PROJECT REPORT

A NOVEL METHOD FOR HANDWRITTEN DIGITRECOGNITION SYSTEM

Submitted by :

PNT2022TMID46401

Team members	Register number
B Abimanyu	820319106001
M Dhinesh	820319106006
B Mahesh Kumar	820319106012
K Muhil	820319106014
S Udhaya	820319106023

TABLE OF CONTENTS

1.INTRODUCTION

1.1 Project overview

- The handwritten digit recognition is the ability of computers to recognize human handwritten digits. It is a hard task for the machine because handwritten digits are not perfect and can be made with many different flavors.
- 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

1.2 Purpose

- The task of handwritten digit recognition, using a classifier, has great importance and use such as online handwriting recognition on computer tablets, recognize zip codes on mail for postal ail sorting, processing bank

check amounts, numeric entries in forms filled up byhand (for example – tax forms) and so on.

- The applications of digit recognition include in postal mail sorting, bank check processing, form data entry, etc.

2.LITERATURE SURVEY

2.1Existing problem

- **Postal sector** - A postal sector worker trying to recognize the pin code and contact number in the postbut he is unable to recognize the numbers because thenumbers are joined up together which makes him feelconfused.

- **Banking sector** – A banking sector worker is trying to recognize the account number, money to be transacted and withdrawn, contact number and pin code of the address of the account holder but he is unable to recognize the numbers because there is huge variation inhandwritten digits form person to person and the numbers are joined up together which makes him feel unclear.
- **An old person** – a old person is trying to recognize thenumbers but he is unable to recognize the numbers because of aging and poor recognizing ability which makes him feel confused.

2.2 References

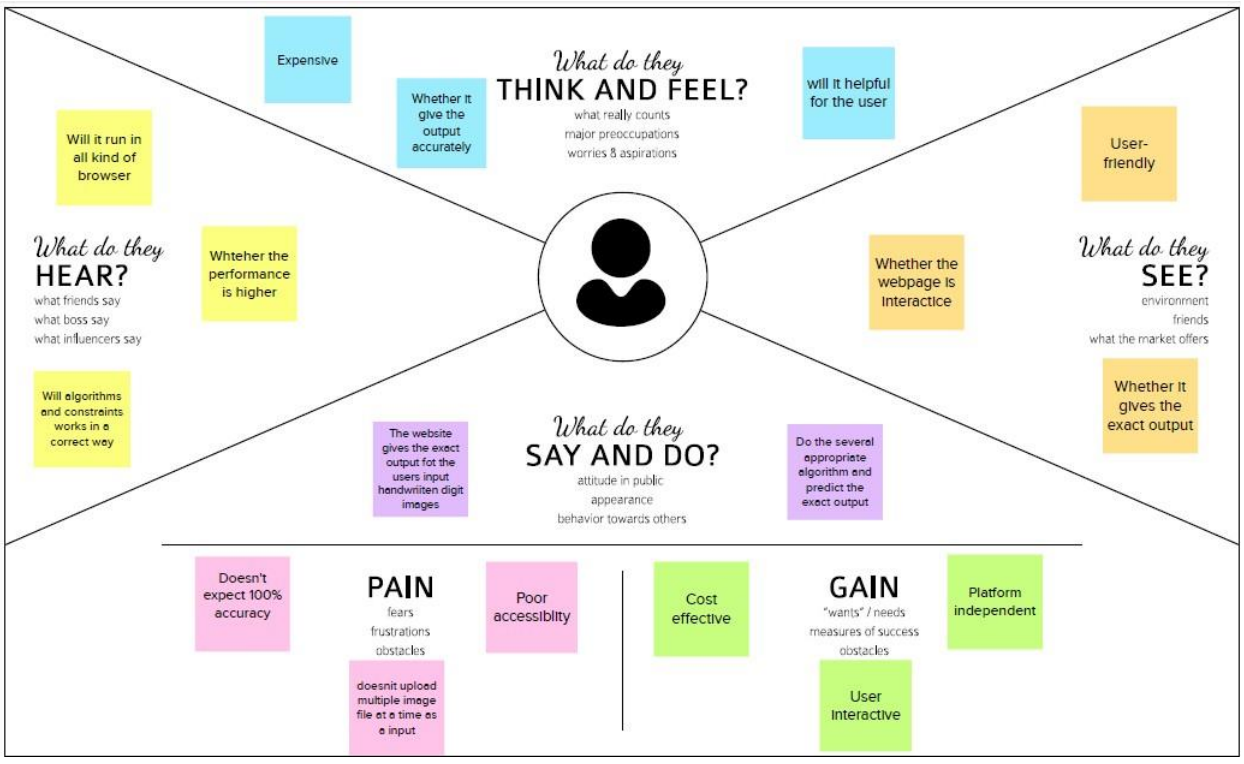
S.NO	PAPER TITLE	AUTHOR NAME	JOURNAL NAME YEAR OF PUBLICATION	REMARKS
1.	A novel method for combined feature execution for recognition.	Tingkai Sun, Songcan Chen	Face recognition, 2008	High processor required for the cost & time consuming.
2.	A novel method for of recognition of isolated handwritten Arabic character.	A Sahlol, C Suen	Novel processing, 2014	Accuracy rate is low to test and our own data can be required.
3.	A novel method for Persian handwritten digit recognition using support vector machine.	M. Mohammad poor, A. Mehdizadeh	Directing persion handwritten digits, 2018	Lack of accuracy due to absence of conduction networks with digits.
4.	Handwritten numeral recognition of	SV. Rajashekararadhya, PV. Ranjan	Handwritten character recognition,	Unable to identify the distorted data

	three popular south Indian script.		2008	that are determined in recognition system.
5.	A novel approach for handwritten Devanagari character recognition.	S. Arora, L Malik, D. Bhattacharjee	Method of recognition, 2010	Consumes more training time for handwritten recognition.

2.3 Problem statement definition

3. IDEATION &PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming

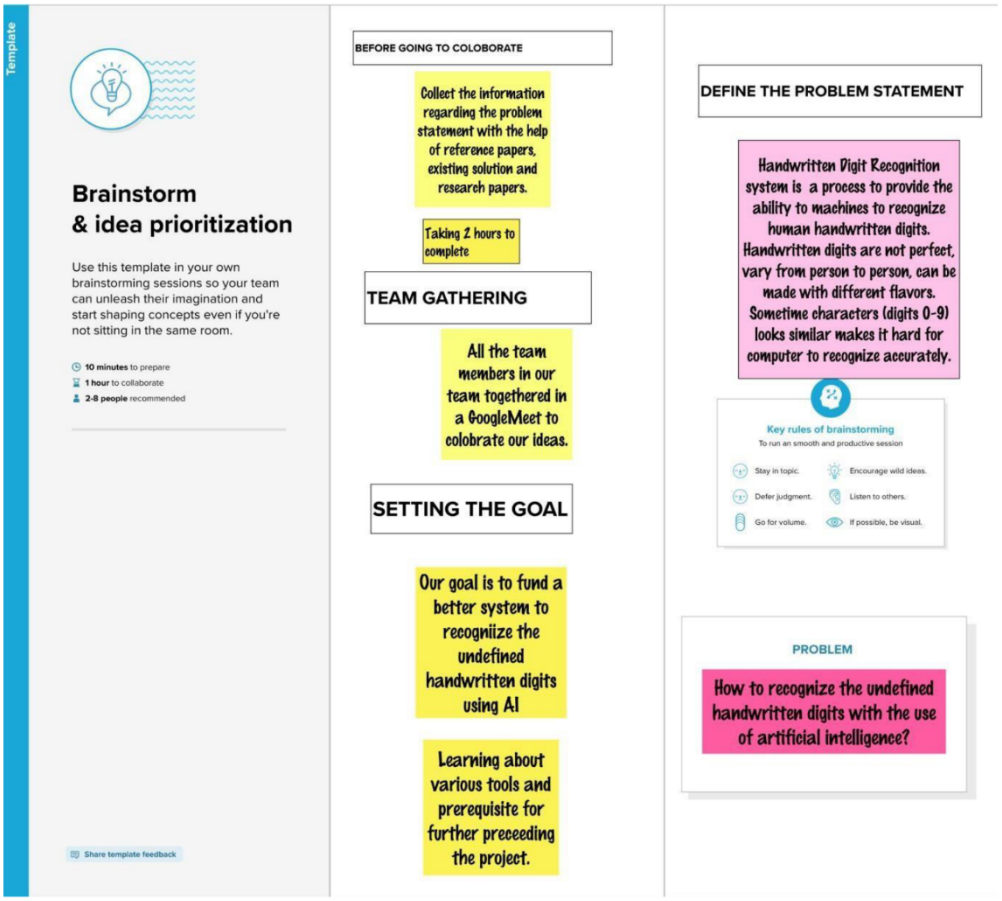
Brainstorm & Idea Prioritization :

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

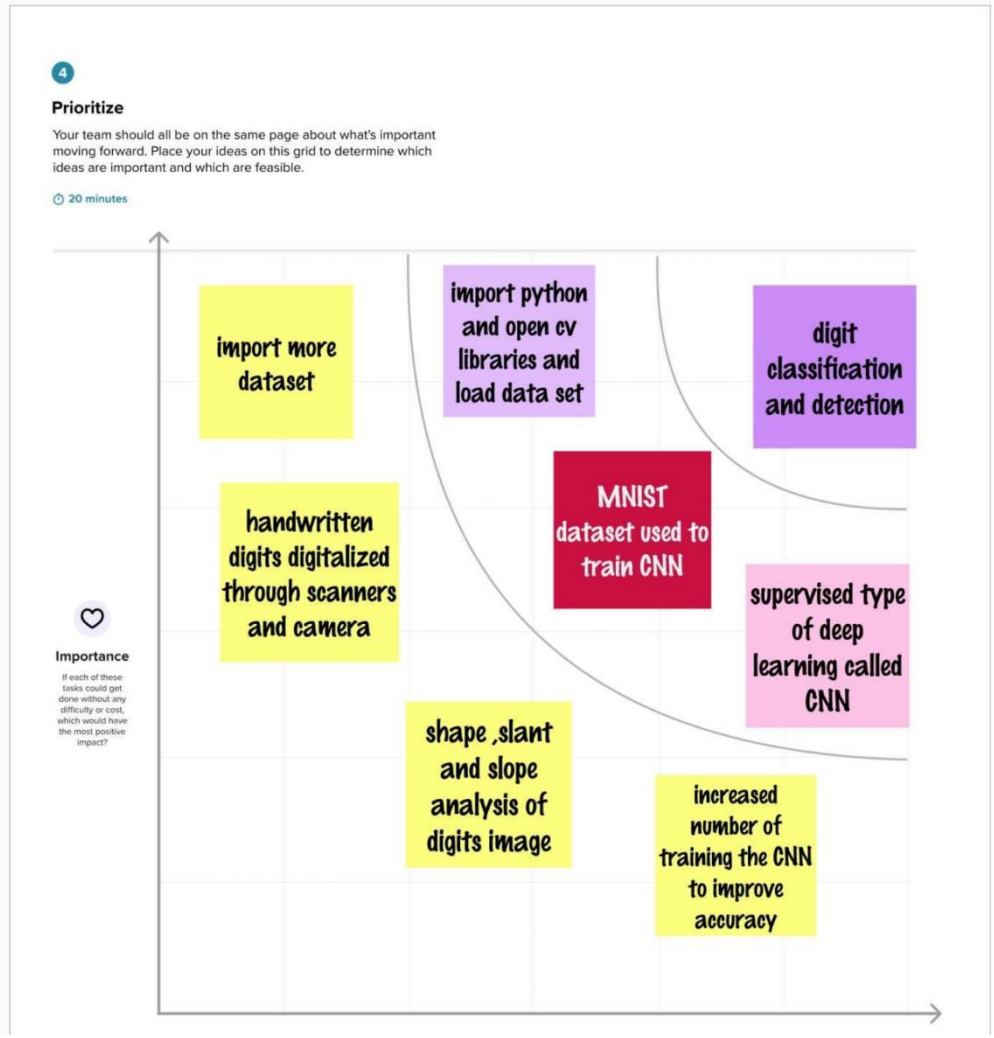
Reference: https://www.mural.co/templates/empathy-map-canvas

Step-1: Team Gathering, Collaboration and Select the Problem Statement



Step-2: Brainstorm, Idea Listing and Grouping

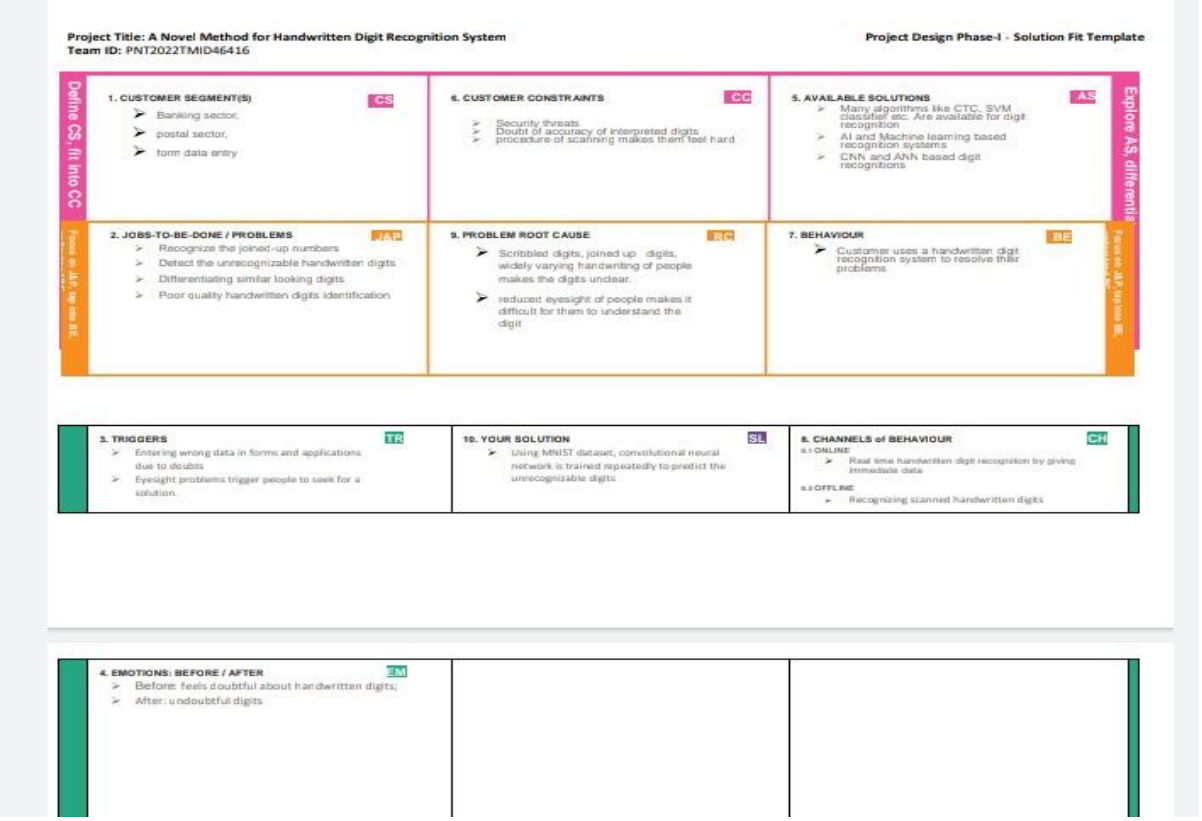
Step-3: Idea Prioritization



3.3 Proposed solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	A process to facilitate machines to recognize human handwritten digits .Handwritten digits are not perfect, it may vary from person to person and some characters looks similar which makes it hard for computer and human to recognize accurately.
2.	Idea / Solution description	MNIST dataset used to train a CNN to predict the given digit image by importing python and open CV libraries.
3.	Novelty / Uniqueness	Increased number of training set makes it fast, accurate and reliable method for digit recognition.
4.	Social Impact / Customer Satisfaction	Postal department and courier services can easily find the digit written. Old people with eyesight disabilities will be benefited.
5.	Business Model (Revenue Model)	Banking sector and postal sector services.
6.	Scalability of the Solution	Handles more number of data without compromising on performance and accuracy of result.

3.4 Problem solution fit



Project Title: A Novel Method for Handwritten Digit Recognition System
Team ID: PNT2022TMID46416

Project Design Phase-I - Solution Fit Template

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS
- Banking sector,
- postal sector,
- form data entry

6. CUSTOMER CONSTRAINTS CC
- Security threats
- Doubt of accuracy of interpreted digits
- procedure of scanning makes them feel hard

5. AVAILABLE SOLUTIONS AS
- Many algorithms like CTC, SVM classifier etc. Are available for digit recognition
- AI and Machine learning based recognition systems
- CNN and ANN based digit recognitions

Explore AS, differentia

Focus on J&P, tap into BE,

2. JOBS-TO-BE-DONE / PROBLEMS J&P
- Recognize the joined-up numbers
- Detect the unrecognizable handwritten digits
- Differentiating similar looking digits
- Poor quality handwritten digits identification

9. PROBLEM ROOT CAUSE RC
- Scribbled digits, joined up digits, widely varying handwriting of people makes the digits unclear.
- reduced eyesight of people makes it difficult for them to understand the digit

7. BEHAVIOUR BE
- Customer uses a handwritten digit recognition system to resolve their problems

Focus on J&P, tap into BE,

3. TRIGGERS TR
- Entering wrong data in forms and applications due to doubts
- Eyesight problems trigger people to seek for a solution.

10. YOUR SOLUTION SL
- Using MNIST dataset, convolutional neural network is trained repeatedly to predict the unrecognizable digits

8. CHANNELS of BEHAVIOUR CH

8.1 ONLINE
- Real time handwritten digit recognition by giving immediate data

8.2 OFFLINE
- Recognizing scanned handwritten digits

4. EMOTIONS: BEFORE / AFTER EM
- Before: feels doubtful about handwritten digits;
- After: undoubtful digits

4. REQUIRMENT ANALYSIS

4.1 Functional requirement

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Website	❖ A Website is created where the user can login into their workspace with their username and password.
FR-2	Input Data	❖ The user should upload the handwritten digits in the form of image (JPG or PNG).
FR-3	Pre-processing of the data	❖ An input image is subjected to various operations like noise reduction, document skew correction, slant correction, normalization, smoothing.
FR-4	Feature extraction	❖ Feature extraction is analysing the images and deriving some characteristics from these images that identify each specific element.
FR-5	Classification	❖ To predict the handwritten digits, the MNIST dataset is used for training and testing the input data performed by Convolution neural networks.
FR-6	Output data	❖ System will produce the output data with better accuracy.

4.2 Non-functional requirements

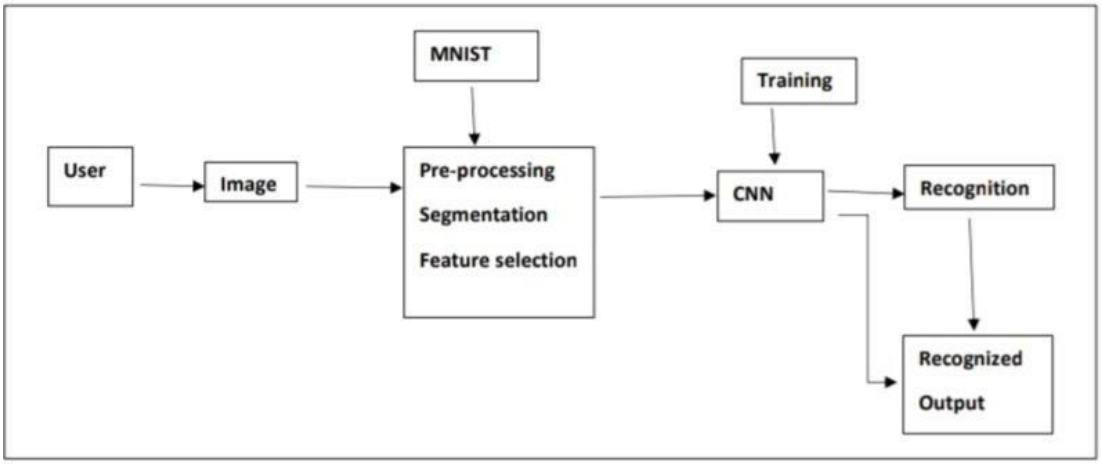
Non-functional Requirements:

Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	❖ Handwritten digit recognition be used are banking sector where it can be used to maintain the security pin numbers.
NFR-2	**Security**	❖ Individual user provided with username and password to access the system. The data those entered in the system are secured and protected from intruder.
NFR-3	**Reliability**	❖ Handwritten digit recognition with 98% accuracy.
		❖ Reliability is the extent to which the software system consistently performs the specified functions without failure.
NFR-4	**Performance**	❖ Quick response ❖ It essentially specifies how the system should behave and that it is a constraint upon the systems behaviour.
NFR-5	**Availability**	❖ Since we use cloud, the availability of this software is all over the world only internet facility is needed.
NFR-6	**Scalability**	❖ Handles more amount of data without compromising on performance and accuracy of result.

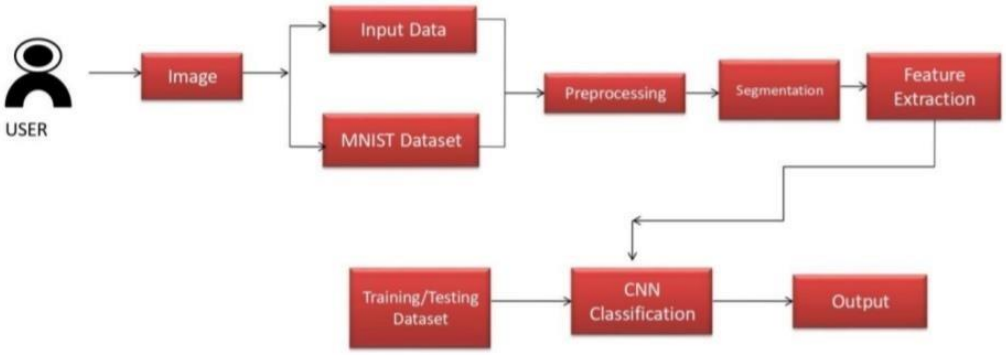
5. Project design

5.1 Data Flow Diagrams

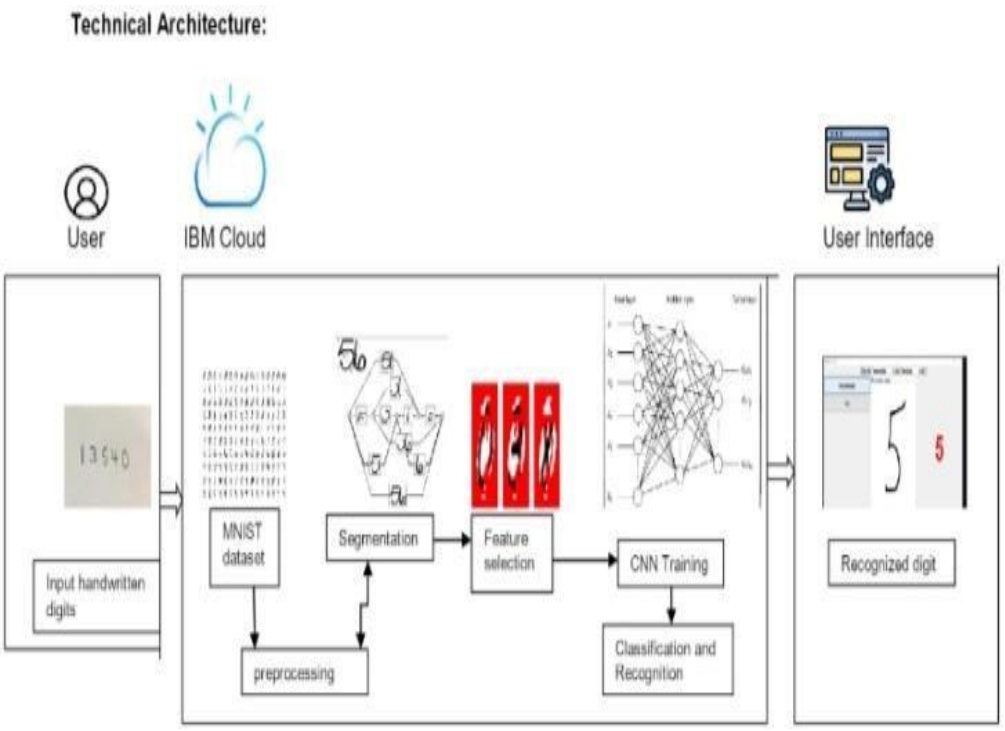


5.2Solution & Technical architecture

Solution architecture



Technical architecture



5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail	I can register and access the dashboard with Google Mail	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	I can login to the application to give my input	High	Sprint-1
	Dashboard	USN-6	As a user. I can move to dashboard section, involves to view various types of visual data	I can see many visual data in dashboard	Low	Sprint-2
Customer (Web user)	Functionality	USN-7	As a web user, I want to give input data to get recognition	I can get the probable match for my input data	High	Sprint-1
Customer Care Executive	Communication	USN-8	As a customer care executive, I can assist the customer to fulfil their expectations	By their suggestions, I can fulfil their needs	High	Sprint-1
Administrator	Classification	USN-9	As an administrator, I can preprocess the data for further process	I can used the pre-processed data	Medium	Sprint-2

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
	Formatting	USN-10	As an administrator, I can check for the suitable format	I will get errors if it is not in proper format	Low	Sprint-3
	Testing and Training	USN-11	As an administrator, I can feed the data to the CNN	I can train and test the data	High	Sprint-1
	Accuracy	USN-12	As an administrator, I can compare the text and give better accuracy	I can get better accurate output	Medium	Sprint-2
	User Experience	USN-13	As an administrator, I can get the most probable	I can get the most probable digit/character	High	Sprint 1

6.PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

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

Velocity:
we have a 6-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

Average Velocity = (20/6) =3.33

6.2 Sprint Delivery Schedule

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration & Login	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	Abimanyu . B
Sprint-2	Image data	USN-2	User upload handwritten image to be recognized	1	High	Mahesh Kumar . B
Sprint-3	Classifier model	USN-3	CNN model for image classification	2	High	Dhinesh . M Muhil. K
Sprint-4	Cloud storage	USN-4	Train the model with help of IBM cloud	2	High	Udhaya . S

6.3Reports from JIRA

Velocity Report

7.CODING & SOLUTIONING

7.1 Feature 1

- Importing the required libraries such as MNIST, Keras, Tensor board, Tensor flow, Tensor flowbase, Tensor flow estimator which are required for the model to run.
- The dataset for this model is imported from the Keras module.
- The dataset contains ten classes: Digits from 0-9.
- The MNIST dataset has 60000 training image and 10000 testing image with 28*28 pixel.
- Sequential class is introduced to this model for creating the CNN model.

- The CNN model layers such as Dense, Dropout, Flatten, Conv2D, Maxpooling 2D are imported from the keras layer.
- To load the data, we split the data into train and test. Using the training dataset we train the model and the testing dataset is used to predict the results.
- The X train & test represents the independent data set and the Y train &test represents the dependent data set.
- We are finding out the shape of X_train and X_test for better understanding. It lists out the dimensions of the data present in it.
- The information of an image lying inside the x_train variable.
- Basically, the pixel values range from 0-255.
- Here we are analyzing the data by printing the first image pixel value which is index[0] of the training data. As you see it is displayed in the output.
- The label of this image will be stored in y_train.
- Matplotlib is a comprehensive library for creating static, animated, and interactive in visualizations Python. By using the Matplotlib library we are displaying the number ‘5’ in the form of an image for proper understanding.

IMPORTING THE LIBRARIES

In [1]:
```
import numpy
import tensorflow
from tensorflow.keras.datasets import mnist
from tensorflow.keras.models import Sequential
from tensorflow.keras import layers
from tensorflow.keras.layers import Dense, Flatten
from tensorflow.keras.layers import Conv2D
from keras.optimizers import Adam
from keras.utils import np_utils
```

LOADING OF THE DATA

In [3]:
```
(X_train, y_train), (X_test, y_test) = mnist.load_data()
```

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

In [4]:
```
print(X_train.shape)
print(X_test.shape)
```

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

ANALYSING THE DATA

In [10]:
```
X_train[4]
```

```
       [  0,   0,   0,   0,   0,   0,   0,  55, 235, 253, 217, 138,  42,
         24, 192, 252, 143,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
         62, 255, 253, 109,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
         71, 253, 252,  21,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0, 253, 252,  21,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
         71, 253, 252,  21,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
        106, 253, 252,  21,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
         45, 255, 253,  21,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0, 218, 252,  56,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,  96, 252, 189,  42,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,  14, 184, 252, 170,  11,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,  14, 147, 252,  42,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
```

In [10]:

```
X_train[4]
```

Out[10]:

```
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,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,  55,
        148, 210, 253, 253, 113,  87, 148,  55,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,  87, 232,
        252, 253, 189, 210, 252, 252, 253, 168,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   4,  57, 242, 252,
        190,  65,   5,  12, 182, 252, 253, 116,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,  96, 252, 252, 183,
```

```
          0,  14, 184, 252, 170,  11,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,  14, 147, 252,  42,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   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)
```

In [11]:

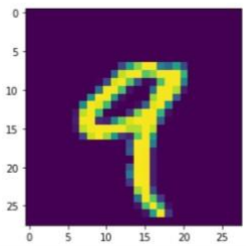
```
y_train[4]
```

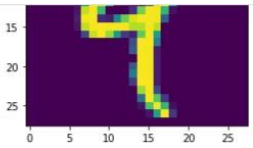
Out[11]: 9

In [12]:

```
import matplotlib.pyplot as plt
plt.imshow(X_train[4])
```

Out[12]:





RESHAPING THE DATA

In [14]:
```
X_train = X_train.reshape(60000, 28, 28, 1).astype('float32')
X_test = X_test.reshape(10000, 28, 28, 1).astype('float32')
```

APPLYING ONE HOT ENCODING

In [17]:
```
number_of_classes = 10
y_train = np_utils.to_categorical(y_train, number_of_classes)
y_test = np_utils.to_categorical(y_test, number_of_classes)
```

In [18]:
```
y_train[4]
```

Out[18]:
```
array([0., 0., 0., 0., 0., 0., 0., 0., 0., 1.], dtype=float32)
```

- You can see the results by replacing the index number.
- We are reshaping the dataset because we are building the model using CNN. As CNN needs four attributes batch, height, width, and channels we reshape the data.
- We use the y_train variable which contains Labels representing the images containing in x_train. AS these are numbers usually they can be considered as numerical or continuous data, but with respect to this project these Numbers are representing a set of class so these are to be represented as categorical data, and we need to binaries these categorical data that's why we are applying One Hot encoding for y_train set.
- The label 5 is index 0 of y_train is converted to the label in the form of 0’s and 1’s and is of type float.

7.2 Feature 2

- This feature includes initializing the model, adding CNN layers, training and testing the model and saving the model.
- A CNN model generally consists of convolutional and pooling layers.
- It works better for data that are represented as grid structures this is the reason why CNN works well for image classification problems.
- Creating the model and adding the input, hidden, and output layers to it. The Sequential model is a linear stack of layers. So, we create a Sequential model by passing a list of layer instances.
- With both the training data defined and model defined, we 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.
- The model.fit() function of Keras will start the training of the model. It takes the training data, validation data, epochs and batch size.
- It takes some time to train the model. After training, we save the weights and model definition in the mnist.h5' file.
- Observing the model includes the test loss and test accuracy.
- The loss value implies how poorly or well a model behaves after each iteration of optimization.
- An accuracy metric is used to measure the algorithms performance in an interpretable way.

- Testing the model is done by slicing the x_test data until the first four images. Then step we print the predicted output.
- As we predicted the input from the x_test. According to that by using argmax function here we are printing the labels with high prediction values.
- Now we saved the model for future purposes. This saved model can also be integrated with an android application or web application in order to predict something.

In [25]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from keras.utils import np_utils
from tensorflow.keras.datasets import mnist
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, Dense, Flatten
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.models import load_model
from PIL import Image, ImageOps
import numpy
```

In [ ]:
```
```

In [2]:
```
(X_train, y_train), (X_test, y_test) = mnist.load_data()
```

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

In [3]:
```
print(X_train.shape)
print(X_test.shape)
```

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

In [4]:
```
X_train[0]
```

Out[4]:
```
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,   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,   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,   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,   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,   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,   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,   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,
   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,   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,   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,   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,   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,   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,   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,   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,   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,
```

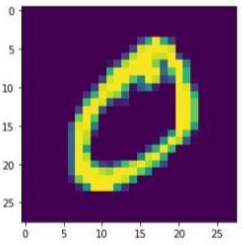
```
        0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
        0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
        0,   0]], dtype=uint8)
```

In [5]:
```
y_train[0]
```

Out[5]: 5

In [11]:
```
plt.imshow(X_train[1])
```

Out[11]:



In [12]:
```
X_train = X_train.reshape(60000, 28, 28, 1).astype('float32')
X_test = X_test.reshape(10000, 28, 28, 1).astype('float32')
```

In [13]:
```
number_of_classes = 10
Y_train = np_utils.to_categorical(y_train, number_of_classes)
Y_test = np_utils.to_categorical(y_test, number_of_classes)
```

In [14]:
```
Y_train[0]
```

Out[14]: array([0., 0., 0., 0., 0., 1., 0., 0., 0., 0.], dtype=float32)

Creating the model

In [15]:
```
model = Sequential()
model.add(Conv2D(64, (3, 3), input_shape=(28, 28, 1), activation="relu"))
model.add(Conv2D(32, (3, 3), activation="relu"))
model.add(Flatten())
model.add(Dense(number_of_classes, activation="softmax"))
```

Compiling the model

In [16]:
```
model.compile(loss='categorical_crossentropy', optimizer="Adam", metrics=["accuracy"])
```

Fitting the model

In [17]:
```
model.fit(X_train, Y_train, batch_size=32, epochs=5, validation_data=(X_test,Y_test))
```

```
Epoch 1/5
1875/1875 [==============================] - 205s 109ms/step - loss: 0.2844 - accuracy: 0.9477 - val_loss: 0.0886 - val_accuracy: 0.9729
Epoch 2/5
1875/1875 [==============================] - 206s 110ms/step - loss: 0.0716 - accuracy: 0.9776 - val_loss: 0.0771 - val_accuracy: 0.9769
Epoch 3/5
1875/1875 [==============================] - 204s 109ms/step - loss: 0.0513 - accuracy: 0.9837 - val_loss: 0.1019 - val_accuracy: 0.9710
Epoch 4/5
1875/1875 [==============================] - 222s 119ms/step - loss: 0.0408 - accuracy: 0.9873 - val_loss: 0.0890 - val_accuracy: 0.9767
Epoch 5/5
1875/1875 [==============================] - 209s 112ms/step - loss: 0.0302 - accuracy: 0.9906 - val_loss: 0.0918 - val_accuracy: 0.9772
```

Observing the metrices

In [18]:
```
metrics = model.evaluate(X_test, Y_test, verbose=0)
print("Metrics (Test Loss & Test Accuracy): ")
print(metrics)
```

```
Metrics (Test Loss & Test Accuracy):
[0.09176069498062134, 0.9771999716758728]
```

Predicting the output

In [19]:
```
prediction = model.predict(X_test[:4])
print(prediction)
```

```
1/1 [==============================] - 0s 86ms/step
[[1.30325325e-11 1.95553570e-17 4.99655983e-10 2.01586161e-07
  2.20858217e-14 9.75999270e-14 9.15056906e-17 9.99999523e-01
  2.50245790e-07 5.38577793e-10]
 [1.79366751e-07 3.38282398e-08 9.99999285e-01 1.66379285e-10
  2.10076114e-14 9.13577841e-17 4.58118137e-07 5.16888727e-15
  2.91890595e-10 3.41794947e-16]
 [5.36327924e-08 9.99545157e-01 1.32159499e-07 6.90754225e-12
  2.88232812e-04 1.63993846e-07 1.19756329e-08 9.81503305e-08
  1.66109443e-04 2.50516774e-10]
 [1.00000000e+00 3.44754093e-19 1.12327614e-14 9.01947470e-16
  7.90818081e-15 2.17030373e-13 1.37855594e-09 8.90824635e-15
  7.06134152e-16 1.75300985e-12]]
```

In [20]:
```
print(numpy.argmax(prediction, axis=1))
print(Y_test[:4])
```

```
[7 2 1 0]
[[0. 0. 0. 0. 0. 0. 0. 1. 0. 0.]
 [0. 0. 1. 0. 0. 0. 0. 0. 0. 0.]
```

saving the model

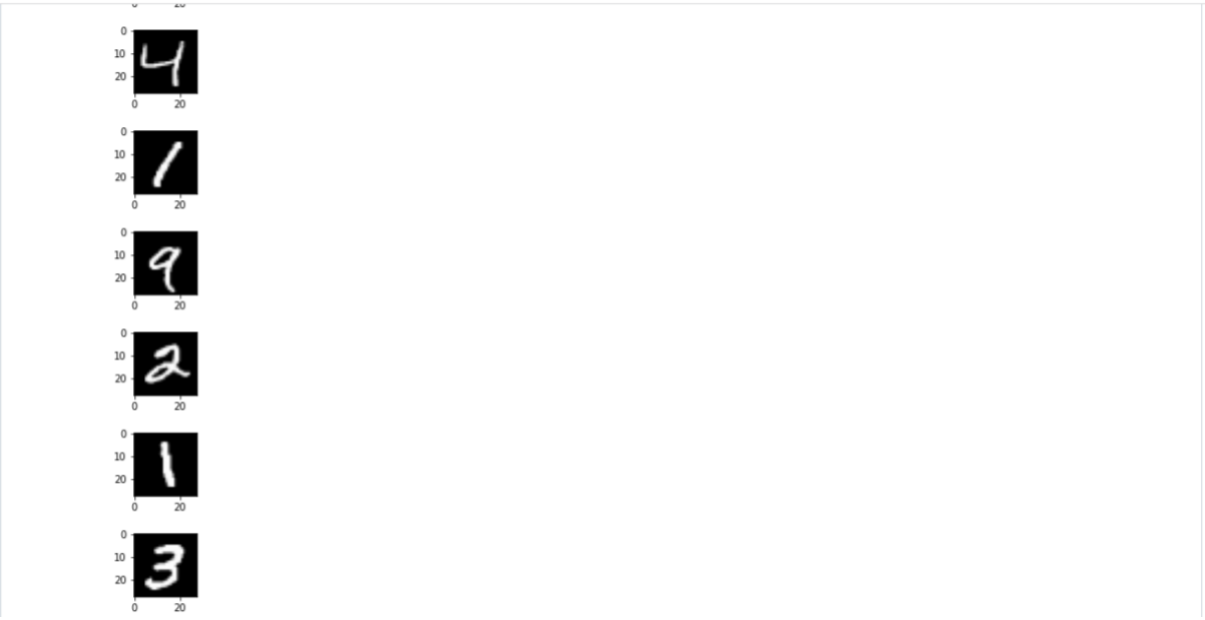
In [21]:
```
model.save("model.h5")
```

Test with saved model

In [26]:
```
model=load_model("model.h5")
```

In [27]:
```
from keras.datasets import mnist
from matplotlib import pyplot
(X_train,y_train),(X_test,y_test)=mnist.load_data()
print('X_train:' +str(X_train.shape))
print('y_train:' +str(y_train.shape))
print('X_test:' +str(X_test.shape))
print('y_test:' +str(y_test.shape))
from matplotlib import pyplot
for i in range(9):
  pyplot.subplot(330+1+i)
  pyplot.imshow(X_train[i],cmap=pyplot.get_cmap('gray'))
  pyplot.show()
```

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



- We saved the model with .h5 extension.
- An H5 file is a data file saved in the Hierarchical Data Format (HDF).
- It contains multidimensional arrays of scientific data.
- Now we are testing the saved model by loading the model which was built. Then we are applying for a loop for the first four images and converting the image to the required format.
- Then we are resizing the input image, converting the image as per the CNN model and we are reshaping it according to the requirement. At last, we are predicting the result.

7.3 Database Schema

- IBM Db2 on Cloud is a fully managed cloud SQL database that offers a dedicated operations team, point-in-time recovery high-availability disaster recovery (HADR) technology with multizone region support and independent scaling to protect your enterprise applications.

8.TESTING

8.1 Test Cases

Test case ID	Feature Type	Component	Test Scenario	Expected Result	Actual Result	Status
Homepage_TC_OO1	Functional	Home Page	Verify user is able to see the Homepage when clicked on the link	Home Page should be displayed.	Working as expected	Pass
Homepage_TC_OO2	UI	Home Page	Verify the UI elements in Homepage	Application should show below UI elements: a.choose file button b.predict button c.clear button	Working as expected	Pass
Homepage_TC_OO3	Functional	Home Page	Verify user is able to choose file from the local system and click on predict	Choose file popup screen must be displayed and user should be able to click on predict button	Working as expected	Pass
Homepage_TC_OO4	Functional	Home page	Verify user able to select invalid file format	Application won't allow to attach formats other than ".png, .jiff, .pjp, .jpeg, .jpg, .pjpeg"	Working as expected	Pass
Predict_TC_OO5	Functional	Predict page	Verify user is able to navigate to the predict to and view the predicted result	User must be navigated to the predict page and must view the predicted result	Working as expected	Pass

8.2 User Acceptance Testing

Defect analysis

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	0	0	0	0	0
Duplicate	0	0	0	0	0
External	0	0	0	0	0
Fixed	0	0	0	0	0
Not Reproduced	0	0	0	0	0
Skipped	0	0	0	0	0
Won't Fix	0	0	0	0	0
Totals	0	0	0	0	0

Test Case Analysis

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

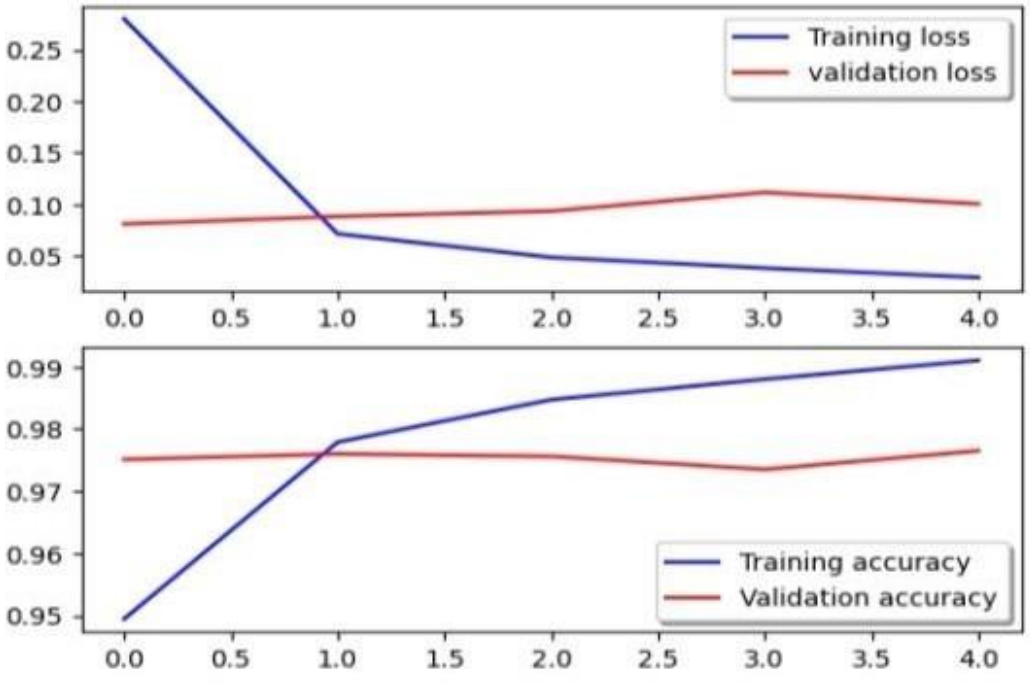
9. RESULTS

9.1Performance 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
_________________________________________________________________
None
```

Accuracy



10.ADVANTAGES & DISADVANTAGES

Advantages

- Detect the digits with accuracy rate.
- Provides offline digit recognition.
- Reliable method of digit recognition.

Disadvantages

- Randomness in detection.
- Joined up handwriting may confuse the system.

11.CONCLUSION

In this project an extensive review of recent advancement in the field of handwritten digit recognition has been presented. The review presented covers all the aspects for handwritten digit recognition like off-line and on-line recognition, different features used, and finally various types of classifies recently used for digit classification. Moreover, all the important and recent works have been discussed with their advantages and limitations. It has been also discussed, most of the available systems were developed for particular database and yet to analyze over real time handwritten digits.

12.FUTURE SCOPE

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.