

# **PROJECT REPORT**

## **A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM**

**Submitted by**

**PNT2022TMID46416**

Sangeetha M – 820319106018

Booshitha M – 820319106004

Harini B – 820319106009

Kaviyarasi K - 820319106010

# **TABLE OF CONTENTS**

## **1. INTRODUCTION**

- 1.1 Project Overview
- 1.2 Purpose

## **2. LITERATURE SURVEY**

- 2.1 Existing problem
- 2.2 References
- 2.3 Problem Statement Definition

## **3. IDEATION & PROPOSED SOLUTION**

- 3.1 Empathy Map Canvas
- 3.2 Ideation & Brainstorming
- 3.3 Proposed Solution
- 3.4 Problem Solution fit

## **4. REQUIREMENT ANALYSIS**

- 4.1 Functional requirement
- 4.2 Non-Functional requirements

## **5. PROJECT DESIGN**

- 5.1 Data Flow Diagrams
- 5.2 Solution & Technical Architecture
- 5.3 User Stories

## **6. PROJECT PLANNING & SCHEDULING**

- 6.1 Sprint Planning & Estimation
- 6.2 Sprint Delivery Schedule
- 6.3 Reports from JIRA

## **7. CODING & SOLUTIONING**

7.1 Feature 1

7.2 Feature 2

7.3 Database Schema

## **8. TESTING**

8.1 Test Cases

8.2 User Acceptance Testing

## **9. RESULTS**

9.1 Performance Metrics

## **10. ADVANTAGES & DISADVANTAGES**

## **11. CONCLUSION**

## **12. FUTURE SCOPE**

## **13. APPENDIX**

Source code

GitHub & Project Demo Link

# **1.INTRODUCTION**

## **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 flavours.
- 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 mail sorting, processing bank

check amounts, numeric entries in forms filled up by hand (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 post but he is unable to recognize the numbers because the numbers are joined up together which makes him feel confused.
- **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 in handwritten 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 the numbers but he is unable to recognize the numbers because of aging and poor recognizing ability which makes him feel confused.

### 2.2 References

1. Roini.M, Dr.D.Surendran. -Technical Research Organization, India.2019 – A Novel Method of handwritten digit recognition using deep learning.

2. Fathima Siddique, shadman Sakib, Md. Abu Bakr Siddique. -5<sup>th</sup> International conference on advances in electrical engineering (ICAEE).26-28 Sept.2019. – Recogniton of Handwritten Digit using Convolutional Neural Network in Python with Tensorflow and comparison of performance for various Hidden layer.
3. Ali Abdullah Yahya, Jiequing Tan, Min Hu -Sensors 2021. 18<sup>th</sup> Sept 2021. – A novel handwritten digit classification system based on convolutional neural network.
4. Mojitaba Mohammadpoor, Abbas Mehdizadeh, Hava Alizadeh Noghabi. – Majlesi Journal of Electrical Engineering. Ssept 2017. – A Novel Method for Persian Handwritten Digit Recognition using Support vector machines.
5. Ssavita Ahlawat, Amit Choudhary, Anand Nayyar, Saurabh singh, Byungun Yoon. – Sensors 2020 12June 2020. – Improve Handwritten digit Recognition Using Convolutional Neural Networks.

● <https://www.researchgate.net/publication/354755659a>  
<http://troindia.in/journal/ijcesr/vol6iss6part2/32-36.pdf>

## 2.3 Problem statement definition



## 3. IDEATION & PROPOSED SOLUTION

### 3.1 Empathy Map Canvas



## 3.2 Ideation & Brainstorming

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

#### Brainstorm & idea prioritization

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.

10 minutes to prepare  
 1 hour to collaborate  
 2-8 people recommended

[Share template feedback](#)

**BEFORE GOING TO COLLABORATE**

Collect the information regarding the problem statement with the help of reference papers, existing solution and research papers.

Taking 2 hours to complete

**TEAM GATHERING**

All the team members in our team togethered in a GoogleMeet to colobrate our ideas.

**SETTING THE GOAL**

Our goal is to fund a better system to recognize the undefined handwritten digits using AI

Learning about various tools and prerequisite for further preceeding the project.

**DEFINE THE PROBLEM STATEMENT**

Handwritten Digit Recognition system is a process to provide the ability to machines to recognize human handwritten digits. Handwritten digits are not perfect, vary from person to person, can be made with different flavors. Sometime characters (digits 0-9) looks similar makes it hard for computer to recognize accurately.

**Key rules of brainstorming**  
Be sure to smooth and productive session.

- Stay on topic
- Encourage wild ideas
- Check judgement
- Listen to others
- No free wheeling
- If possible, be visual

**PROBLEM**

How to recognize the undefined handwritten digits with the use of artificial intelligence?

### Step-2: Brainstorm, Idea Listing and Grouping

**2**

**Brainstorm**

Write down any ideas that come to mind that address your problem statement.

10 minutes

**3**

**Group ideas**

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you can break it up into smaller sub-groups.

20 minutes

Sangeetha.M	Booshitha.M	Harini.B	Kaviyarasi.K
Use MNIST dataset for data recognition	artificial neural network and deep learning	connection temporal classification (cte) algorithm	decision tree for repeated training
Using Touch panel for entering the data.	shape, slant and slope analysis of digit's image	neural network uses examples to automatically infer rules for digit recognition	OCR technology
Import more training and testing dataset.	handwritten digits digitalized through scanners and cameras	increased number of training improves accuracy	image classification, object detection, instance segmentation methods
Performing Convolution layer and ANN for data processing	OCR techniques to recognize each digit	offline recognition by optical scanning & intelligent word recognition	Using MNIST dataset
Usage Open CV	import and use vast libraries present in python	object identification, edge detection techniques	using convolutional neural network in deep neural network

use MNIST dataset which classifies given digit into one of ten classes representing integer values from 0 to 9

use CNN because CNN is SUPERVISED type of DEEP LEARNING which is most prefferable in image recognition & computer vision

MNIST dataset uses CNN to get accuracy

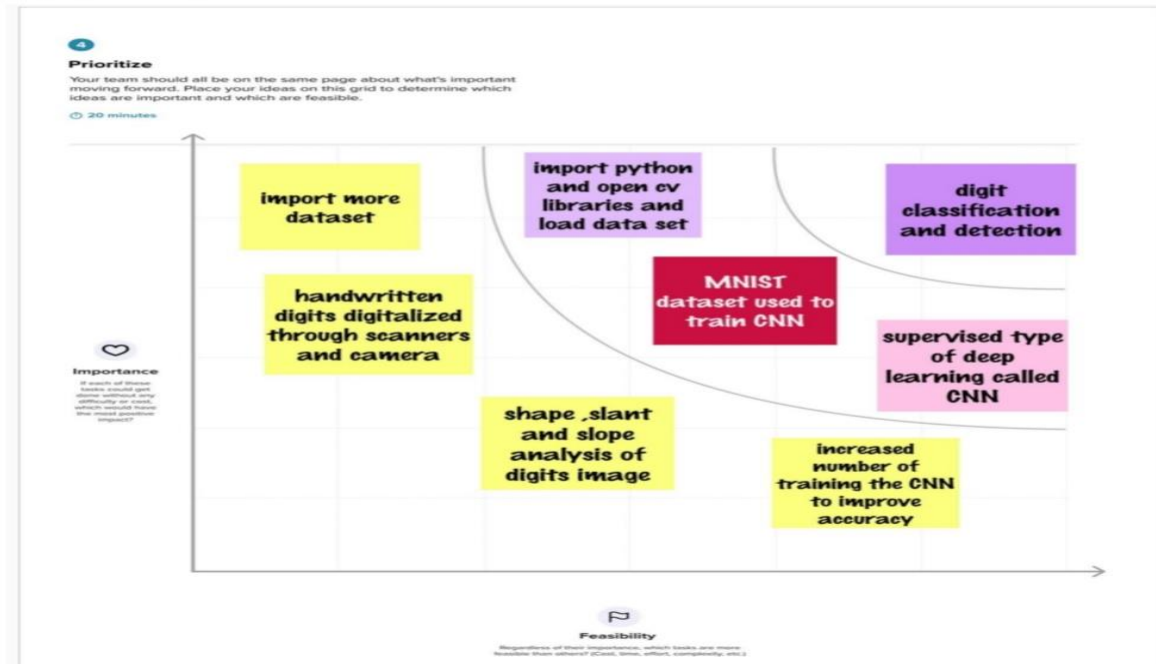
the CNN is applicable for 1D and 2D array of data

import python and open cv libraries and load the data set

the MNIST database can be used to train a CNN to predict the given digit image



### 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: PNT2022TMD46416

Project Design Phase-I - Solution Fit Template

Define CS, fit into CC	<b>1. CUSTOMER SEGMENT(S)</b> <span>CS</span> <ul style="list-style-type: none"> <li>➤ Banking sector,</li> <li>➤ postal sector,</li> <li>➤ form data entry</li> </ul>	<b>6. CUSTOMER CONSTRAINTS</b> <span>CC</span> <ul style="list-style-type: none"> <li>➤ Security threats</li> <li>➤ Doubt of accuracy of interpreted digits</li> <li>➤ procedure of scanning makes them feel hard</li> </ul>	<b>5. AVAILABLE SOLUTIONS</b> <span>AS</span> <ul style="list-style-type: none"> <li>➤ Many algorithms like CTC, SVM classifier etc. Are available for digit recognition</li> <li>➤ AI and Machine learning based recognition systems</li> <li>➤ CNN and ANN based digit recognitions</li> </ul>	Explore AS, differentiate
	<b>2. JOBS-TO-BE-DONE / PROBLEMS</b> <span>JBP</span> <ul style="list-style-type: none"> <li>➤ Recognize the joined-up numbers</li> <li>➤ Detect the unrecognizable handwritten digits</li> <li>➤ Differentiating similar looking digits</li> <li>➤ Poor quality handwritten digits identification</li> </ul>	<b>3. PROBLEM ROOT CAUSE</b> <span>RPC</span> <ul style="list-style-type: none"> <li>➤ Scribbled digits, joined up digits, widely varying handwriting of people makes the digits unclear.</li> <li>➤ reduced eyesight of people makes it difficult for them to understand the digit</li> </ul>	<b>7. BEHAVIOUR</b> <span>BE</span> <ul style="list-style-type: none"> <li>➤ Customer uses a handwritten digit recognition system to resolve their problems</li> </ul>	
	<b>3. TRIGGERS</b> <span>TR</span> <ul style="list-style-type: none"> <li>➤ Entering wrong data in forms and applications due to doubts</li> <li>➤ Eyesight problems trigger people to seek for a solution.</li> </ul>	<b>10. YOUR SOLUTION</b> <span>SL</span> <ul style="list-style-type: none"> <li>➤ Using MNIST dataset, convolutional neural network is trained repeatedly to predict the unrecognizable digits</li> </ul>	<b>8. CHANNELS of BEHAVIOUR</b> <span>CH</span> <ul style="list-style-type: none"> <li>8.1 ONLINE <ul style="list-style-type: none"> <li>➤ Real time handwritten digit recognition by giving immediate data</li> </ul> </li> <li>8.2 OFFLINE <ul style="list-style-type: none"> <li>➤ Recognizing scanned handwritten digits</li> </ul> </li> </ul>	
	<b>4. EMOTIONS: BEFORE / AFTER</b> <span>EM</span> <ul style="list-style-type: none"> <li>➤ Before: feels doubtful about handwritten digits;</li> <li>➤ After: undoubtful digits</li> </ul>			

## 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:

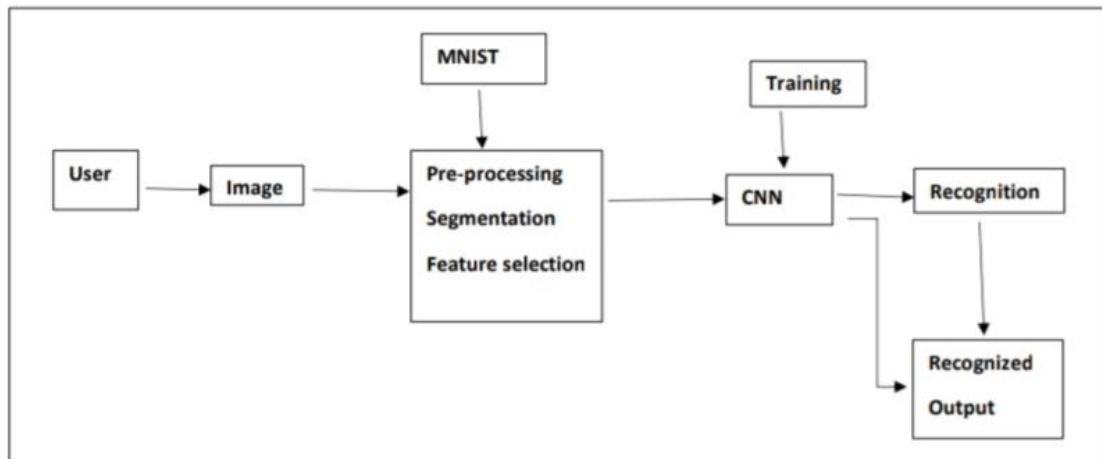
Following are the non-functional requirements of the proposed solution.

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.

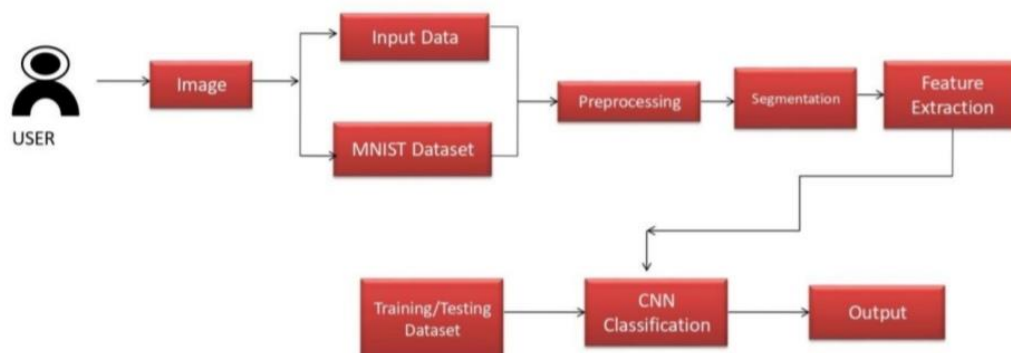
## 5. Project design

### 5.1 Data Flow Diagrams

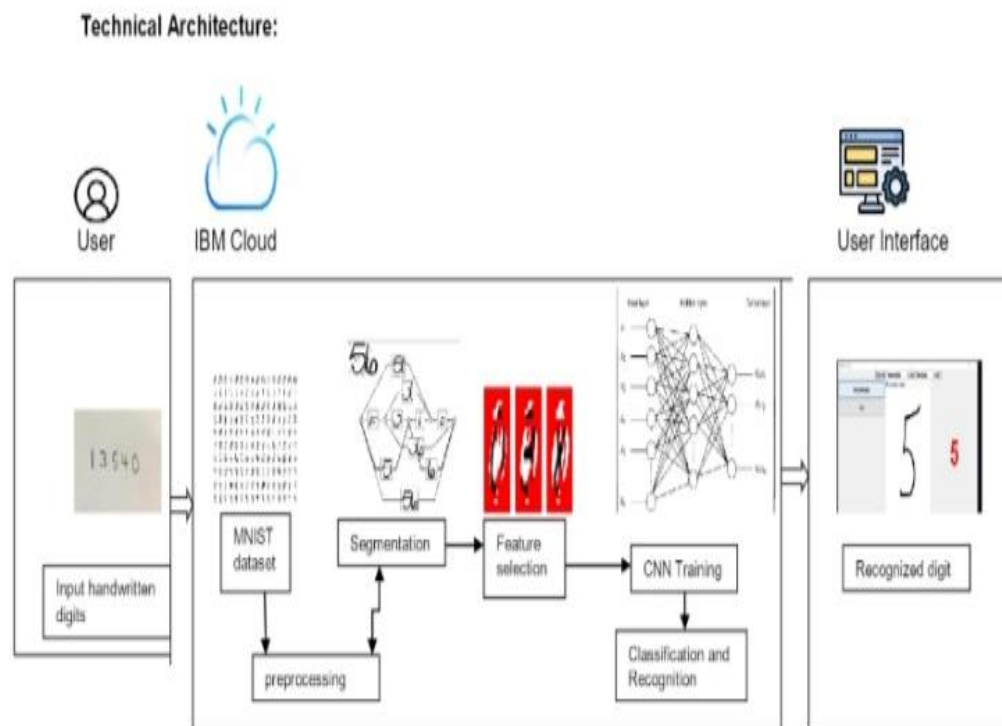


### 5.2 Solution & Technical architecture

#### 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	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)

$$\text{Average Velocity} = (20/6) = 3.33$$

### 6.2 Sprint Delivery 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	Sangeetha. M
Sprint-2	Image data	USN-2	User upload handwritten image to be recognized	1	High	Booshitha. M
Sprint-3	Classifier model	USN-3	CNN model for image classification	2	High	Kaviyarasi.K
Sprint-4	Cloud storage	USN-4	Tain the model with help of IBM cloud	2	High	Harini.B

## 6.3 Reports 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]
```

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

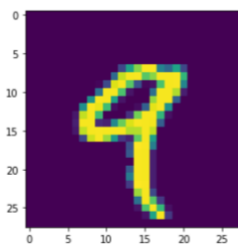
```
0, 14, 184, 252, 170, 11, 0, 0, 0, 0, 0, 0, 0,
0, 0],
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 14, 147, 252, 42, 0, 0, 0, 0, 0, 0, 0,
0, 0],
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0]], dtype=uint8)
```

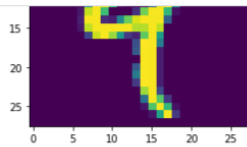
```
In [11]: y_train[4]
```

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.



[illegible]

doi:10.1017/S0022292400001710

Observing the metrics

```
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 9.9999285e-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
  7.90818061e-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.]
 [0. 1. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
```

saving the model

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

Test with saved model

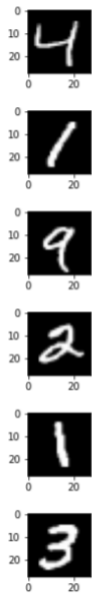
```
In [26]: model=load_model("model.h5")
```

```
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, .jpeg"	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

### 9.1 Performance Metrics

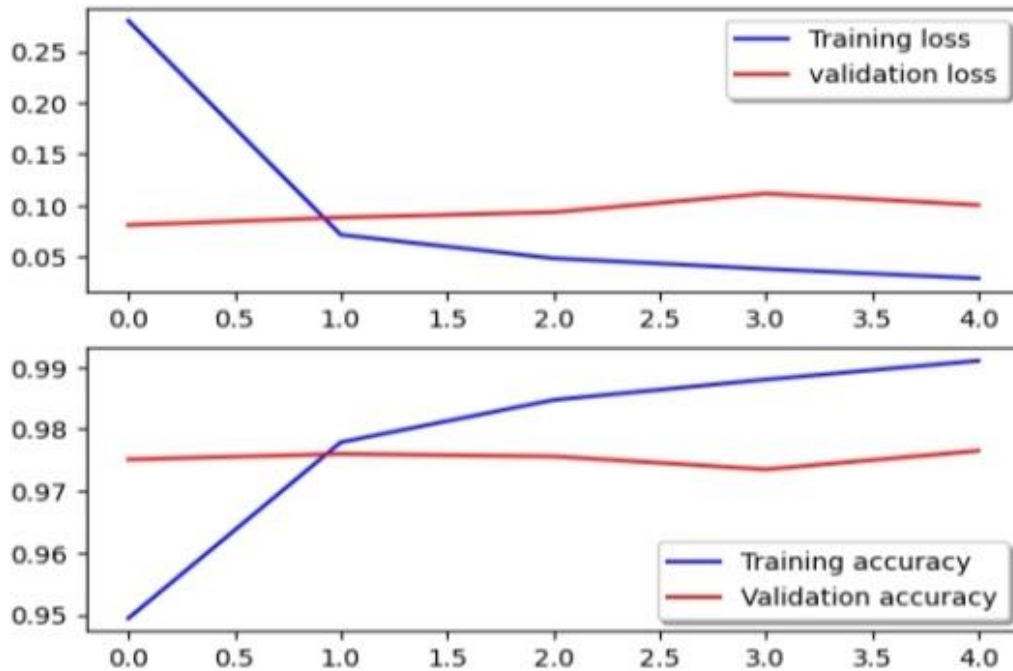
#### Model Summary

```
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  
=====  
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 classifiers 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.

