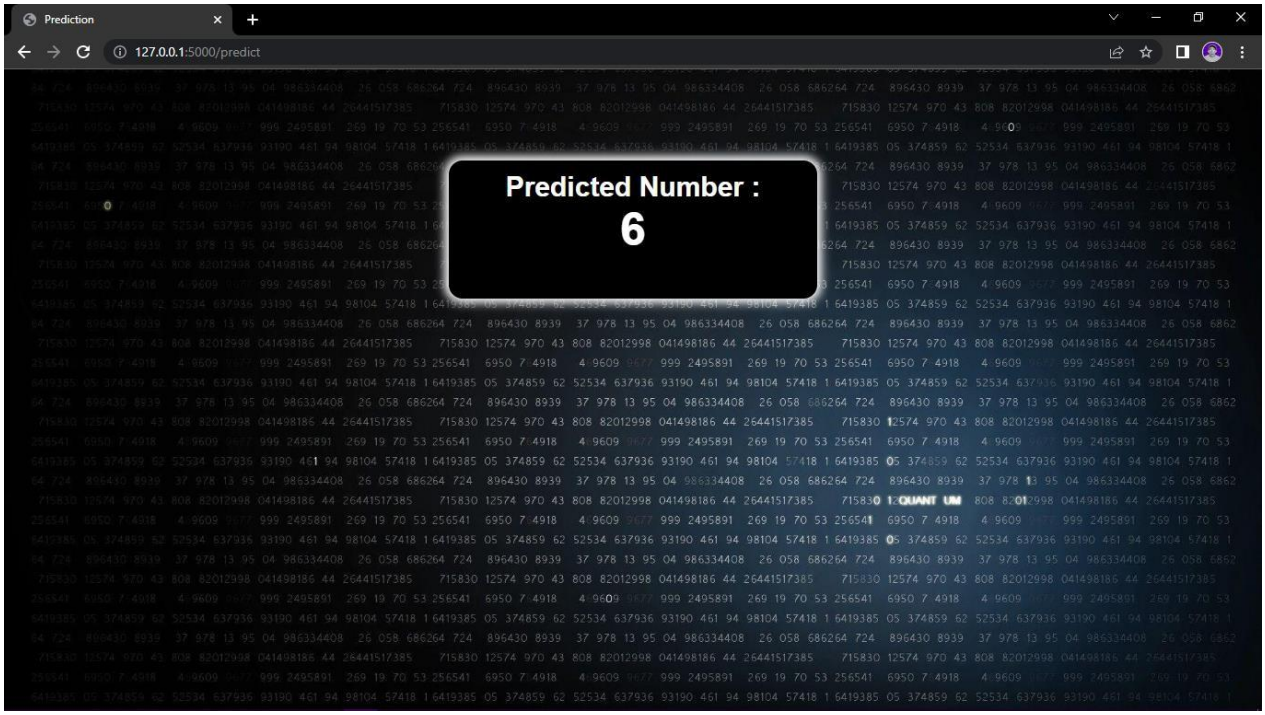
8.TESTING

8.1. Test Cases:



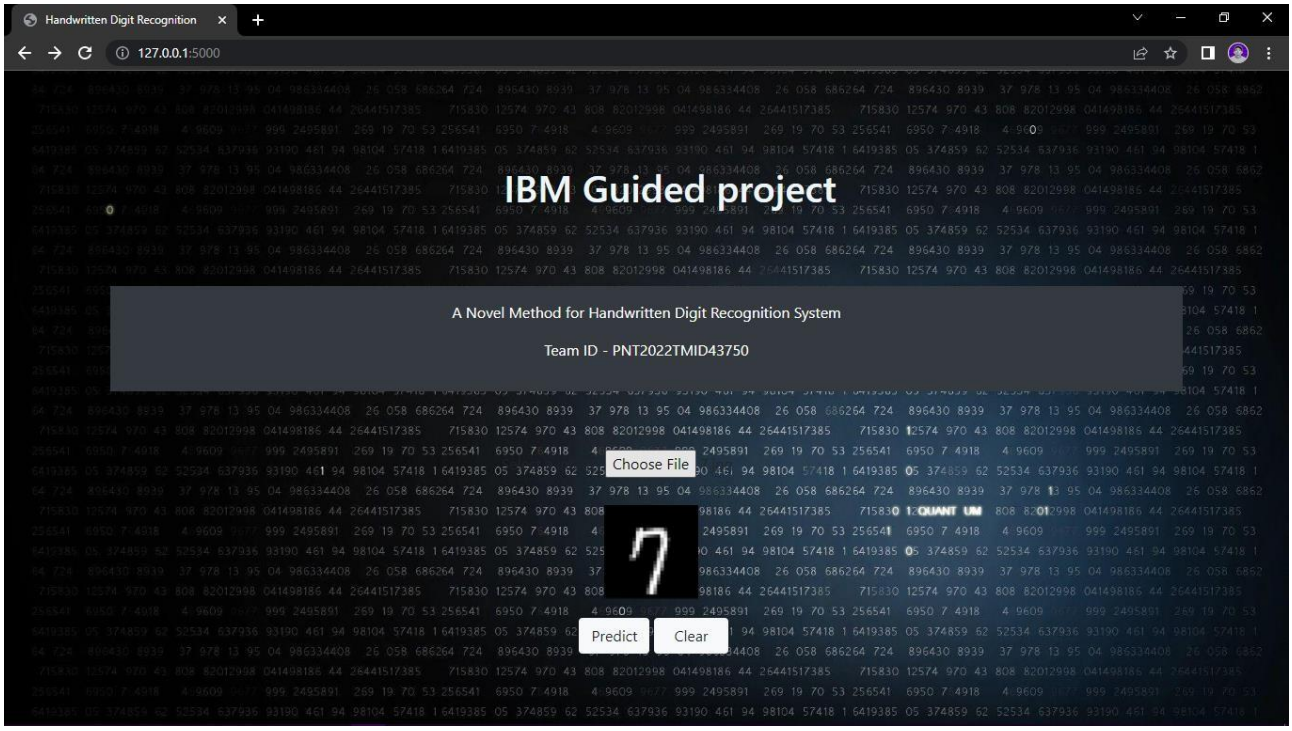


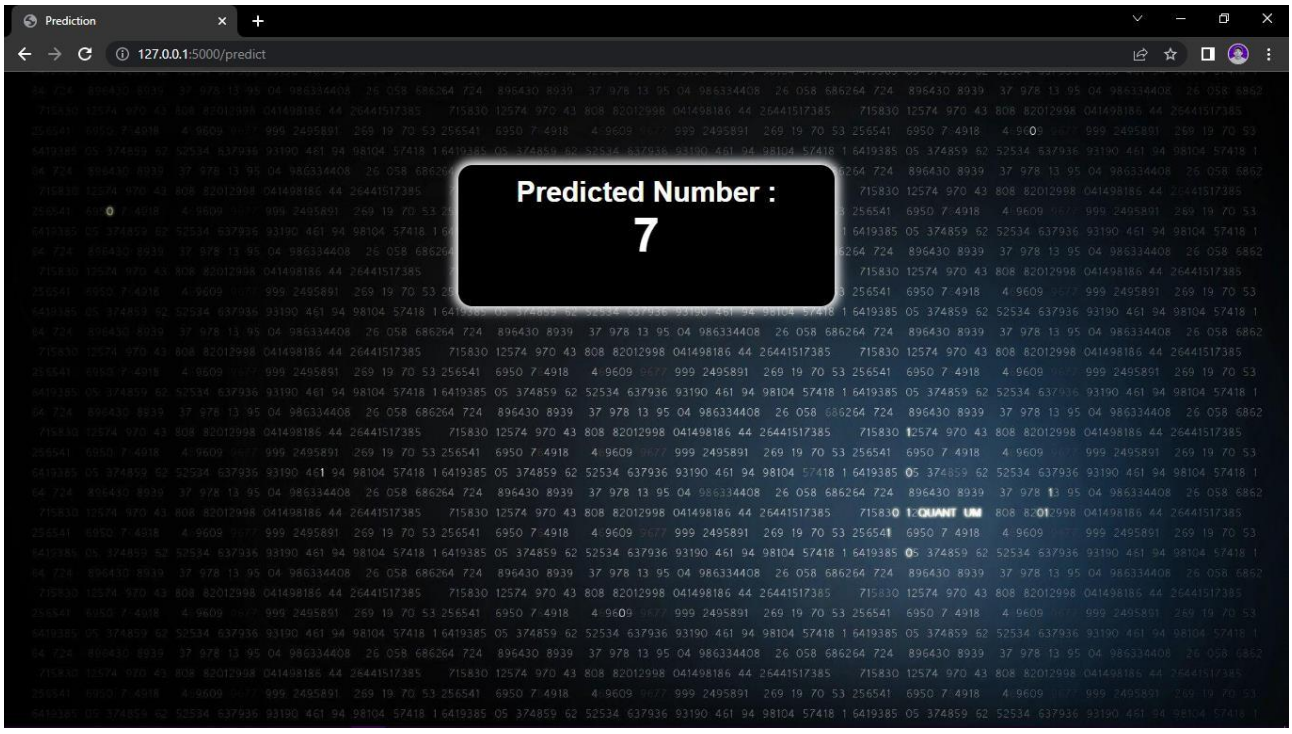




Input:6

Output:6

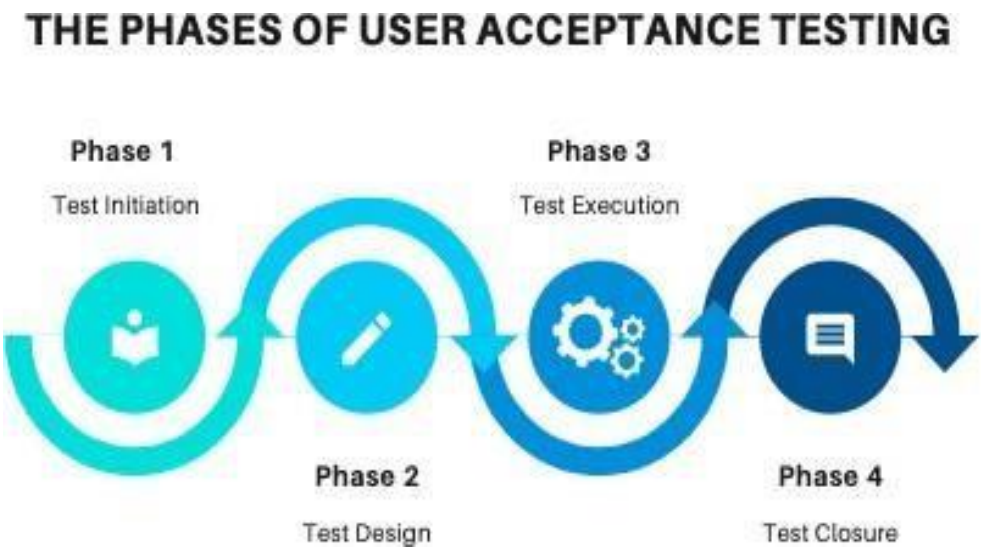




Input:7

Output:7

8.2. User Acceptance Testing:



UAT TEST INITIATION:

- UAT test approach is defined
- Business users who would be performing this testing areidentified
- Environments are sorted out
- Test data requirements are identified
- Required support from all other teams are discussed andsupport team identified

UAT TEST DESIGN:

- Business sceanrious to be validated are identified anddocument
- Relevant test data is identified
- Appropriate user accesses are requested and out

UAT TEST EXECUTION:

- Test execution of the business scenarios are prefrimed

- Appropriate defects are raised in the management tool
- Defect re-testing and regression testing is performed

UAT TEST CLOSURE:

- UAT closure report is produced
- GO/NO-GO decision is discussed andrecommended

9.RESULTS

9.1. Performance Metrics:

After implementing all the three algorithms that are SVM, MLP and CNN we have compared their accuracies and execution time with the help of experimental graphs for perspicuous understanding. We have taken into account the Training and Testing Accuracy of all the models stated above. After executing all the models, we found that SVM has the highest accuracy on training data while on testing dataset CNN accomplishes the utmost accuracy. Additionally, we have compared the execution time to gain more insight into the working of the algorithms. Generally, the running time of an algorithm depends on the number of operations it has performed. So, we have trained our deep learning model up to 30 epochs and SVM models according to norms to get the apt outcome. SVM took the minimum time for execution while CNN accounts for the maximum running time.

10. ADVANTAGES & 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 writing style.

2. The generative models can perform recognition driven segmentation.

3.Handwriting forces your brain to mentally engage with the information, improving both literacy and reading comprehension.

4.Digitalization

5.Data Collection.

10.2. DISADVANTAGES:

1. Despite that there are enormous convolutional neural network algorithms proposed for handwritten digit recognition, issues such as recognition accuracy.

2. Get alternative, less likely predictions when available.

3. Anyway Higher processor is required.

4. High cost

5. Time consuming

6. computation time still require for further improvement.

11. CONCLUSION

Recognition of characters and digits is viral in today's digitized world, especially in organizations that deal with handwritten documents that they need to analyze using computer systems. Convolutional Neural Network gets trained from the real-time data and makes the model very simple by reducing the number of variables and gives relevant accuracy. A comparison on different Machine Learning algorithms like Random Forest Classifier, Convolutional Neural Network, Linear Regression, K-Nearest Neighbors, Support vector machine is done, in which the accuracy for CNN is 99.63%. It can be used to convert books, newspapers and handwritten notes into digital text format using machine learning models.

CONCLUSION The paper discusses in detail all advances in the area of handwritten character recognition. The most accurate solution provided in this area directly or indirectly depends upon the quality as well as the nature of the material to be read. Various techniques have been described in this paper for character recognition in handwriting recognition system. A sort comparison is shown between the different methods proposed so far in the above table.

12. FUTURE SCOPE

The proposed recognition system is implemented on handwritten digits taken from MNIST database. Handwritten digit recognition system can be extended to a recognition system that can also able to recognize handwritten character and handwritten symbols. Future studies might consider on hardware implementation of recognition system.

The task of handwritten digit recognition, using a classifier, has great importance and use such as online handwriting recognition on computer tables, recognize zip codes on mail for postal mail sorting, processing, bank check amounts, numeric entries in forms filling up by hand and so on.

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

13.1. Source Code:

App.py:

```
import numpy as np
import os
from PIL import Image
from flask import Flask, request, render_template, url_for
from werkzeug.utils import secure_filename, redirect
#from gevent.pywsgi import WSGIServer
from keras.models import load_model
from keras.preprocessing import image
from flask import send_from_directory
UPLOAD_FOLDER = 'C:/Users/navie/Desktop/IBM Project/Project Development Phase/Sprint
    3/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))
        upload_img = os.path.join(UPLOAD_FOLDER, filepath)
        img = Image.open(upload_img).convert("L")  # convert image to monochrome
        img = img.resize((28, 28))  # resizing of input image
        im2arr = np.array(img)  # converting to image
        im2arr = im2arr.reshape(1, 28, 28, 1)  # reshaping according to our requirement
        pred = model.predict(im2arr)
        num = np.argmax(pred, axis=1) # printing our Labels
        return render_template('predict.html', num=str(num[0]))
if__name__== '__main__':
  app.run(debug=True, threaded=False)
```

Index.html:

```
<html>
<head>
  <title>Handwritten Digit Recognition</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-
     ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
     crossorigin="anonymous">
  <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+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>
  <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;
    background-size: cover;
   }
</style>
<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>IBM Guided project</h1>
        <div class="container p-3 my-3 bg-dark text-white">
           <p style = "text-align:center"> A Novel Method for Handwritten Digit Recognition
       System</p>
           <p style = "text-align:center"> Team ID - PNT2022TMID43750</P>
        </div>
        <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-light">Predict</button>
               <button type="button" class="btn btn-light">  Clear  </button>
             </div>
           </form>
           </div>
        </section>
  </body>
</html>
```

Predict.html:

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>Prediction</title>

 </head>

 <style>

    body{

    background-image: url('static/images/bc1.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%;
    font-family: Arial, Helvetica, sans-serif;
    color: white;
 }
 #num{
    font-size: 50px;
 }
  </style>
```

```
<body>

 <div id="rectangle">

     <h1 id="head">Predicted Number : <br><center id="num">{{num}}</center></h1>

 </div>

 </body>

</html>
```

13.2. GitHub & Project Demo Link:

GitHub: https://github.com/IBM-EPBL/IBM-Project-46828-1660792604.

Project Demo: https://github.com/IBM=EPBL/IBM-Project-46828-1660792604/blob/main/Final%20Deliverables/Demonstration%20video/Demonstration%20video.mp4.