PROJECT REPORT

A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

Submitted By

PNT2022TMID03914

Roobinee R - 412519104111

Swetha Lakshme S - 412519104150

Deepthi Sherly J - 412519104025

Indira Priyadharshini S - 412519104046

Meenashree S S - 412519104074

TABLE OF CONTENTS

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 digitized 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 analyzed by the model and the detected result is returned to the UI.

1.2 Purpose

Digital recognition is also an important issue. As handwritten digits are not the same size, thickness, position and direction, in this case by the way, various difficulties should be considered to find the handwritten digital recognition problem. I unique and a variety of creative styles for different people moreover have an influence on the model as well the presence of digits. It is a strategy to again edit written digits. It has a wide variety of applications, for example, scheduled bank checks, post offices and tax documents and so on. The purpose of this project is to use the classification algorithm to identify handwritten digits.

2. LITERATURE SURVEY

2.1 Existing problem

These days, a growing number of people are using images to transfer data. In addition it is the main distribution to separate the important data from the images. Image Recognition is an important research area for your most used apps. In general, in the field of pattern recognition, one of the most difficult tasks is the precise computerization of human handwriting. Without a doubt, this is a very difficult subject because there are so many variations of handwriting from person to person. Despite the fact that this difference does not cause problems for humans, however, it is becoming increasingly difficult to instruct computers to interpret common handwriting. In the case of image recognition, for example, classification by hand, it is important to know how the information is displayed in the pictures.

The Handwritten Recognition from the MNIST database is well known to scientists as through the use of different parameters separators, the error rate is reduced.for example, from the line phase (1 layer NN) and 12% to 0.23% with a board of 35 convolution neural systems. The scope of this is to use the Handwritten Digital Awareness Framework and think of different categories and strategies by focusing on how to achieve closeness to personal performance. In the task of naming different digits (0-9) for different people the most common issue to be dealt with is the issue of digit order and the closeness between digits such as 1 and 7, 5 and 6, 3 and 8, 9 and 8 and so on. .

In addition, people create the same digit with different ideas, the diversity and diversity in the handwriting of different people also contributes to the development and existence of digits.

2.2 References

[1] Handwritten Digit Classification Using the MNIST Dataset by M. Wu and Z. Zhang, 2010. [2] Handwritten digit recognition with decoding, A. Dutta and A. Dutta Handwritten digital recognition is utilized in this article. Comprehensive learning techniques have been established. The same data was used to train and evaluate a number of popular machine learning methods, including KNN, SVM, RFC, and CNN, to uncover comparisons across divisions. The more deeply you learn using these techniques, the more accurate you will be. In contrast to previous research

techniques, this technique focuses on which category helps construct models with separation accuracy of greater than 99%. When CNN is used as the backend and Tensorflow as the software, it can deliver accuracy of roughly 98.72%. CNN provides 98.72% accuracy in the first test.

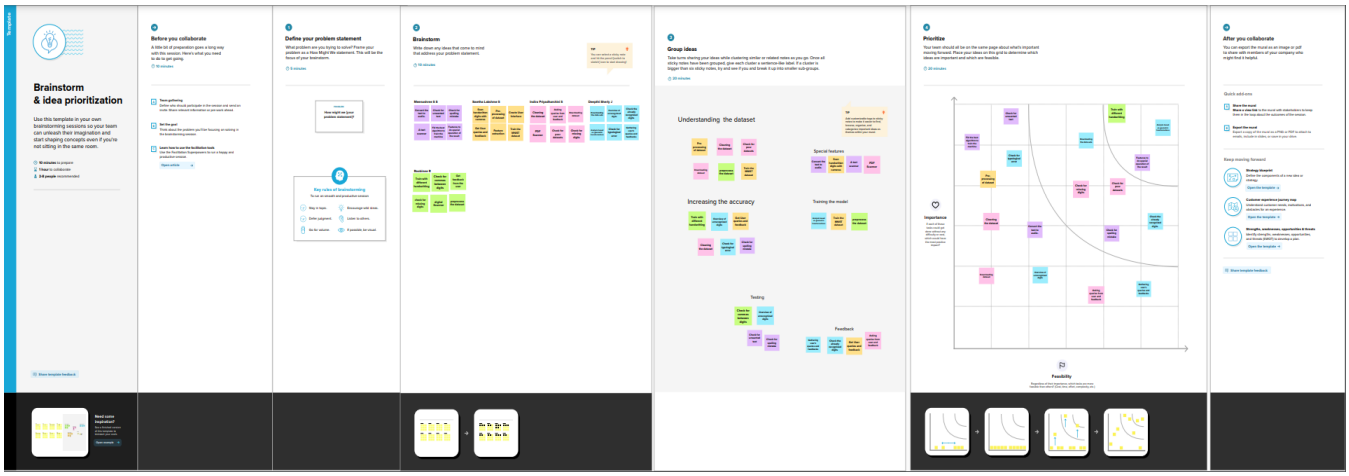
2.3 Problem Statement Definition

Abstract Handwritten Digit Recognition is the ability of computer systems to recognise handwritten digits from various sources, such as images, documents, and so on. Handwritten digit recognition has a wide range of applications, including number plate identification, bank check processing, address interpretation from envelopes, and so on. The fundamental problem with handwritten digit recognition is that handwritten digits do not always have the same size, width, orientation, and margins since they vary from person to person. Additionally, there would be issues with identifying the numbers because of similarities between numerals like 1 and 7, 5 and 6, 3 and 8, 2 and 5, 2 and 7, etc. Finally, the individuality and variation of each individual's handwriting influence the structure and appearance of the digits. Therefore, this project aims to address these problems and provide a novel handwritten digit recognition system.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

3.2 Ideation & Brainstorming

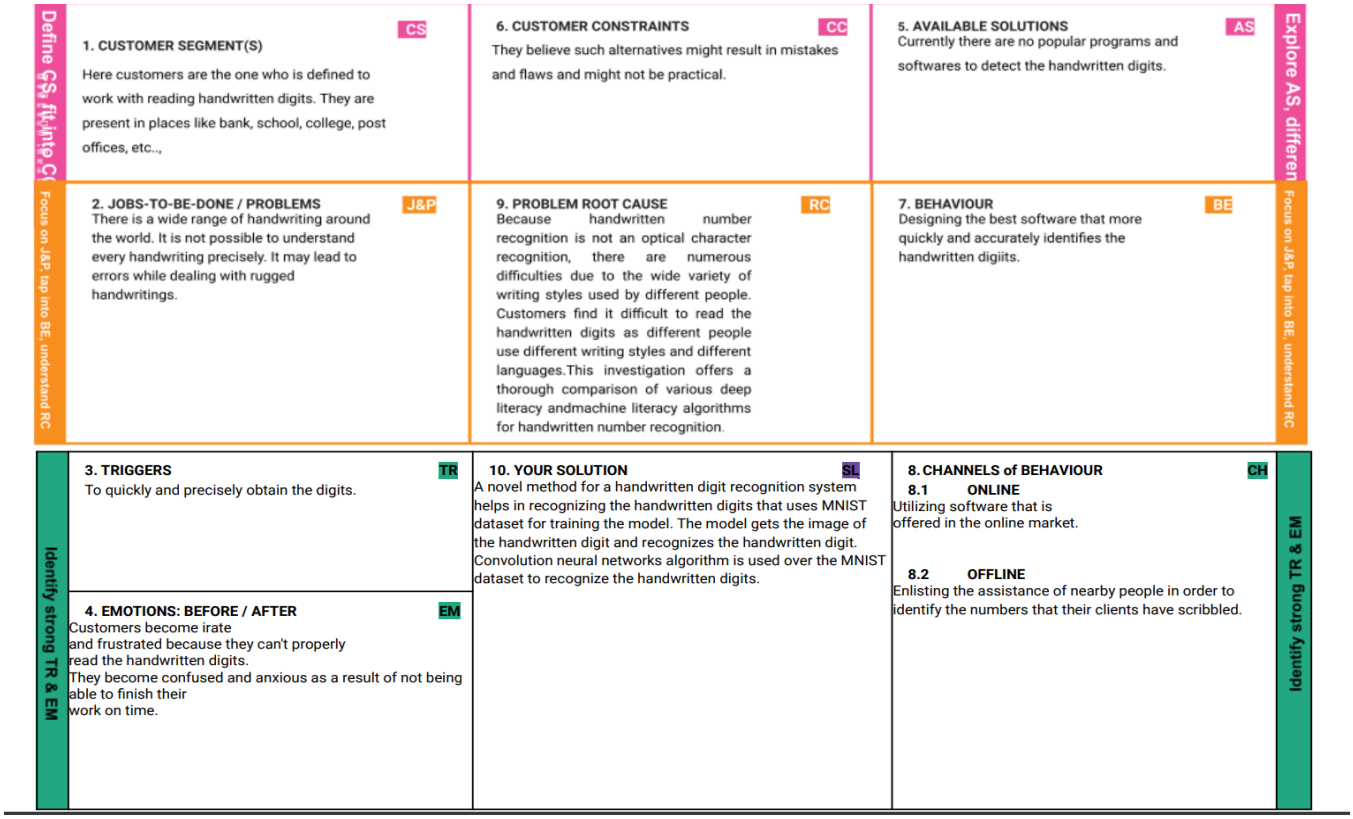


3.3 Proposed Solution

S.No.	Parameter	Description
1.	**Problem Statement (Problem to be solved)**	Statement-The capacity of computer programmes to detect human handwritten digits is known as handwritten digit recognition. Description: It is a hard task for the machine because handwritten digits are not perfect and can be made with many different shapes and sizes.
2.	**Idea / Solution description**	1. It is a computer's ability to celebrate the mortal handwritten numbers from many sources, such as photographs, papers, and touch defences. 2. All of those signatures and notes can be converted into electronic words in a text document

		format, and this data only needs a fraction of the physical storage space of the physical copies.
3.	**Novelty / Uniqueness**	Instead of recognising every character like OCR, accurately recognise the numbers and uses geometric transformations
4.	**Social Impact / Customer Satisfaction**	1. Handwritten digit Recognizer is an app that was created using artificial intelligence. 2. It approximates the printed word digitally and uses sophisticated algorithms to recognise characters before producing a digital approximation.
5.	**Business Model (Revenue Model)**	1. For efficient traffic control, this system can be linked with traffic Model) surveillance cameras to read licence plates. 2. Pin-code details can be easily identified and recognised by integrating with the postal system.
6.	**Scalability of the Solution**	Being able to distinguish numbers in noisy environments. The maximum number of digits that can be recognised is unrestricted.

3.4 Problem Solution fit



4. REQUIREMENT ANALYSIS

4.1 Functional Requirements

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	In essence, the product transforms handwritten digits into digital data.	In order to compare the data and produce an output in digital form, the user is first asked to draw a number on the canvas. The model that is then created is then used.
FR-2	Identifying and displaying the handwritten digit.	identifying and displaying the handwritten digit.

FR-3	Using a command that downloads the dataset from the program's website, you can import the dataset file directly into the software. In the same directory as the programme, save the dataset file.	Installing packages and applications.

4.2 Non Functional Requirements

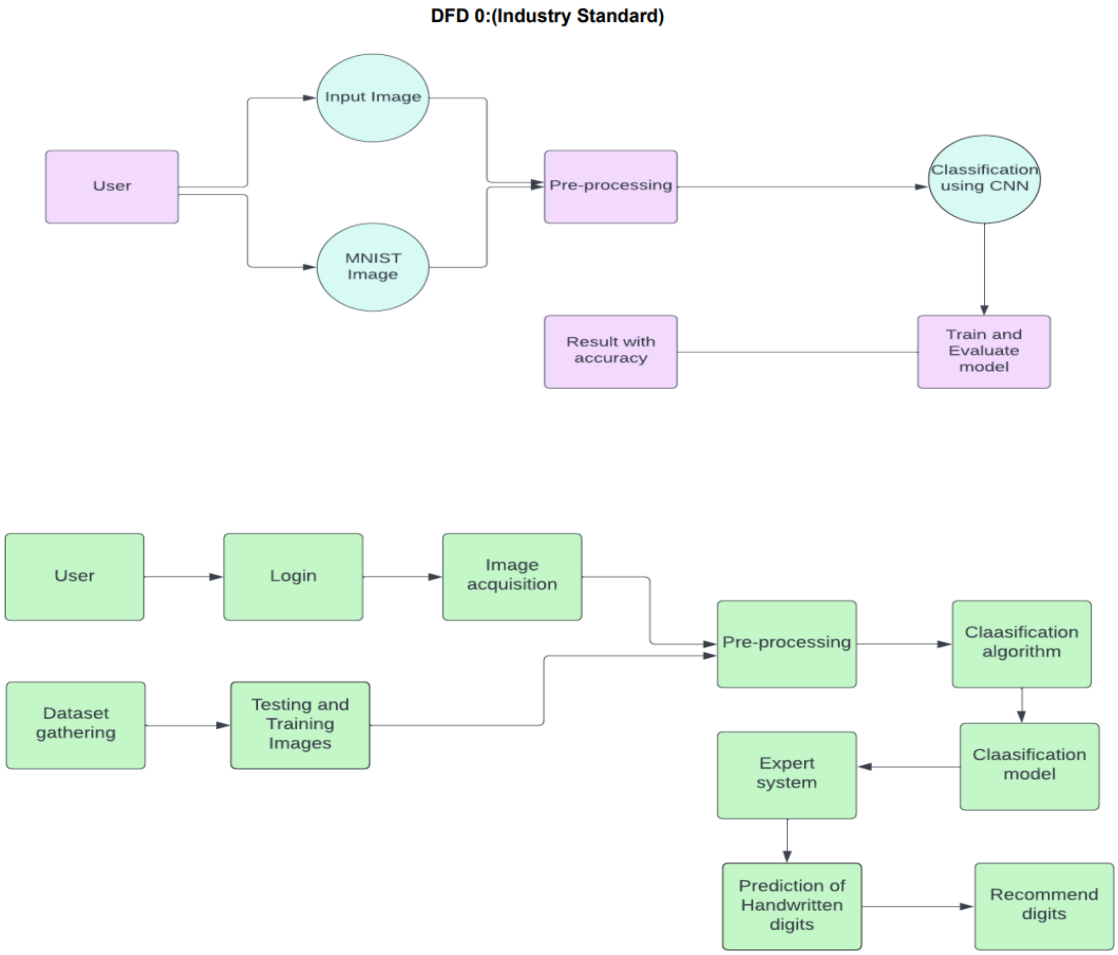
FR No.	Non-Functional Requirement	Description
NFR-1	Usability	System design should be easily understood and user friendly to users. Furthermore, users of all skill levels of users should be able to navigate it without problems.
NFR-2	Security	The system should automatically be able to authenticate all users with their unique username and 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

NFR-5	Scalability	the system should be able to handle 10000 users accessing the site at the same time

5. PROJECT DESIGN

5.1 Data Flow Diagrams

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



5.2 Solution & Technical Architecture

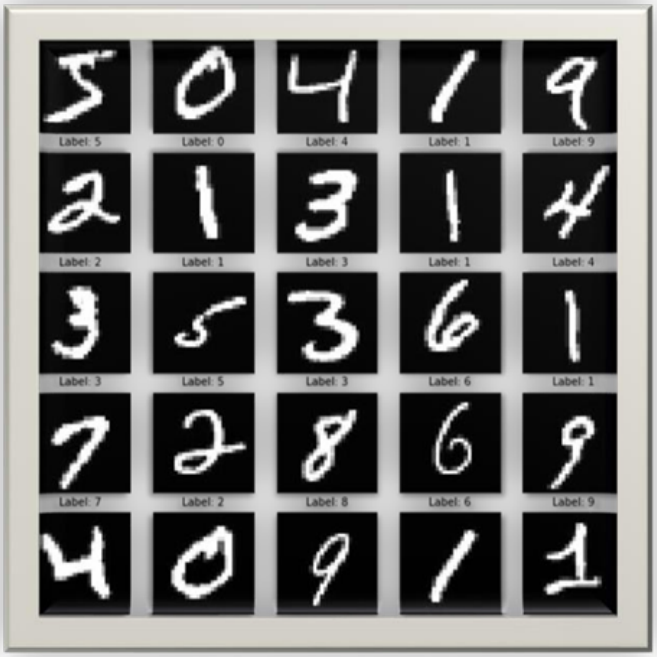
MNIST Dataset Description

Because everyone in the world has a unique writing style, handwriting identification is one of the fascinating research projects now being conducted. It is the ability of a computer to automatically recognise

and comprehend handwritten numbers or letters. Every aspect of life is being digitized to lessen the need for human labor as a result of advancements in science and technology. Thus, handwritten digit recognition is required in many real-time applications. The MNIST data collection, which contains 70000 handwritten digits, is frequently utilized for this recognition method. In order to train these photos and create a deep learning model, we use artificial neural networks. A web application is developed that allows users to upload pictures of handwritten numbers. The model examines this picture.

The 60,000 training and 10,000 testing labeled handwritten digit images in the MNIST Handwritten Digit Recognition Dataset.

Each image has a total of 784 (2828) pixels, or 28 pixels in height and 28 pixels in width. There is just one pixel value assigned to each pixel. It displays the brightness or darkness of that pixel (larger numbers indicate darker pixel). The integer for this pixel value ranges from 0 to 255.



PROCEDURE

- Install the TensorFlow library.
- Prepare the model's dataset.
- For the purpose of classifying the handwritten digits, create a single layer perceptron model.
- Plot the accuracy change over time.
- Analyze the test data to evaluate the model.
- Speculate on the model summary.
- To make the model a multi-layer perceptron, add a hidden layer.

- To avoid overfitting and assess its impact on accuracy, include Dropout.
- Increase the number of hidden layer neurons and assess how accuracy is affected.
- Test the impact of various optimizers on accuracy.
- Increase the hidden layers and assess the accuracy impact.
- Change the batch size and epochs, then assess the impact on accuracy.

For the recognition of handwritten digits, the MNIST dataset is frequently used. 10,000 test photos make up the dataset, which includes 60,000 training images. The discipline of image processing relies heavily on artificial neural networks because they most closely resemble the human brain.

A significant project done with the use of neural networks is the recognition of handwritten digits using the MNIST dataset. In essence, it recognizes digits that were scribbled and scanned.

Our handwritten digit identification system goes a step further in that it can now recognize handwritten numbers typed directly on the screen with the aid of an integrated GUI in addition to detecting them in scanned photos.

APPROACH

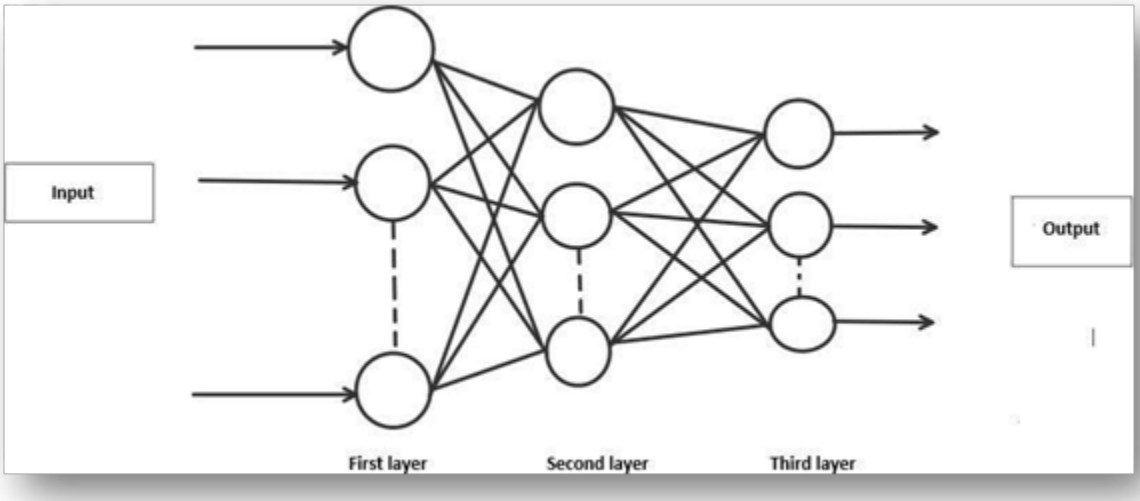
This project will be approached utilizing a three-layered neural network.

- **The input layer:** The input layer transfers the information from our example systems to the following layer so that the latter can compute its activations.
- **The hidden layer:** The network's nonlinear ties are provided by hidden units termed activations that make up the hidden layer. Depending on our needs, there can be a variety of concealed layers.
- **The output layer:** The nodes in this stratum are referred to as output units. It gives us access to the neural network's final prediction, which may be used to make final predictions.

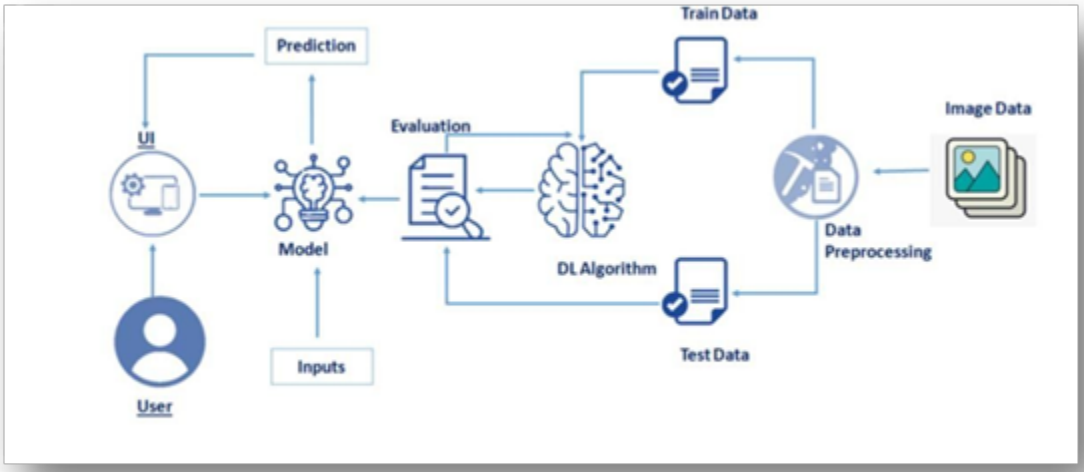
A neural network is a model of the brain's operations. It is made up of numerous layers with a variety of activations; these activations mimic the neurons in our brain. An attempt is made by a neural network to learn a set of parameters from a set of data that might aid in understanding the underlying relationships. Since neural networks are capable of adapting to changing input, the network can produce the best outcome without having to change the output criterion.

METHODOLOGY

A neural network with one hidden layer and 100 activation units has been put into practice (excluding bias units). The features (X) and labels (Y) were retrieved after the data was loaded from a.mat file. To prevent overflow during computation, features are then scaled into a range of [0,1] by dividing by 255. 10,000 testing cases and 60,000 training examples make up the data. With the training data, feedforward is used to calculate the hypothesis, and backpropagation is then used to lower the error between the layers. To combat overfitting, the regularization parameter lambda is set to 0.1. To identify the model that fits the situation the optimizer runs for 70 times.



TECHNICAL ARCHITECTURE



5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Understanding Data	USN-1		The user will get the desired output for the handwritten digit.	High	Sprint-1
	Preprocessing the data	USN-2		Only concentrating the dataset with the necessary content.	High	Sprint-1
	Model Building	USN-3		Adding the CNN layers and training the model for more accuracy	High	Sprint-2
	Dashboard	USN-4	Visit the dashboard for accessing the features of the application	Additionally, I'm able to comprehend directions, and the homepage is user-friendly.	Low	Sprint-3

	Upload Image	USN-5	As a user, I can able to input the images of digital documents to the application	As a user, I can able to input the images of digital documents to the application	High	Sprint-3
	Predict	USN-6	I can obtain the recognised digit from the images of digital documents or photos as an end user.	From digital documents or photos, I may access the identified digits.	High	Sprint-3
	Train the model using IBM	USN-7	I will train and evaluate the input as a user to ensure the output is as accurate as possible.	I'm able to train and test the application till the results are as accurate as possible.	Medium	Sprint-4
Administrator	Testing	USN- 8	Testing the features of the system	The system must recognize the digits in the most accurate manner.	High	Sprint-4
Customer Care Executive	Dashboard	USN-9	Upload the image	Recognizing and get the output	High	Sprint-3
Administrator	Security	USN-10	updated the features	checking the security	Medium	Sprint-4

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Understanding Data	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	Roobinee R Swetha Lakshme S Meenashree S S Deepthi Sherly J Indira Priyadharshni S
Sprint-1	Preprocessing the data	USN-2		2	High	Roobinee R Swetha Lakshme S Meenashree S S Deepthi Sherly JIndira Priyadharshni S
Sprint-2	Model Building	USN-2	Adding the CNN layers and training the model for more accuracy	1	High	Roobinee R Swetha Lakshme S Meenashree S S Deepthi Sherly J Indira Priyadharshni S
Sprint-3	Dashboard	USN-4	Visit the dashboard for accessing the features of the application	2	Medium	Roobinee R Swetha Lakshme S Meenashree S S Deepthi Sherly J Indira Priyadharshni S

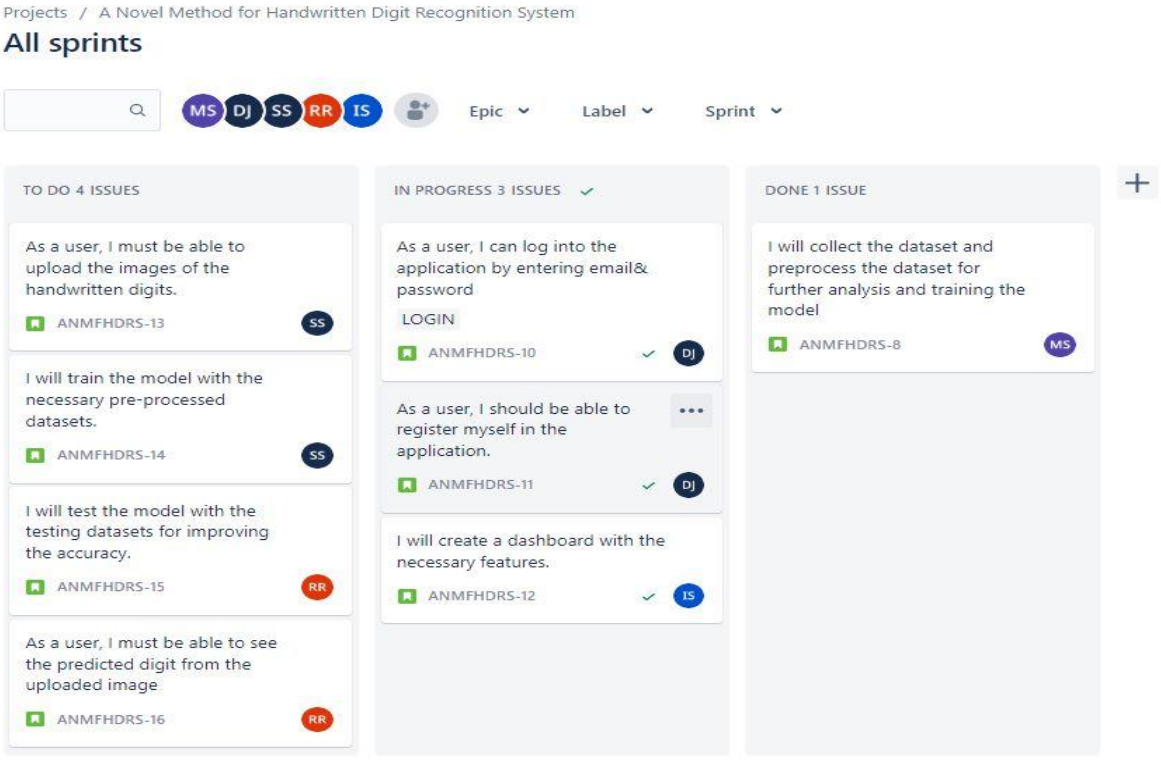
Sprint-3	Upload Image	USN-5	As a user, I can able to input the images of digital documents to the application	2	High	Roobinee R Swetha Lakshme S Meenashree S S Deepthi Sherly J Indira Priyadharshni S
Sprint-3	Predict	USN-6	I can obtain the recognised digit from the images of digital documents or photos as an end user.	2	High	Roobinee R Swetha Lakshme S Meenashree S S Deepthi Sherly J Indira Priyadharshni S
Sprint -4	Train the model using IBM	USN-7	I will train and evaluate the input as a user to ensure the output is as accurate as possible.	2	Mediu m	Roobinee R Swetha Lakshme S Meenashree S S Deepthi Sherly J Indira Priyadharshni S
Sprint-4	Testing	USN- 8	Testing the features of the system	2	High	Roobinee R Swetha Lakshme S Meenashree S S Deepthi Sherly J Indira Priyadharshni S

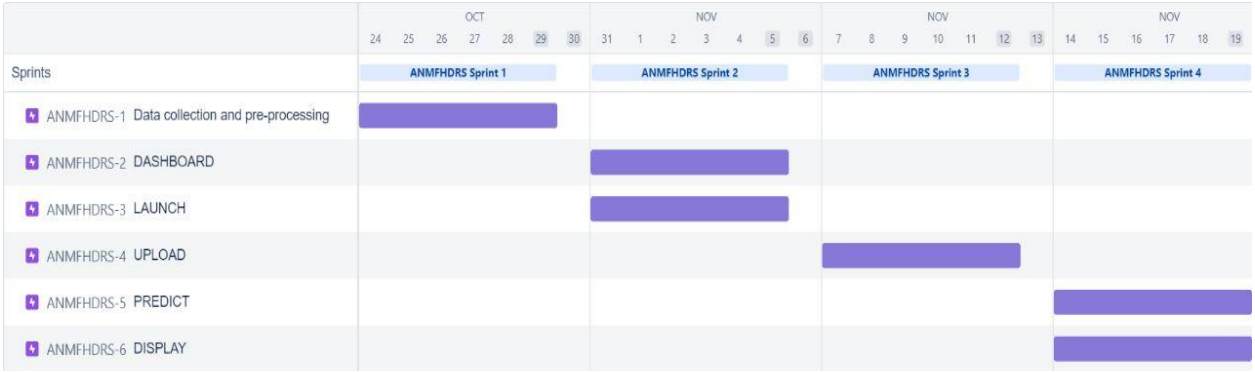
6.2 Sprint Delivery Schedule

Sprint	**Total Story Points**	**Duration**	**Sprint Start Date**	**Sprint End Date (Planned)**	**Story Points Completed (as on Planned End Date)**	**Sprint Release Date (Actual)**
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022

Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

6.3 Reports from JIRA





7. CODING & SOLUTIONING

```
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 = 'Static/uploads'

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)
```

8. TESTING

8.1 Test Cases

Test case ID	Feature Type	Component	Test Scenario	Steps To Execute	Test Data	Expected Result	Actual Result	Status	BUG ID	Executed By
HP_TC_001	UI	Home Page	Verify UI elements in the Home Page	1) Open the page 2) Check if all the UI elements are displayed 3)Click on My Account dropdown button 4)Verify login/Singup popup displayed or not	127.0.0.1.5000	The Home page must be displayed properly	Working as expected	Pass		Indira Priyadharshini S Deepthi Sherly J
HP_TC_002	UI	Home Page	Check if the UI elements are displayed properly in different screen sizes	1) Open the page in a specific device 2) Check if all the UI elements are displayed properly 3) Repeat the above steps with different device sizes	Screen sizes: 2560 x 1801, 1440 x 970, 1024 x 840, 768 x 630, 320 x 630	The Home page must be displayed properly in all sizes	Working as expected	Pass		Meenashree S S Roobinee R
HP_TC_003	Functional	Home page	Check if user can upload their file	1) Open the page 2) Click on select button 3) Select the input imag	1.png	The input image should be uploaded to the application successfully	Working as expected	Pass		Indira Priyadharshini S Roobinee R
HP_TC_004	Functional	Home Page	Check if user cannot upload unsupported files	1) Open the page 2) Click on select button 3) Select a random input file	installer.exe	The application should not allow user to select a non image file	User is able to upload any file	Fail	BUG_ID_001	Meenashree S S Indira Priyadharshini S

HP_TC_005	Functional	Home Page	Check if the page redirects to the result page once the input is given	1) Open the page 2) Click on select button 3) Select the input image 4) Check if the page redirects	1.png	The page should redirect to the results page	Working as expected	Pass		Deepthi Sherly J Swetha Lakshme S
BE_TC_001	Functional	Backend	Check if all the routes are working properly	1) Go to Home Page 2) Upload the input image 3) Check the reults page	1.png	All the routes should properly work	Working as expected	Pass		Meenashree S S Swetha Lakshme S
M_TC_001	Functional	Model	Check if the model can handle various image sizes	1) Open the page in a specific device 2) Upload the input image 3) Repeat the above steps with different input image	1.png, 2.png, 3.png	The model should rescale the image and predict the results	Working as expected	Pass		Swetha Lakshme S Roobinee R
M_TC_002	Functional	Model	Check if the model predicts the digit	1) Open the page 2) Click on select button 3) Select the input image 4) Check the results	1.png	The model should predict the number	Working as expected	Pass		Indira Priyadharshini S Deepthi Sherly J
M_TC_003	Functional	Model	Check if the model can handle complex input image	1) Open the page 2) Click on select button 3) Select the input image 4) Check the results	4.png	The model should predict the number in the complex image	Working as expected	Pass		Swetha Lakshme S Deepthi Sherly J
RP_TC_001	UI	Result Page	Verify UI elements in the Result Page	1) Open the page 2) Click on select button 3) Select the input image 4) Check if all the UI elements are displayed properly	1.png	The Result page must be displayed properly	Working as expected	Pass		Meenashree S S Roobinee R
RP_TC_002	UI	Result Page	Check if the input image is displayed	1) Open the page 2) Click on select button 3) Select the input image 4)	1.png	The input image should be displayed	Working as expected	Pass		Meenashree S S Deepthi Sherly J

			properly	Check if the input image are displayed		properly				
RP_TC_003	UI	Result Page	Check if the result is displayed properly	1) Open the page 2) Click on select button 3) Select the input image 4) Check if the result is displayed	1.png	The result should be displayed properly	Working as expected	Pass		Indira Priyadharshini S Deepthi Sherly J
RP_TC_004	UI	Result Page	Check if the other predictions are displayed properly	1) Open the page 2) Click on select button 3) Select the input image 4) Check if all the other predictions are displayed	1.png	The other predictions should be displayed properly	Working as expected	Pass		Swetha Lakshme S Roobinee R

8.2 User Acceptance Testing

User Acceptance Testing (UAT) is a type of testing performed by the end user or the client to verify/accept the software system before moving the software application to the production environment. UAT is done in the final phase of testing after functional, integration and system testing is done. The main Purpose of UAT is to validate end to end business flow. It does not focus on cosmetic errors, spelling mistakes or system testing. User Acceptance Testing is carried out in a separate testing environment with production-like data setup. It is a kind of black box testing where two or more end-users will be involved. Need of User Acceptance Testing arises once software has undergone Unit, Integration and System testing because developers might have built software based on requirements document by their own understanding and further required changes during development may not be effectively communicated to them, so for testing whether the final product is accepted by client/end-user, user acceptance testing is needed.

Defect Analysis

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

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	1	0	1	1	3
Duplicate	1	0	0	0	1
External	0	0	2	0	2
Fixed	4	1	0	1	6
Not Reproduced	0	0	1	1	2
Skipped	0	0	1	1	2

Won't Fix	1	0	1	0	2
Total	7	1	6	4	18

Test Case Analysis

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

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

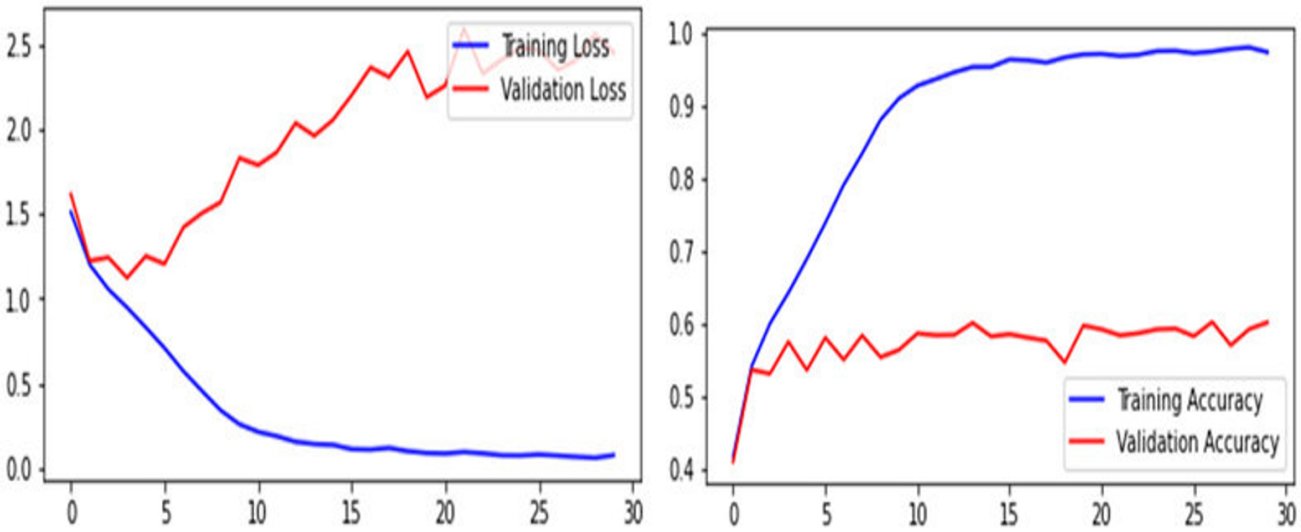
9. RESULTS

9.1 Performance Metrics

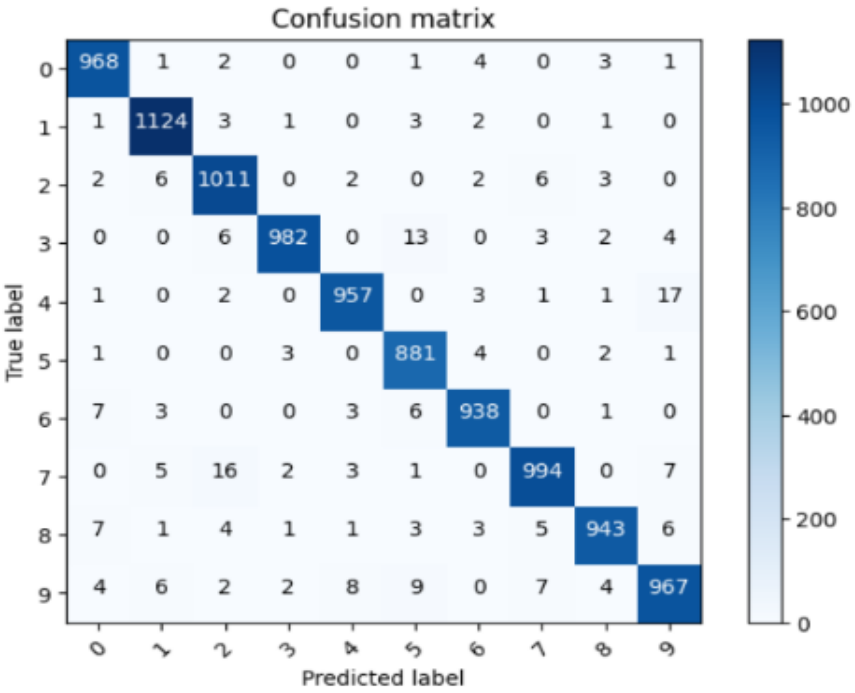
Model Summary

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 26, 26, 64)        640

 conv2d_1 (Conv2D)           (None, 24, 24, 32)        18464

 flatten (Flatten)           (None, 18432)             0

 dense (Dense)               (None, 10)                184330

=================================================================
Total params: 203,434
Trainable params: 203,434
Non-trainable params: 0
_________________________________________________________________
```

Accuracy



Confusion Matrix



Classification Report

	precision	recall	f1-score	support
0	1.00	0.97	0.98	980
1	0.99	0.99	0.99	1135
2	0.96	0.99	0.97	1032
3	0.97	1.00	0.98	1010
4	1.00	0.95	0.98	982
5	0.96	1.00	0.98	892
6	0.99	0.96	0.97	958
7	0.99	0.98	0.99	1028
8	0.99	0.99	0.99	974
9	0.97	0.99	0.98	1009
accuracy			0.98	10000
macro avg	0.98	0.98	0.98	10000
weighted avg	0.98	0.98	0.98	10000

10. ADVANTAGES & DISADVANTAGES

Advantages

1. Better readability of digits from different handwriting
2. Used in various sectors like hospitals, banking, post office, sectors involving data entries.
3. 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;
4. The generative models can perform recognition driven segmentation

Disadvantages

1. It requires much more computation than more standard OCR techniques.
2. It is not done in real time as a person writes and therefore not appropriate for immediate text input.

11. CONCLUSION

There are numerous uses for handwritten digit recognition in the fields of medicine, banking, student administration, taxation, etc. To extract the digit from the handwritten image, a variety of classifiers including KNN, SVM, and CNN are employed. According to the evaluation, CNN performs better than the competition. This study discusses the stages of HDR using a CNN classifier. The MNIST dataset is a common dataset used to evaluate the performance of classifiers. It consists of handwritten numbers from 0 to 9. Three separate stages make up HDR. The

first step is preprocessing, which involves converting the dataset into binary format and applying image processing on it. Segmentation, the second stage, involves dividing the image into several pieces. The third stage is feature extraction, during which image features are found. CNN is utilised in the classification stage, which comes last. The CNN classifier greatly enhances the outcomes of HDR, but it is still possible to further enhance the complexity, execution time, and accuracy of the results by combining classifiers or utilising other algorithms in addition to CNN.

12. FUTURE SCOPE

There is still much work to be done on this project, and it may use a lot of improvement. The following are a few ways this project could be improved:

- Add the ability to save the results of multiple image detection from digits.
- Adding capability to recognise multiple digits
- To detect numbers from complicated images, improve the model.
- Adding support for additional languages will benefit users worldwide.

This undertaking has limitless potential and may constantly be improved. By putting this idea into practise in the real world, numerous sectors will gain, many workers' workloads will be reduced, and overall work efficiency will increase.

13. APPENDIX

SOURCE CODE

1. 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 = 'Static/uploads'
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)
```

2. index.html

```
<html>
<head>
  <title>Handwritten Digit Recognition</title>
  <meta name="viewport" content="width=device-width">
  <!-- GoogleFont -->
        <link      href="https://fonts.googleapis.com/css2?family=Prompt:wght@600&display=swap"
rel="stylesheet">
  <link href="https://fonts.googleapis.com/css2?family=Varela+Round&display=swap" rel="stylesheet">
<linkhref="https://fonts.googleapis.com/css2?family=Source+Code+Pro:wght@500&display=swap"
rel="stylesheet">
```

```
                                                                                <link
href="https://fonts.googleapis.com/css?family=Calistoga|Josefin+Sans:400,700|Pacifico&display=swap"
rel="stylesheet">
  <!-- bootstrap -->
      <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">
  <link rel="stylesheet" type="text/css" href="{{ url_for('static',filename='css/style.css') }}">
  <!-- fontawesome -->
  <script src="https://kit.fontawesome.com/b3aed9cb07.js" crossorigin="anonymous"></script>

                                          <script          src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
crossorigin="anonymous"></script>
                    <script      src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"
integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1"
crossorigin="anonymous"></script>
                        <script      src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"
integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM"
crossorigin="anonymous"></script>
  <script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@latest"></script>
  <style>
    body {
      background-image: url('static/images/index.png');
      background-repeat: no-repeat;
      background-size: cover;
    }

    img,
    input {
      color: #fff;
    }
  </style>
</head>
```

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

   $(document).ready(function() {
      $('#clear_button').on('click', function() {
         $('#image').val('');
         $('#frame').attr('src', "");
      });
   });
</script>
<body>
   <nav class="navbar navbar-expand-lg navbar-light bg-dark">
      <a class="navbar-brand text-white">IBM PROJECT </a>
            <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav"
aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse justify-content-end" id="navbarNav">
         <ul class="navbar-nav">
            <li class="nav-item active">
                              <a class="nav-link text-white" href="#">PID - PNT2022TMID03914 <span
class="sr-only">(current)</span></a>
            </li>
         </ul>
      </div>
   </nav>
   <section id="title">
      <h2 class="heading text-white">HANDWRITTEN DIGIT RECOGNITION</h2><br><br>
          <p class="text-white">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. The dataset consist of 60,000 training images and 10,000 test images. The artificial neural networks can all most mimic the human brain and are a k

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.

```
</section>
<section id="content">
  <div class="leftside">
    <form action="/predict" method="POST" enctype="multipart/form-data">
      <label class="text-white">Select a image:</label>
                    <input id="image" type="file" name="image" accept="image/png, image/jpeg" onchange="preview()"><br><br>
      <img id="frame" src="" width="100px" height="100px" alt="Input Image" />
      <div class="buttons_div">
        <button type="submit" class="btn" id="predict_button">Recognize</button>
        <button type="button" class="btn" id="clear_button">  Clear  </button>
      </div>
    </form>
  </div>
</section>
</body>
</html>
```

3. predict.html

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

```
    <title>Prediction</title>
</head>

<style>
    body {
        background-image: url('static/images/bg.jpg');
        background-repeat: no-repeat;
        background-size: cover;
    }
    #rectangle {
        width: 600px;
        height: 120px;
        background-color: #d51c1c;
        border-radius: 25px;
        position: absolute;
        top: 45%;
        left: 50%;
        transform: translate(-50%, -50%);
    }
    #ans {
        text-align: center;
        font-size: 40px;
        margin: 0 auto;
        padding: 3% 5%;
        padding-top: 6%;
        color: white;
    }
</style>
<body>
    <div id="rectangle">
        <h1 id="ans">Recognized Number : {{num}}</h1>
    </div>
</body>
</html>
```

4. **style.css**

```
#clear_button {
    margin-left: 15px;
    font-weight: bold;
    color: blue;
}
#confidence {
    font-family: 'Josefin Sans', sans-serif;
    margin-top: 7.5%;
}
#content {
    margin: 0 auto;
    padding: 2% 15%;
    padding-bottom: 0;
}
.welcome {
    text-align: center;
    position: relative;
    color: honeydew;
    background-color: greenyellow;
    padding-top: 1%;
    padding-bottom: 1%;
    font-weight: bold;
    font-family: 'Prompt', sans-serif;
}
#team_id {
    text-align: right;
    font-size: 25px;
    padding-right: 3%;
}
#predict_button {
    margin-right: 15px;
```

```
    color: rgb(255, 0, 21);
    font-weight: bold;
}
#prediction_heading {
    font-family: 'Josefin Sans', sans-serif;
    margin-top: 7.5%;
}
#result {
    font-size: 5rem;
}
#title {
    padding: 1.5% 15%;
    margin: 0 auto;
    text-align: center;
}
.btn {
    font-size: 15px;
    padding: 10px;
    -webkit-appearance: none;
    background: #eee;
    border: 1px solid #888;
    margin-top: 20px;
    margin-bottom: 20px;
}
.buttons_div {
    margin-bottom: 30px;
    margin-right: 80px;
}
.heading {
    font-family: 'Varela Round', sans-serif;
    font-weight: 700;
    font-size: 2rem;
    display: inline;
}
```

```
.leftside {
    text-align: center;
    margin: 0 auto;
    margin-top: 2%;
}
#frame {
    margin-right: 10%;
}
.predicted_answer {
    text-align: center;
    margin: 0 auto;
    padding: 3% 5%;
    padding-top: 0;
    /* padding-left: 10%; */
}
p {
    font-family: 'Source Code Pro', monospace, sans-serif;
    margin-top: 1%;
}
@media (min-width: 720px) {
    .leftside {
        padding-left: 10%;
    }
}
```

GitHub & Project Demo Link

GitHub Link

https://github.com/IBM-EPBL/IBM-Project-25103-1659953734

Project Demo Link

https://youtu.be/dS9liYzieCI