A Novel Method for Handwritten Digit Recognition System

PROJECT REPORT

Submitted by

Abinesh S (950019106001)

Harini K(950019106010)

Jubin SJ(950019106018)

Ramasubramaniyan M (950019106303)

TEAM ID :PNT2022TMID49604

DEPARTMENT OF

ELECTRONICS AND COMMUNICATION

ENGINEERING

ANNA UNIVERSITY REGIONAL CAMPUS
TIRUNELVELI

SI.NO 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 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. Artificial neural networks is used 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.

1.2 PURPOSE

- To convert handwritten digits into machine readable formats
- 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
- The recognition systems can recognize familiar patterns quickly and accurately.

2.LITERATURE SURVEY

2.1 EXISTING PROBLEM

The problem that cursive handwritten can cause is that letters may not be as easily recognised and possibly cause false and incorrect information being processed

2.2 REFERNCES

YEAR	AUTHOR &TITLE	TECHNIQUE
2019	Beatrice Lopez et al & Modified MNIST	ResNet implementation

- Their overall approach was to implement their own CNNs from scratch.
- That involves pre-processing the data, image classification and training various network.
- Their ResNet implementation achieved an accuracy of 97.553% on the Kaggle leaderboard.
- https://nminnie.github.io/pdf/MNIST-recognition.pdf

YEAR	**AUTHOR & TITLE**	**TECHNIQUE**
2020	Hui-huang Zhao et al & Multiple classifiers fusion and CNN feature extraction for handwritten digits recognition	CNN with MNIST dataset

- They proposed a framework that involves CNN-base feature extraction from the MNIST data set and algebraic fusion of multiple classifiers trained an different feature sets which are prepared through feature selection applied to the original feature set extracted using CNN.
- The experimental results show that the classifiers fusion can achieve the classification accuracy of ≥98%.
- https://link.springer.com/article/10.1007/s41066-019-00158-6

YEAR	**AUTHOR & TITLE**	**TECHNIQUE**
2021	Khedidija Derdour et al & Multiple Features Extraction and Classifiers Combination Based Handwriting Digit Recognition	By using three methods of classifiers combination rule employ in MNIST database

- They proposed a system for handwriting digit recognition using different invariant features extraction and multiple classifiers.
- To achieve the best possible classification performance in terms of recognition rate, three methods of classifiers Combination rule employed: majority vote, Borda count and maximum rule.

- Experiments are performed on the well-known MNIST database of handwritten digits.
- https://www.proquest.com/openview/72acbcf9bceca23bc38c7d1df8734acb/1?pq-origsite=gscholar&cbl=316223

YEAR	AUTHOR &TITLE	TECHNIQUE
2021	Ali Abdullah Yahya et al A Novel Handwritten Digit Classification System Based on Convolutional Neural Network Approach	CNN ,MNIST Dataset with addictive white Gaussian noise

- They presented a novel convolutional neural network architecture based on data preparation, receptive field, data augmentation, optimization, normalization, and regularization techniques for handwritten digit recognition.
- They propose to add an additive white Gaussian noise with σ = 0.5 to the MNIST dataset.
- As a result, their 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.
- https://www.mdpi.com/1424-8220/21/18/6273

YEAR	**AUTHOR & TITLE**	**TECHNIQUE**
2022	Aby Motty et al SMART DRAGOMAN: Handwritten Digit Recognition and Text to Speech Translation	Keras library which contains some dataset

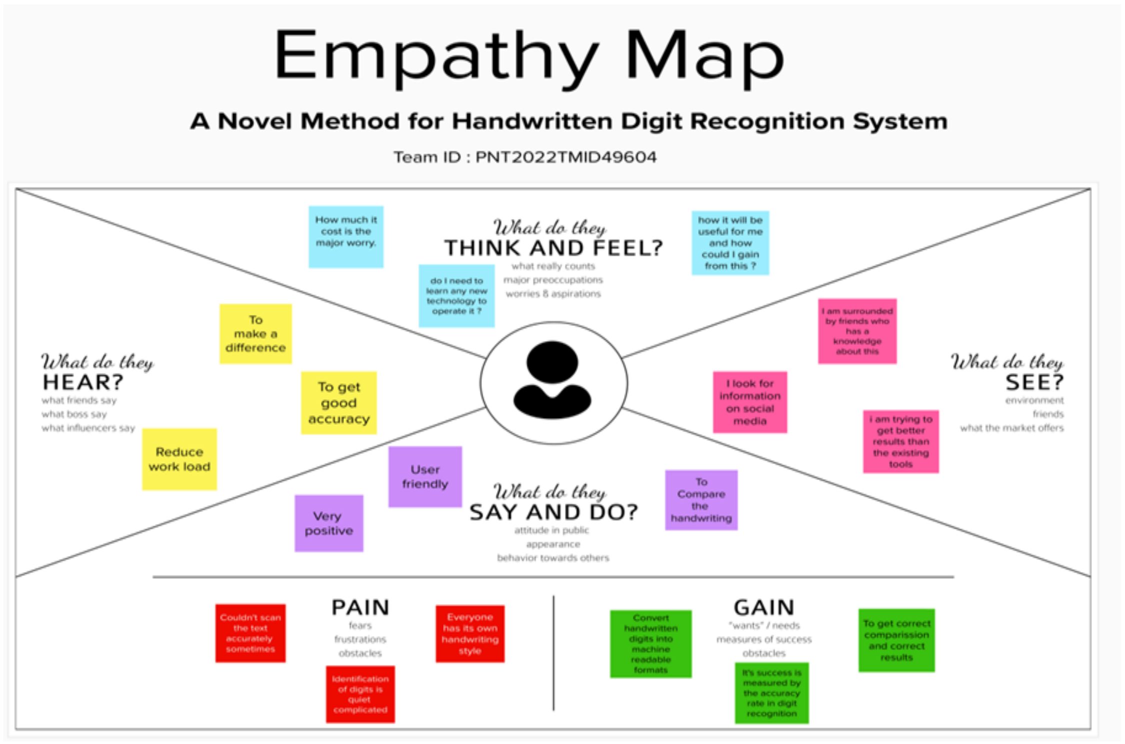
- They import all the modules that needed to train model. They use Keras library which contains some datasets and MNIST is one of them.
- In Smart Dragoman, using MNIST dataset for the purpose of digit recognition they have achieved better performance for the CNN.
- In their experiment, they have found the maximum training accuracy100% and maximum validation accuracy 99.92% both at epoch 15.
- The overall performance of the network is found99.21% and the overall loss ranged from 0.026303 to 0.049449.
- https://www.ijres.org/papers/Volume-10/Issue-7/10078084.pdf

2.3 PROBELM STATEMENT DEFINITION

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

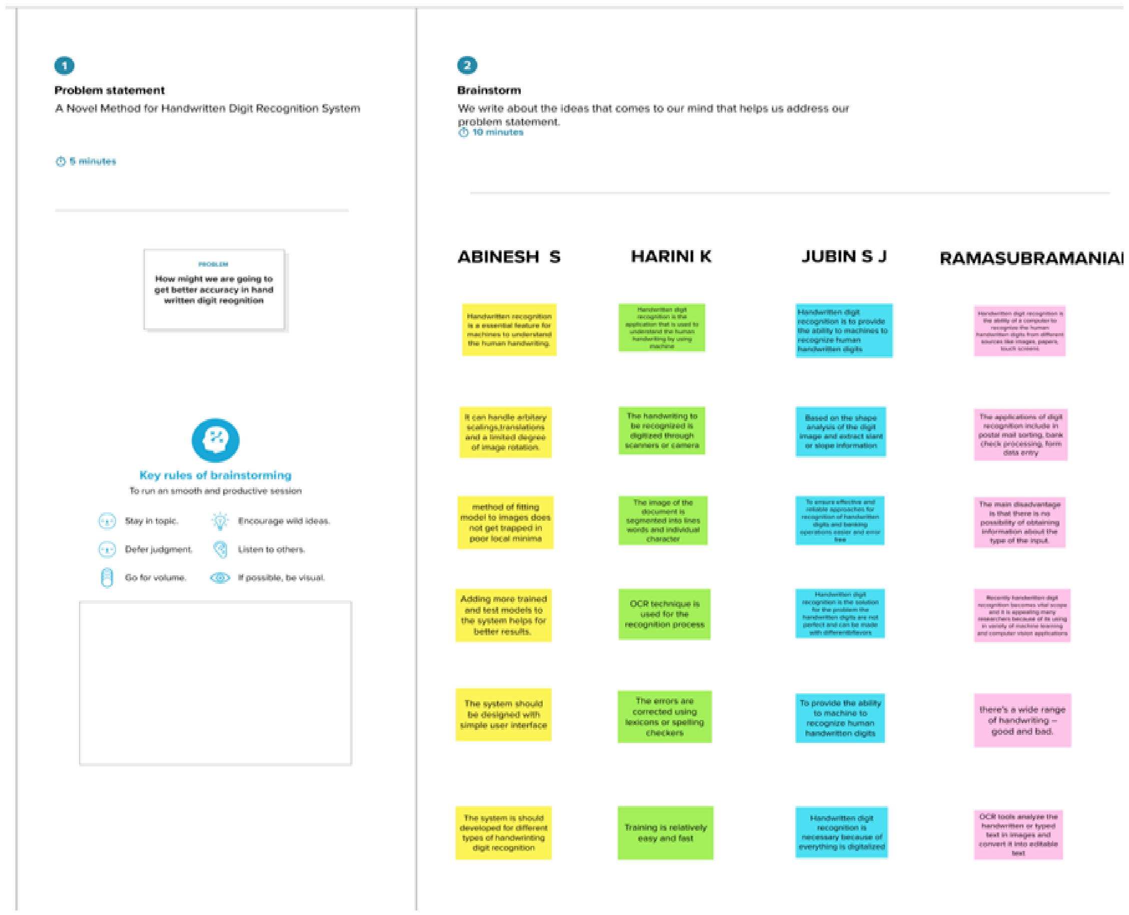
3 IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS



Empathy Map
A Novel Method for Handwritten Digit Recognition System
Team ID : PNT2022TMID49604
What do they THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations
How much it cost is the major worry.
do I need to learn any new technology to operate it ?
how it will be useful for me and how could I gain from this ?
What do they HEAR?
what friends say
what boss say
what influencers say
To make a difference
To get good accuracy
Reduce work load
What do they SEE?
environment
friends
what the market offers
I am surrounded by friends who has a knowledge about this
I look for information on social media
i am trying to get better results than the existing tools
What do they SAY AND DO?
attitude in public
appearance
behavior towards others
User friendly
Very positive
To Compare the handwriting
PAIN
fears
frustrations
obstacles
Couldn't scan the text accurately sometimes
Identification of digits is quiet complicated
Everyone has its own handwriting style
GAIN
"wants" / needs
measures of success
obstacles
Convert handwritten digits into machine readable formats
It's success is measured by the accuracy rate in digit recognition
To get correct comparission and correct results

3.2 IDEATION & BRAINSTORMING



1

Problem statement

A Novel Method for Handwritten Digit Recognition System

5 minutes

PROBLEM

How might we are going to get better accuracy in hand written digit reognition

Key rules of brainstorming

To run an smooth and productive session

Stay in topic. Encourage wild ideas.

Defer judgment. Listen to others.

Go for volume. If possible, be visual.

2

Brainstorm

We write about the ideas that comes to our mind that helps us address our problem statement.

10 minutes

ABINESH S

- Handwritten recognition is a essential feature for machines to understand the human handwriting.
- It can handle arbitary scalings,translations and a limited degree of image rotation.
- method of fitting model to images does not get trapped in poor local minima
- Adding more trained and test models to the system helps for better results.
- The system should be designed with simple user interface
- The system is should developed for different types of handwrinting digit recognition

HARINI K

- Handwritten digit recognition is the application that is used to understand the human handwriting by using machine
- The handwriting to be recognized is digitized through scanners or camera
- The image of the document is segmented into lines words and individual character
- OCR technique is used for the recognition process
- The errors are corrected using lexicons or spelling checkers
- Training is relatively easy and fast

JUBIN S J

- Handwritten digit recognition is to provide the ability to machines to recognize human handwritten digits
- Based on the shape analysis of the digit image and extract slant or slope information
- To ensure effective and reliable approaches for recognition of handwritten digits and banking operations easier and error free
- Handwritten digit recognition is the solution for the problem the handwritten digits are not perfect and can be made with differentdifferent
- To provide the ability to machine to recognize human handwritten digits
- Handwritten digit recognition is necessary because of everything is digitalized

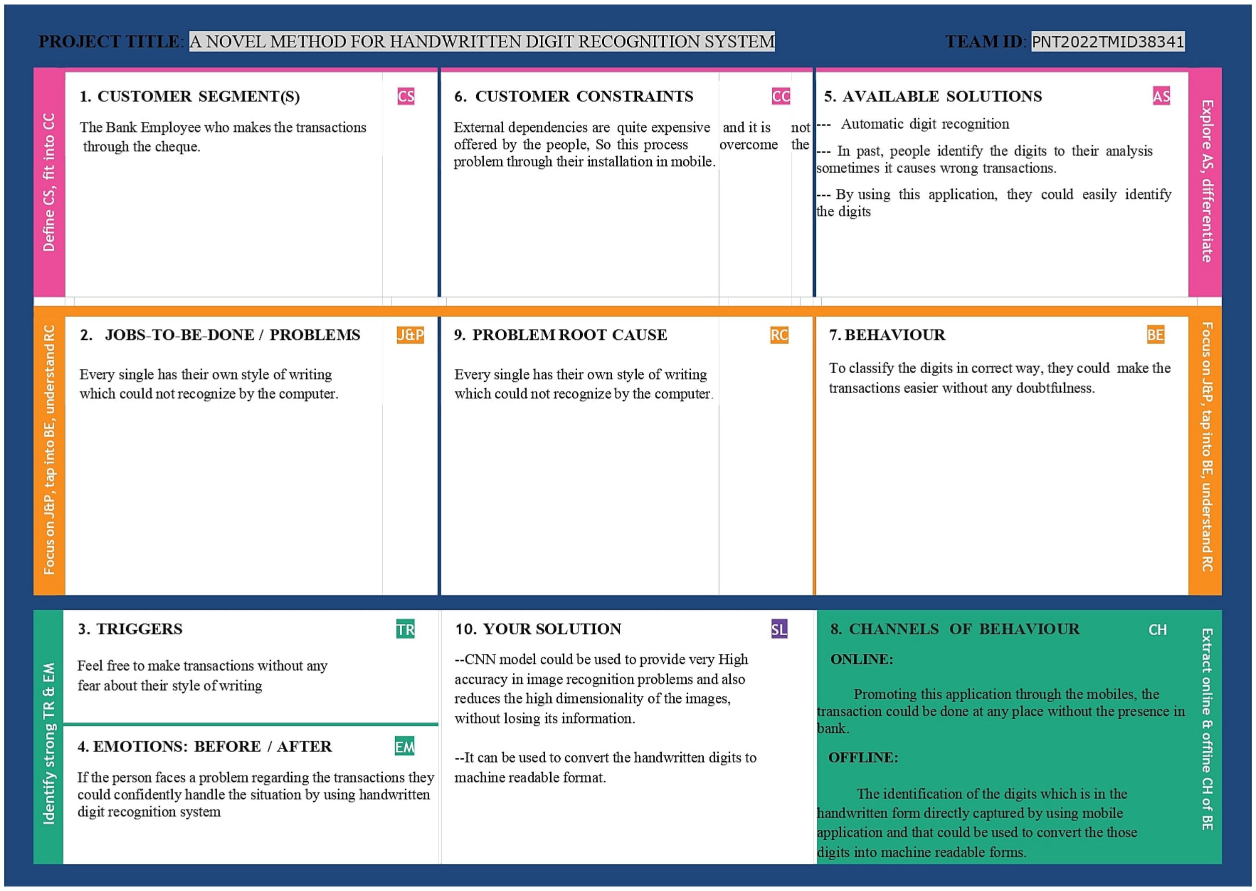
RAMASUBRAMANIAI

- Handwritten digit recognition is the ability of a computer to recognize the human handwritten digits from different sources like images, papers, touch screens
- The applications of digit recognition include in postal mail sorting, bank check processing, form data entry
- The main disadvantage is that there is no possibility of obtaining information about the type of the input.
- Recently handwritten digit recognition becomes vital scope and it is appealing many researchers because of its using in variety of machine learning and computer vision applications
- there's a wide range of handwriting – good and bad.
- OCR tools analyze the handwritten or typed text in images and convert it into editable text

3.3 PROPOSED SOLUTION

S.No	Parameter	Description
1.	Problem Statement (Problem to be solved)	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.
2.	Idea / Solution description	The application is the capability of the computer to identify and understand the human handwritten digits automatically instead of typing every time.
3.	Novelty / Uniqueness	If any unidentified handwritten digits are uploaded, it shows similar matched digits from the database and ask confirmation from the user.
4.	Social Impact / Customer Satisfaction	Handwritten digit recognition has recently gained importance, and it is attracting many researchers due to its usage in a number of machine learning and computer vision applications.
5.	Business Model (Revenue Model)	Number recognition has numerous operations like number plate recognition, postal correspondence sorting, bank check processing.
6.	Scalability of the Solution	The model that would be able to recognize and determine the handwritten digits from its image, and give better accuracy.

3.4 PROBLEM SOLUTION FIT



PROJECT TITLE: A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

TEAM ID: PNT2022TMID38341

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

The Bank Employee who makes the transactions through the cheque.

6. CUSTOMER CONSTRAINTS CC

External dependencies are quite expensive and it is not offered by the people, So this process overcome the problem through their installation in mobile.

5. AVAILABLE SOLUTIONS AS

--- Automatic digit recognition

--- In past, people identify the digits to their analysis sometimes it causes wrong transactions.

--- By using this application, they could easily identify the digits

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

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

Every single has their own style of writing which could not recognize by the computer.

9. PROBLEM ROOT CAUSE RC

Every single has their own style of writing which could not recognize by the computer.

7. BEHAVIOUR BE

To classify the digits in correct way, they could make the transactions easier without any doubtfulness.

Focus on J&P, tap into BE, understand RC

Identify strong TR & EM

3. TRIGGERS TR

Feel free to make transactions without any fear about their style of writing

4. EMOTIONS: BEFORE / AFTER EM

If the person faces a problem regarding the transactions they could confidently handle the situation by using handwritten digit recognition system

10. YOUR SOLUTION SL

--CNN model could be used to provide very High accuracy in image recognition problems and also reduces the high dimensionality of the images, without losing its information.

--It can be used to convert the handwritten digits to machine readable format.

8. CHANNELS OF BEHAVIOUR CH

ONLINE:

Promoting this application through the mobiles, the transaction could be done at any place without the presence in bank.

OFFLINE:

The identification of the digits which is in the handwritten form directly captured by using mobile application and that could be used to convert the those digits into machine readable forms.

Extract online & offline CH of BE

4. REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

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 thenutilised to compare thedata and provide an output in digitalized form.
FR-2	Recognizing thehandwrittendigit and displaying.	Recognizing the handwritten digitand displaying.
FR-3	Import dataset file directly to theprogram 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 withanumber 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

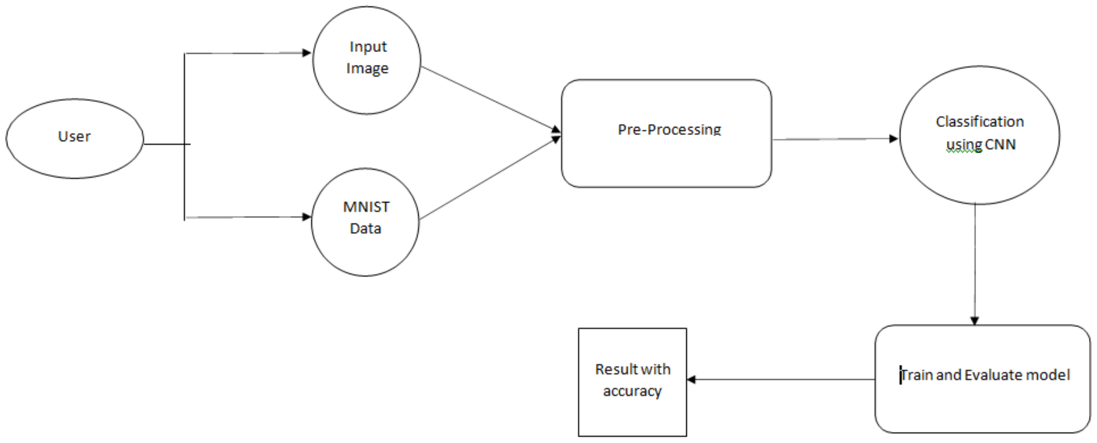
4.2 NON-FUNCTIONAL REQUIREMENTS

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 ableto authenticate all users with their unique username andpassword
NFR-3	**Performance**	Should reducethe delay in information when hundreds of requests are given.
NFR-4	**Availability**	Information is restricted to each userslimited access
NFR-5	**Scalability**	the system should be able to handle 10000 users accessing the site at the same time

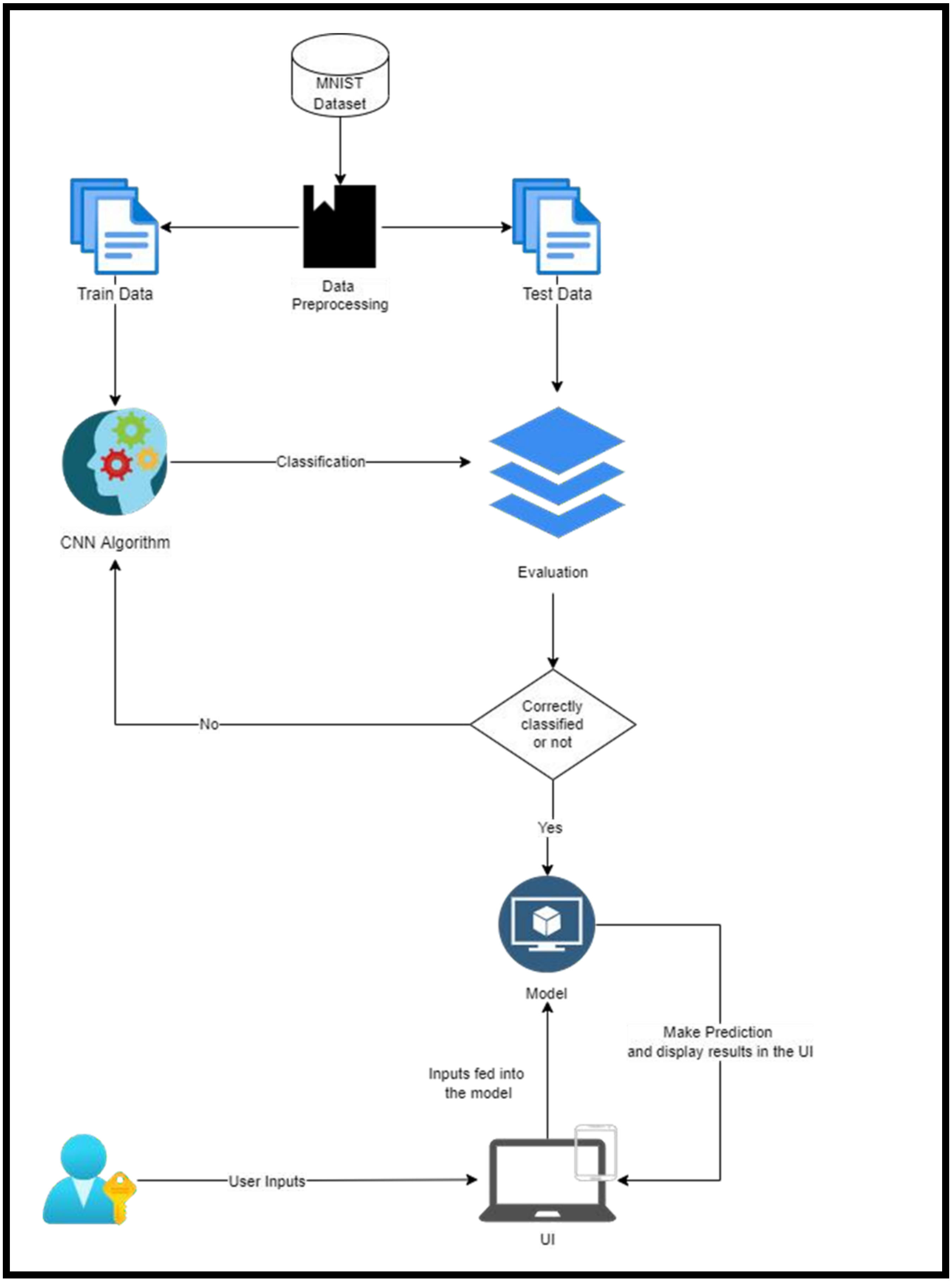
5. PROJECT DESIGN

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 amountof the system requirement graphically. It shows howdata enters and leavesthe system, what changes the information, and where data is stored.



5.2 SOLUTION & TECHNICAL ARCHITECTURE


MNIST Dataset
Train Data
Data Preprocessing
Test Data
Classification
CNN Algorithm
Evaluation
Correctly classified or not
No
Yes
Model
Make Prediction and display results in the UI
Inputs fed into the model
User Inputs
UI

5.3 USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer	Application	USN-1	As a user, I can application by opening it easily.	I can download the application	High	Sprint-1
		USN-2	As a user, I will be given access to the canvas board to draw or write the number	I can access thecanvas	High	Sprint-1
		USN-3	As a user, I can change the colour of the pen ink.	I can use the canvas pen	Medium	Sprint-2

6. PROJECT PLANNING AND SCHEDULING

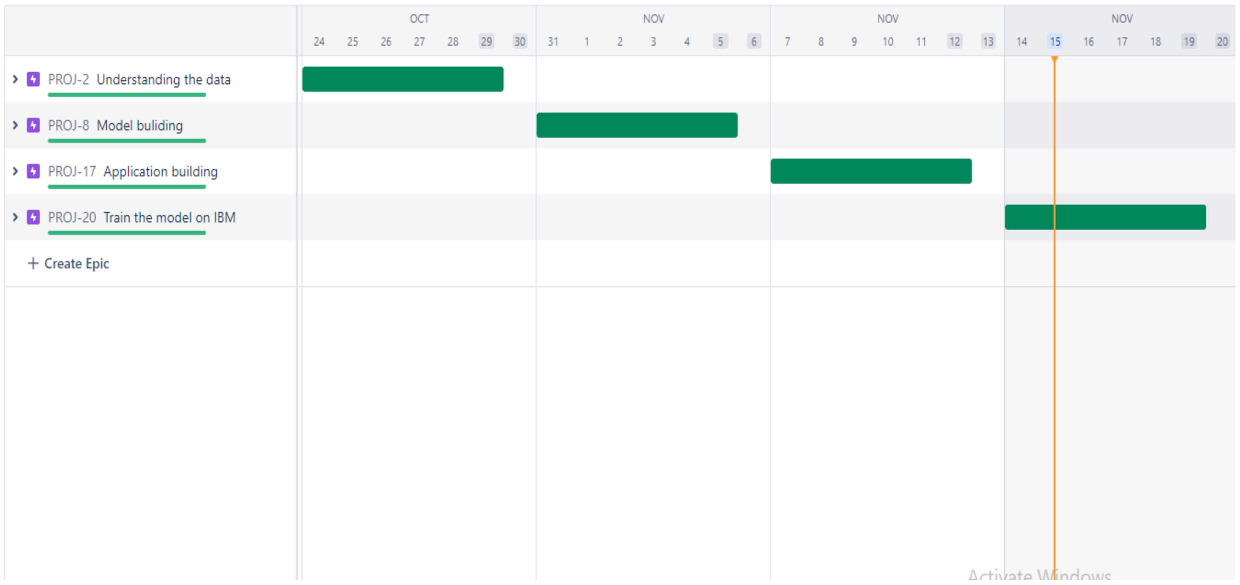
6.1 SPRINT PLANNING AND ESTIMATION

Sprint	Functional Requirement (Epic)	User Story Number	UserStory / Task	Story Points	Priority	TeamMembers
Sprint-1	Understanding the data	USN-1	Analyze the data, Importing the required libraries, Loadthe data, One hot encoding, Understanding thedata.	2	High	Abinesh S Harini K Jubin SJ Ramasubramaniyam M
Sprint-2	Model Building	USN-2	Adding layers ,compiling the model, model building, observing the metrices, save the model, test the model, test with saved model, train the model.	2	High	Abinesh S Harini K Jubin SJ Ramasubramaniyam M
Sprint-3	Application building	USN-3	Create HTML pages. Link python files into HLML.	2	High	Abinesh S Harini K Jubin SJ Ramasubramaniyam M
Sprint-4	Train the model onIBM	USN-4	Train the final modelon IBM and observe the metrices.	2	High	Abinesh S Harini K Jubin SJ Ramasubramaniyam M

6.2 SPRINT DELIVERY SCHEDULE

Sprint	Total story points	Duration	Sprint start date	Sprint end date (planned)	Story plan 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	5 Nov 2022	20	5 Nov 2022
Sprint-3	20	6 Days	7 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 REPORT FROM JIRA



7.CODING & SOLUTION (Explain the features added in the project along with code)

7.1 FEATURE 1

1.Importing The Required Libraries

Importing the required libraries which are required for the model to run. The dataset for this model is imported from the Kerasmodule.The dataset contains ten classes: Digits from 0-9. Each digit is taken as a class.

2.Loading The Data

Then split the data into train and test. Using the training dataset, train the model and the testing dataset is used to predict the results.

Then find out the shape of X_train and x_test for better understanding. It lists out the dimensions of the data present in it.In the trainset consists of 60000 images, and in the test set consists 10000 images.

3.Analyzing The Data

Basically, the pixel values range from 0-255. Print the first image pixel value which is index[0] of the training data. As you see it is displayed in the output.

With respect to the image, the label of this image will be stored in y_train let's see what is the label of this image by grabbing it from the y_train variable.

After the pixel values are printed, find which image the pixel values belong to. The output will display, what is that image ?

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib library is used to displaying the number in the form of an image for proper understanding.

Plot the image on a graph using the Matplot library.

4.Add CNN Layers

Creating the model and adding the input, hidden, and output layers to it.The Sequential model is a linear stack of layers. The Sequential model is created by passing a list of layer instances to the constructor.

5.Compiling The Model

With both the training data defined and model defined, it's time to configure the learning process. This is accomplished with a call to the compile () method of the Sequential model class. Compilation requires 3 arguments: an optimizer, a loss function, and a list of metrics.

6.Train The Model

Now, let train the model with the image dataset.

6(a) Arguments:

steps_per_epoch : it specifies the total number of steps taken from the generator as soon as one epoch is finished and the next epoch has started. The value of steps_per_epoch is calculated as the total number of samples in your dataset divided by the batch size.

Epochs: an integer and number of epochs, want to train the model for.

6(b) Validation_data :

- an inputs and targets list
- a generator
- inputs, targets, and sample_weights list which can be used to evaluate the loss and metrics for any model after any epoch has ended.

validation_steps: only if the validation_data is a generator then only this argument can be used. It specifies the total number of steps taken from the generator before it is stopped at every epoch and its value is calculated as the total number of validation data points in your dataset divided by the validation batch size.

7.Observing The Metrics

Print the metrics which lists out the Test loss and Test accuracy

- Loss value implies how poorly or well a model behaves after each iteration of optimization.
- An accuracy metric is used to measure the algorithm's performance in an interpretable way.

8.Test The Model

Firstly slice the x_test data until the first four images. In the next step print the predicted output.

9.Save The Model

Your model is to be saved for future purposes. This saved model can also be integrated with an android application or web application in order to predict something.

The model is saved with .h5 extension as follows:

- An H5 file is a data file saved in the Hierarchical Data Format (HDF).
- It contains multidimensional arrays of scientific data.

10.Test With Saved Model

Firstly load the model which was built. Then apply for a loop for the first four images and converting the image to the required format. Then resize the input image, converting the image as per the CNN model and reshape it according to the requirement. At last, predict the result.

11.Application Building

Build a web application that is integrated into the model that was builted. A UI is provided for the uses where he has uploaded an image. The uploaded image is given to the saved model and prediction is showcased on the UI.

This section has the following tasks

- Building HTML Pages
- Building server-side script

7.2 FEATURE 2

1.Importing the required libraries

```
import numpy as np
import tensorflow #open source used for both ML and DL for computation
from tensorflow.keras.datasets import mnist #mnist dataset
from tensorflow.keras.models import Sequential #it is a plain stack of layers
 from tensorflow.keras import layers #A Layer consists of a tensor- in tensor-out computat ion
function
 from tensorflow.keras.layers import Dense, Flatten #Dense-Dense Layer is the regular deeply
connected r
#faltten -used fot flattening the input or change the dimension
from tensorflow.keras.layers import Conv2D #onvoLutiona l Layer
from keras.optimizers import Adam #opt imizer
from keras. utils import np_utils #used for one-hot encoding
```

```
import matplotlib.pyplot as plt   #used for data visualization
```

2.Load data

```
(x_train, y_train), (x_test, y_test)=mnist.load_data () #splitting the mnist data into train and test
```

```
print (x_train.shape)  #shape is used for give the dimens ion values #60000-rows 28x28-pixels
print (x_test.shape)
```

```
(60000, 28, 28)
(10000, 28, 28)
```

```
x_train[0]
```

```
array([[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
            0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
           0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   3,
         18,  18,  18, 126, 136, 175,  26, 166, 255, 247, 127,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,  30,  36,  94, 154, 170,
        253, 253, 253, 253, 253, 225, 172, 253, 242, 195,  64,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,  49, 238, 253, 253, 253, 253,
        253, 253, 253, 253, 251,  93,  82,  82,  56,  39,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,  18, 219, 253, 253, 253, 253,
        253, 198, 182, 247, 241,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,  80, 156, 107, 253, 253,
        205,  11,   0,  43, 154,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,  14,   1, 154, 253,
         90,   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, 139, 253,
          190,   2,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,  11, 190,
        253,  70,   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,  35,
        241, 225, 160, 108,   1,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
         81, 240, 253, 253, 119,  25,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,  45, 186, 253, 253, 150,  27,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,  16,  93, 252, 253, 187,   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, 249, 253, 249,  64,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,  46, 130, 183, 253, 253, 207,   2,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,  39,
        148, 229, 253, 253, 253, 250, 182,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,  24, 114, 221,
        253, 253, 253, 253, 201,  78,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,  23,  66, 213, 253, 253,
               253, 253, 198,  81,   2,   0,   0,   0,   0,   0,   0,   0,   0,
                 0,   0],
              [  0,   0,   0,   0,   0,   0,  18, 171, 219, 253, 253, 253, 253,
               195,  80,   9,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
                 0,   0],
              [  0,   0,   0,   0,  55, 172, 226, 253, 253, 253, 253, 244, 133,
                11,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
                 0,   0],
              [  0,   0,   0,   0, 136, 253, 253, 253, 212, 135, 132,  16,   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]], dtype=uint8)


          plt.imshow(x_train[5100])     #ploting the index=image


          <matplotlib.image.AxesImage at 0x1ca4f63ea60>
       []
        np.argmax(y_train[5100])
        0
```

3.Reshaping Dataset

```
#Reshaping to format which CNN expects (batch, height, width, channels)
x_train=x_train.reshape (60000, 28, 28, 1).astype('float32')
x_test=x_test.reshape (10000, 28, 28, 1).astype ('float32')
```

4.Applying One Hot Encoding

```
number_of_classes = 10  #storing the no of classes in a variable

y_train = np_utils.to_categorical (y_train, number_of_classes) #converts the output in binary format
y_test = np_utils.to_categorical (y_test, number_of_classes)
```

5.Add CNN Layers

```
#create model
model=Sequential ()

#adding modeL Layer
model.add(Conv2D(64, (3, 3), input_shape=(28, 28, 1), activation='relu'))
model.add(Conv2D(32, (3, 3), activation = 'relu'))

#flatten the dimension of the image
```

```
model.add(Flatten())


#output layer with 10 neurons
model.add(Dense(number_of_classes,activation = 'softmax'))
```

6.Compiling the model

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


x_train = np.asarray(x_train)
y_train = np.asarray(y_train)
```

7.Train the model

```
#fit the model
model.fit(x_train, y_train, validation_data=(x_test, y_test), epochs=5, batch_size=32)


Epoch 1/5
1875/1875 [==============================] - 137s 69ms/step - loss: 0.1989 - accuracy: 0.9543 - val_loss: 0.0839 - val_accuracy: 0.9761
Epoch 2/5
1875/1875 [==============================] - 136s 72ms/step - loss: 0.0614 - accuracy: 0.9811 - val_loss: 0.0764 - val_accuracy: 0.9789
Epoch 3/5
1875/1875 [==============================] - 115s 61ms/step - loss: 0.0448 -
```

```
accuracy: 0.9863 - val_loss: 0.0905 - val_accuracy: 0.9765
   Epoch 4/5
      1875/1875 [==============================] - 110s 58ms/step - loss: 0.0343 -
accuracy: 0.9896 - val_loss: 0.0978 - val_accuracy: 0.9753
   Epoch 5/5
      1875/1875 [==============================] - 117s 62ms/step - loss: 0.0286 -
accuracy: 0.9912 - val_loss: 0.0864 - val_accuracy: 0.9779

   <keras.callbacks.History at 0x2674c8c0820>
```

8.Observing the metrics

```
   # Final evaluation of the model
   metrics = model.evaluate(x_test, y_test, verbose=0)
   print("Metrics (Test loss &Test Accuracy) : ")
   print(metrics)
```

```
   Metrics (Test loss &Test Accuracy) :
   [0.08635331690311432, 0.9779000282287598]
```

```
   prediction=model.predict(x_test[6000:6001])
   print(prediction)
```

```
   1/1 [==============================] - 4s 4s/step
   [[3.3375277e-13 2.3497841e-14 2.2172753e-14 6.0079930e-07 1.6755180e-04
     2.7669451e-09 3.6617560e-13 9.0538032e-07 1.4666308e-07 9.9983084e-01]]
```

```
plt.imshow(x_test[5100])
```

```
<matplotlib.image.AxesImage at 0x2675ffabca0>
```

[]

```
import numpy as np
print(np.argmax(prediction, axis=1)) #printing our Labels from first 4 images
```

```
[9]
```

```
np.argmax(y_test[5100:5101]) #printing the actual labels
```

```
9
```

9.Save The model

```
# Save the model
model.save('models/mnistCNN.h5')
```

8. TESTING

8.1 TEST CASES

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

Section	Total Cases	Not Tested	Fail	Pass
Client Application	5	0	0	5
Security	5	0	0	5
Final Report Output	5	0	0	5
Performance	5	0	0	5

8.2 USER ACCEPTANCE TESTING

1. Purpose of Document

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

2. 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	5	2	1	0	8
Duplicate	5	1	0	1	7
External	1	1	1	1	4
Fixed	3	3	0	0	6
Not Reproduced	0	1	2	0	3
Skipped	3	1	1	2	7
Won't Fix	2	0	0	0	2

Totals	19	10	5	4	38

3. Test Case Analysis

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

Section	Total Cases	Not Tested	Fail	Pass
Client Application	5	0	0	5
Security	5	0	0	5
Final Report Output	5	0	0	5
Performance	5	0	0	5

9. RESULT

9.1 PERFORMANCE METRICS

S.NO	PARAMETER	VALUES
1.	Model Summary	Handwritten digit recognition has many direct applications, including recognizing license plates, zip codes for mail sorting, bank cheque amounts and numeric entries in tax forms. In research, it is used as an important benchmark for computer vision. For many handwritten digit recognition tasks, the original MNIST dataset1 is largely used for training various image processing systems. In this project, we worked with a modified version of the MNIST dataset to explore the applications of deep neural networks in computer vision and image analysis prediction. The Modified MNIST dataset consists of 50,000 grey-scale 64 x 64 pixel images with each containing more than one digit. For training and testing, 40,000 images are used as training set and 10,000 for testing set. This project sought to predict the digit which occupies the most space in each image i.e. the digit with the largest bounding box. Every image in the dataset comes with an associated label that identifies this digit, and the 4096 pixels that make up the image
2.	Accuracy	The maximum training accuracy-100% The maximum validation accuracy- 99.92% The overall performance of the network -99.21%

10. ADVANTAGE & DISADVANTAGE

ADVANTAGES

- 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
- The generative models can perform recognition driven segmentation
- The method involves a relatively small number of parameters and hence training is relatively easy and fast
- Unlike many other recognition schemes, it does not rely on some form of pre-normalization of input images, but can handle arbitrary scaling ,translations and a limited degree of image rotation.

DISADVANTAGE

- The Characters look very similar, making it hard for a computer to recognise accurately.
- Joined-up handwriting is another challenge for computers, when your letters all connect it makes it hard for computers to recognise individual characters.

11.CONCLUSION

We learned that the performance of a network depends not only on the architecture of the network, but also on the nature of the image data and what pre-processing tasks were applied to the data. Regarding runtime, we concluded that batch normalization significantly reduces training time, as CNN-7- BN trained much faster than CNN-6 and the one major difference between their architecture is the presence of batch normalization. We also noticed that generally, a larger batch size correlated with longer runtime and more computational power - but not necessarily higher accuracy.

12.FUTURE SCOPE

Recently handwritten digit recognition becomes vital scope and it is appealing many researchers because of its use in variety of machine learning and computer vision applications. Currently, the scope of our engine extends to recognizing one character at a time. We propose to extend this functionality to enable the accurate prediction of multiple characters simultaneously-thereby enabling truly real time character recognition.

13.APPENDIX

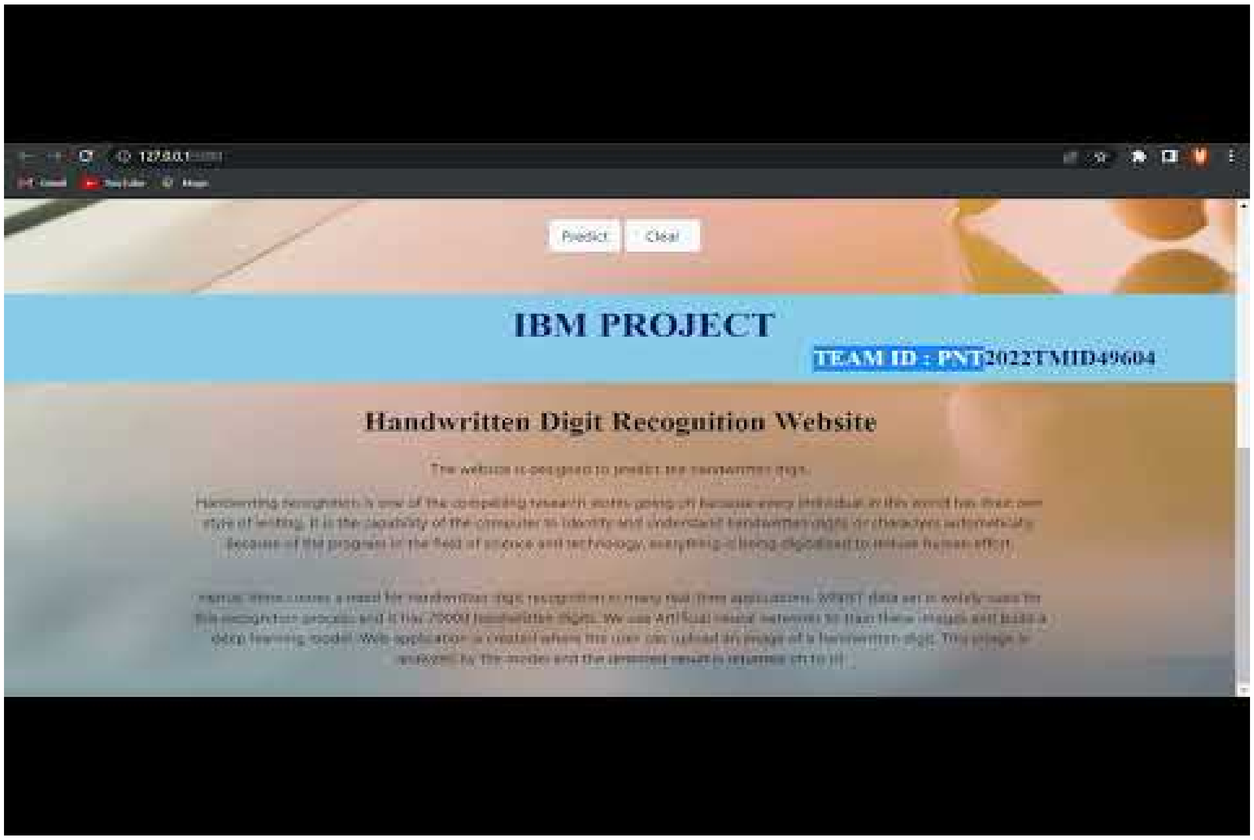
SOURCE CODE:

Our project source code link: https://colab.research.google.com/drive/1YfWiUmikDtY-7Ik2M0hHBlP_IXkqrEQ9#scrollTo=qMOYBE-8HAPy

Our Github link:: https://github.com/IBM-EPBL/IBM-Project-42591-1660669024

DEMO VIDEO:

Demo vedio link: https://youtu.be/sMeHehILgwE



IBM PROJECT

TEAM ID : PNT2022TMID49604

Handwritten Digit Recognition Website

The website is designed to predict the handwritten digit.

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.

[illegible] 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.