PROJECT REPORT

A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

Submitted by

PNT2022TMID35466

Mageswari S J	2019105548
Gayathiri N	2019105011
Prathisha A	2019105556
Divya R	2019105528

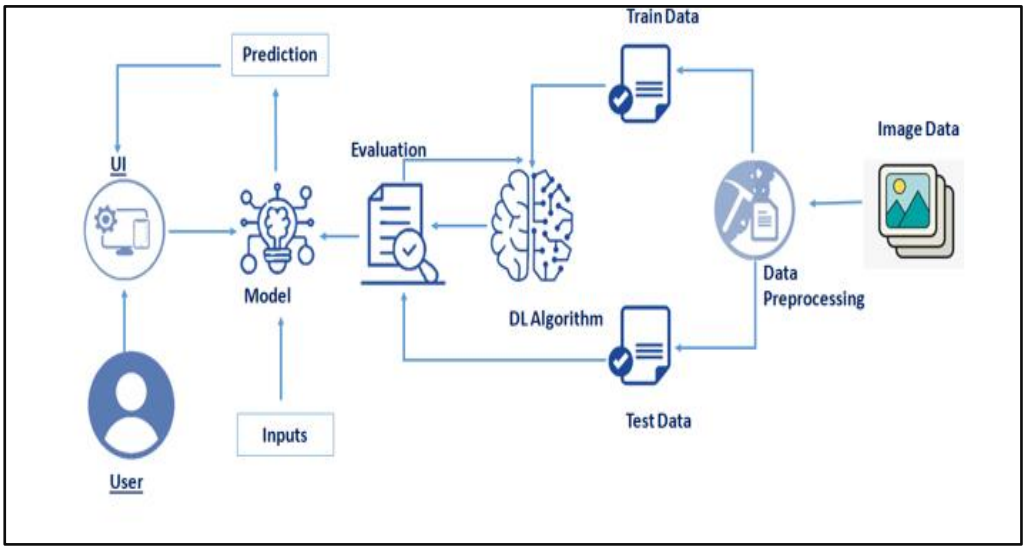
TABLE OF CONTENTS

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 character 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 analysed by the model and the detected result is returned on to UI.

TECHNICAL ARCHITECTURE:



1.2 PURPOSE

Handwritten digits are not perfect and can be made with many different flavours. 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. Digit recognition systems are capable of recognizing the digits from different sources like emails, bank cheque, papers, images, etc. and in different real-world scenarios for online handwriting recognition on computer tablets or system, recognize number plates of vehicles, processing bank cheque amounts, numeric entries in forms filled up by hand (tax forms) and so on.

LITERATURE SURVEY

2.1 EXISTING PROBLEM:

The issue is that there's a wide range of handwriting – good and bad. This makes it tricky for programmers to provide enough examples of how every character might look. Sometimes, characters look very similar, making it hard for a computer to recognise accurately. The fundamental problem with handwritten digit recognition is that handwritten digits do not always have the same size, width, orientation, and margins since they vary from person to person. Additionally, there would be issues with identifying the numbers because of similarities between numerals like 1 and 7, 5 and 6, 3 and 8, 2 and 5, 2 and 7, etc. Finally, the individuality and variation of each individual's handwriting influence the structure and appearance of the digits.

2.2 REFERENCES:

- This paper attempts to use deep learning tools to train a classifier to recognize handwritten digits. Also, the use of techniques in Computer Vision was explored to investigate the effect of selection image preprocessing, feature extraction and classifiers on the overall accuracy. The dataset used for the experiment is MNIST dataset originally constituted of 60,000 training, and 10,000 testing images which are 28 x 28 grayscale (0-255) labeled and bitmap format. Trained the model in 12 epochs and got accuracy of 99.87% and test loss of 0.043464561576.[1]

- Handwritten digit recognition has recently been of very interest among the researchers because of the evolution of various Machine Learning, Deep Learning and Computer Vision algorithms. In this paper the results of some of the most widely used Machine Learning Algorithms like SVM, KNN & RFC and with Deep Learning algorithm like multilayer CNN using Keras with Theano and Tensor flow are compared. [2]

- CNN is able to the learning full architecture CNN (for example, LeNet5) to provide remarkable recognition accuracy in the MNIST handwritten digit recognition. LeNet5 - simple convolutional neural network its recognizing simple digit images. n. For all experimental images, the pixel's values also have been normalized to a range. Where the black pixels are pixels with negative value, the white pixels are pixel with positive value, and the grey pixels are pixels with zero.[3]

- Mahmoud M. Abu Ghosh and Ashraf Y. Maghari compared DNN, CNN and DBN to determine the best algorithm for recognition by considering accuracy, performance and execution time. After experimentations the results were summarized as DNN to be the best algorithm in terms of accuracy and performance, CNN and DNN algorithms were almost equal in terms of accuracy and DNN algorithm was better than CNN and DBN in terms of execution time.[4]

- 'Unconstrained Handwritten Numeral Recognition Using Majority Voting Classifier', the authors, Rajiv Kumar, Pervez Ahmed, Mayank Kumar Goyal, Amresh Kumar presented a simple profile, combined local & global features and majority voting scheme classifier for unconstrained handwritten numeral recognition. Linear discriminant analysis and KNN classifiers are used for classifying these features. A majority voting scheme has been performed with three neural network classifiers and KNN classifiers. The performance is tested on MNIST dataset. The network was trained on 60,000 and tested on 10,000 numeral samples of which 98.05 % test samples are correctly recognized.[5]

2.3 PROBLEM STATEMENT DEFINATION:

- 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.

- 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.

- In different real-world scenarios like recognizing number plates of vehicles, processing bank cheque amounts, identifying numeric entries in forms filled up by hand (say tax forms) and so on , handwritten digit recognition by machine is used.

- The handwritten digits are not always of the same size, width, orientation as they differ from writing of person to person.

- The problem would be while classifying the digits due to the similarity between digits such as 1 and 7, 5 and 6, 3 and 8, 2 and 5, 2 and 7.

- This problem is faced more when many people write a single digit with a variety of different handwritings. The uniqueness and variety in the handwriting of different individuals also influence the formation and appearance of the digits.

- Nowadays the whole world is a shift in the digital world. They want everything in digital form, they not ready for manual work or any manual handwritten transaction.

- Also, they want to avoid the handwritten data.

- Depositing cash requires the physical presence of the depositor at the bank, and cashier needs to enroll the transaction into the system, which slows down the rate of money deposit and tellers activity.

- To overcome such issue, we are proposing to develop this system.

- A lot of information is available on paper, and processing of digital files is cheaper than processing traditional paper files.

- We will be reading images containing handwritten digits extracted from the database and try to recognize which digit is represented by that image.

- Handwriting recognition is the computer recognition of handwritten letters, numbers and characters.

- This process, which is very simple for a human, is difficult for computers.

- In other words, making sense of lines, symbols, and their combined shapes at the word level is difficult for computers.

- Handwriting features such as the presence of characters that are different in many languages, the fact that each person has different handwriting, and the presence of combined handwriting make it difficult for computer systems to recognize handwriting.

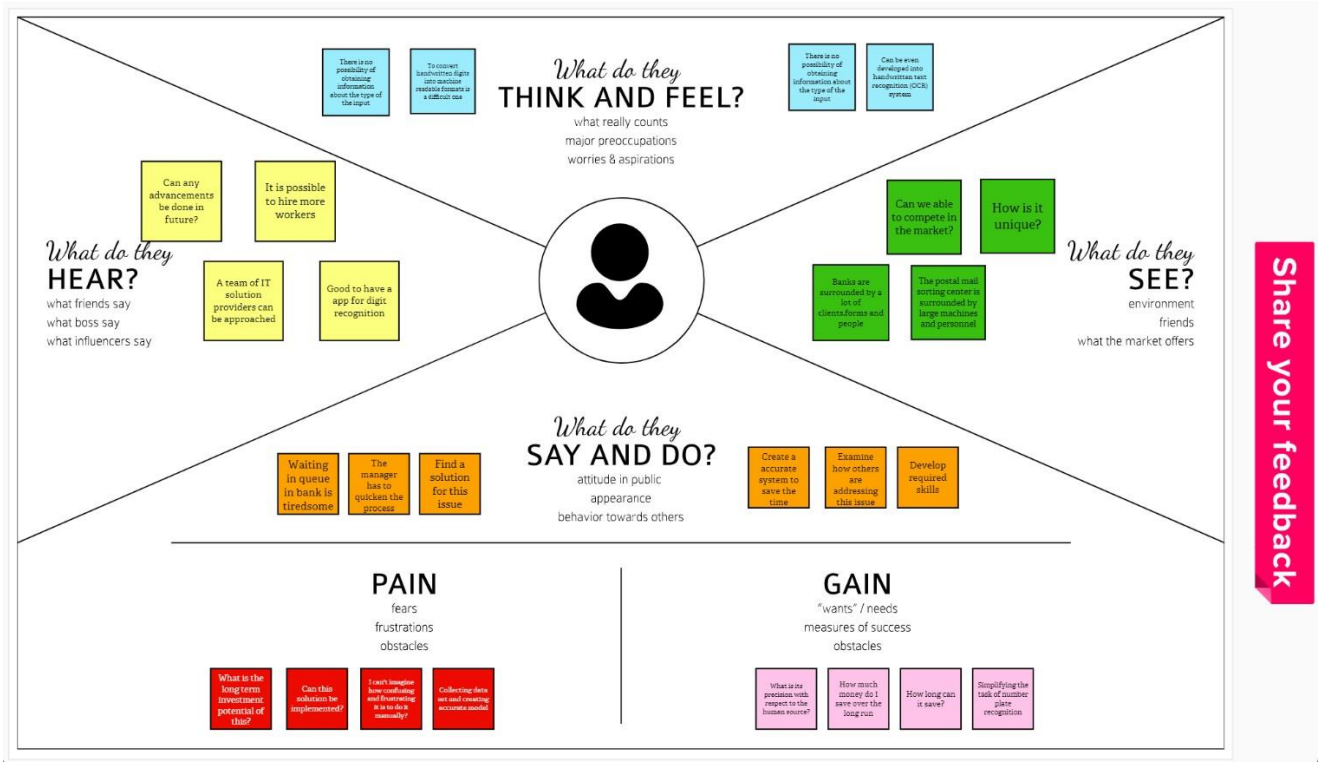
- This topic is not yet fully developed and it is an area of limited efficiency

- When this technology is developed, which is mainly used in tablet computers and for which there are already examples, it will be possible to store and organize any handwritten information in a digital environment without using a keyboard.

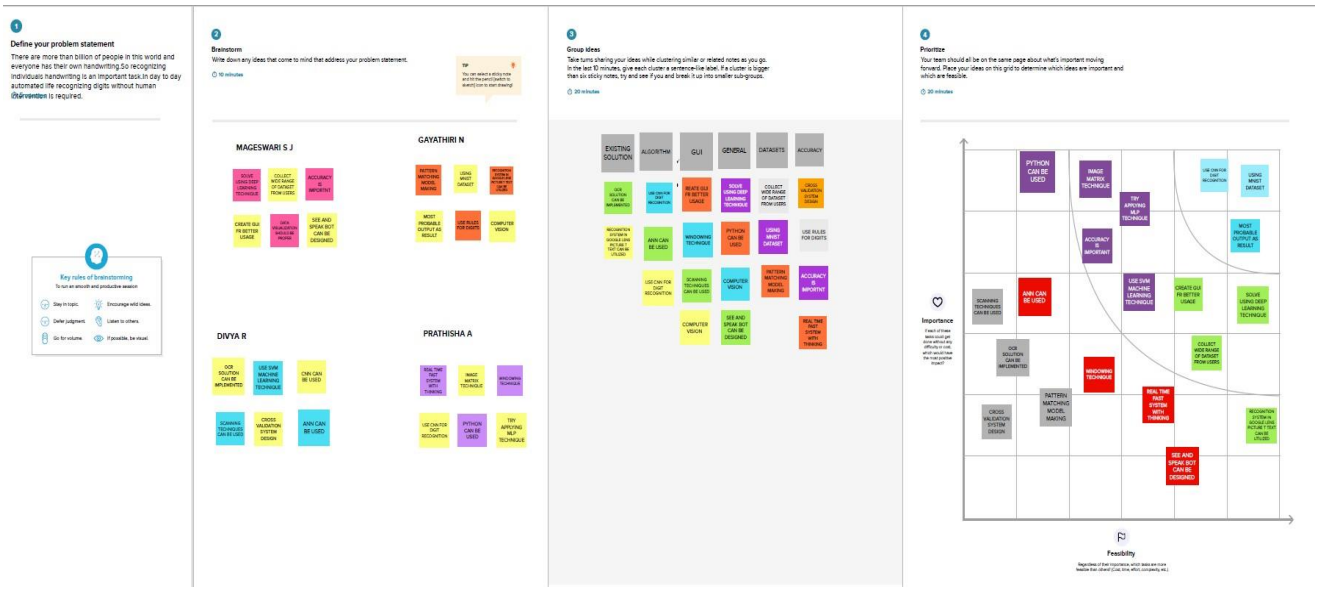
Who does the problem affect?	Postal office workers, courier office employees, bankers and elder people.
What are the boundaries of the problem?	Postal department, courier service and Banking sector
What is the issue?	Sometimes handwritten digits are confusing. so, the important details such as zip code, account number, figure of cash and checks may go wrong. By fixing this problem handwritten digits are recognized correctly. If we didn't solve this problem the mail sorting and transactions may gone wrong.
When does the issue occurs?	• When the banking sector has a failed transaction due to wrong entry of account number, cash. • When postal and courier service are unable to recognize zip code or postal code for mail and courier sorting. • When elder people are unable to understand the handwritten digits.
Where is the issue occurring?	• The issue occurs in the banking sector where more handwritten numbers are involved like figure of cash, account number and checks. • The issue occurs in the postal sector where recognizing zip code or postal code is required for mail sorting.
Why is it important that we fix the problem?	• Postal department and courier services can easily find the digits written. • Old people who will have eye sight issues with handwritten digits can recognize easily.

IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS:



3.2 IDEATION AND BRAINSTORMING:



3.3 PROPOSED SOLUTION:

S.NO	Parameter	Description
1.	Problem Statement (Problem to be solved)	**Statement**-The handwritten digit recognitionis the capability of computer applications to recognize the human handwritten digits. **Description**: It is a hard task for the computer because handwritten digits are not perfect and can be made with many different shapes and sizes and angles.
2.	Idea / Solution description	1. It is the capability of a computer to fetch the handwritten integers from different sources like images and papers. 2. It allows users to translate all those signatures and notes into electronic words ina text document format and this data only requires far less physical space than the storage of the physical copies.
3.	Novelty / Uniqueness	Accurately recognize the digits rather than recognizing all the characters likeOCR.
4.	Social Impact / Customer Satisfaction	1.Develop the application to recognize the digits. 2.It converts the written word into digitalapproximations and utilizes complex algorithms to identify numbers.
5.	Business Model (Revenue Model)	1.This system can be integrated with traffic surveillance cameras to recognize the vehicle's number plates for effective trafficmanagement. 2.Can be integrated with the Postal system to identify and recognize the pin-code details and amount details easily.
6.	Scalability of the Solution	1.Ability to recognize digits in more noisy environments. 2.There is no limit in the number of digits it can be recognized

3.4 PROBLEM SOLUTION FIT:

Define CS, fit into CC

1. CUSTOMER SEGMENT(S)

Customer are the one who are working with reading handwritten digits manually.
Bank employees and postal mail sorters are the major customers.

6. CUSTOMER CONSTRAINTS

Whether digits are recognized Accurately ?

Is this product trustworthy ?

5. AVAILABLE SOLUTIONS

There are no popular softwares to detect the handwriting , they check with the other people to confirm what number it is .

Explore AS, differentiate

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

2. JOBS-TO-BE-DONE / PROBLEMS

Sometimes, the Handwritten digits are very hard to understand and interpret . It may lead to errors while dealing with rugged handwritings

9. PROBLEM ROOT CAUSE

Wide variety of writing styles used by different people.

It is easy for the human to perform a task accurately by practicing it repeatedly and memorizing it for the next time

7. BEHAVIOUR

Find a right product that recognizes the digits written in all kinds of handwriting accurately and fast .

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

3. TRIGGERS

To get number accurately and quickly as possible.

4. EMOTIONS: BEFORE / AFTER

Time consuming ,Manual effort ,Irritated

After: Fast, Less manual effort , Happy customers

10. YOUR SOLUTION

The Handwritten digit recognition system which uses the image of a digit and recognizes the digit present in the image. Convolutional Neural Network model over the MNIST dataset to recognize handwritten digits can be deployed.

8. CHANNELS of BEHAVIOUR

Online: Utilizing software that is offered in the online market.

Offline: Getting help from the persons nearby in order to recognize the digits written by their customers.

REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS:

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Input correlation	Image Correlation is a technique used to recognize characters from images.
FR-2	Data Preparation	Collecting data and prepare it for training
FR-3	Feature extraction	Feature extraction is analysing the images and derive some characteristics from these images that identify each specific element
FR-4	Character classification	During the classification phase, the attributes of the data in the picture are compared to the classes in the database to determine which class the picture belongs to.

4.2 NON-FUNCTIONAL REQUIREMENTS:

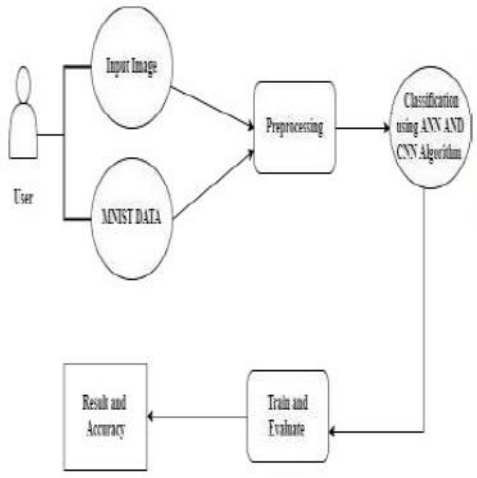
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The software is very easy to use and reduces the learning work.To recognize the digits from bank cheque,papers,numeric entry in forms etc.
NFR-2	**Security**	The handwritten digit recognition can be used by banking sector where it can be used to maintain the security pin numbers, it can be also used for blind peoples by using sound output.
NFR-3	**Reliability**	This software will work reliably for low resolution images and not for graphical images.
NFR-4	**Performance**	Handwritten characters in the input image will be recognized with an accuracy of about 90% and more.
NFR-5	**Availability**	This system will retrieve the handwritten text regions only if the image contains written text in it.
NFR-6	**Scalability**	It contains thousands of handwritten digits that have been used in the development of programs .

PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS:

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

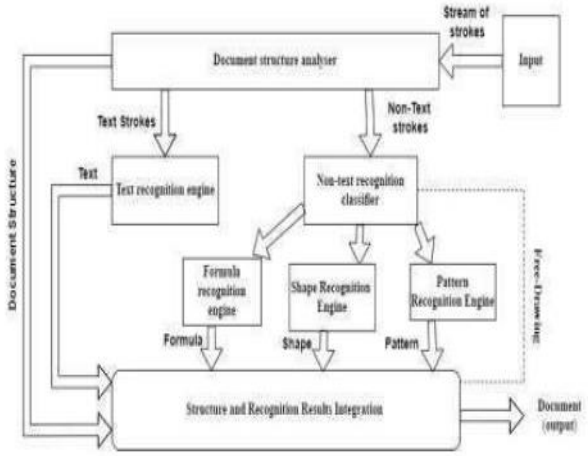
Example:



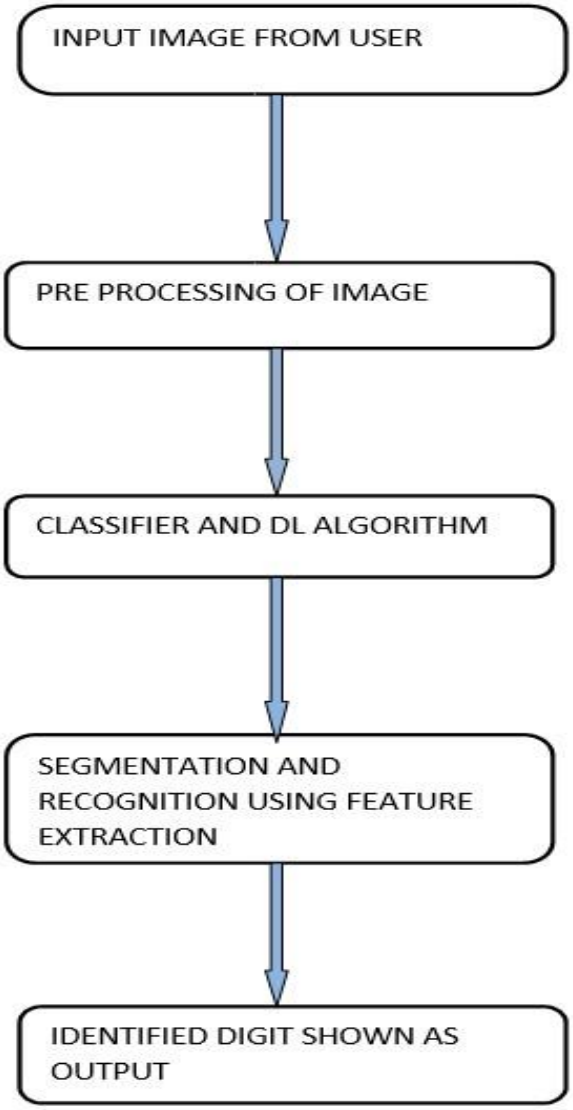
User Stories

Use the below template to list all the user stories for the product.

Example: DFD Level 0 (Industry Standard)

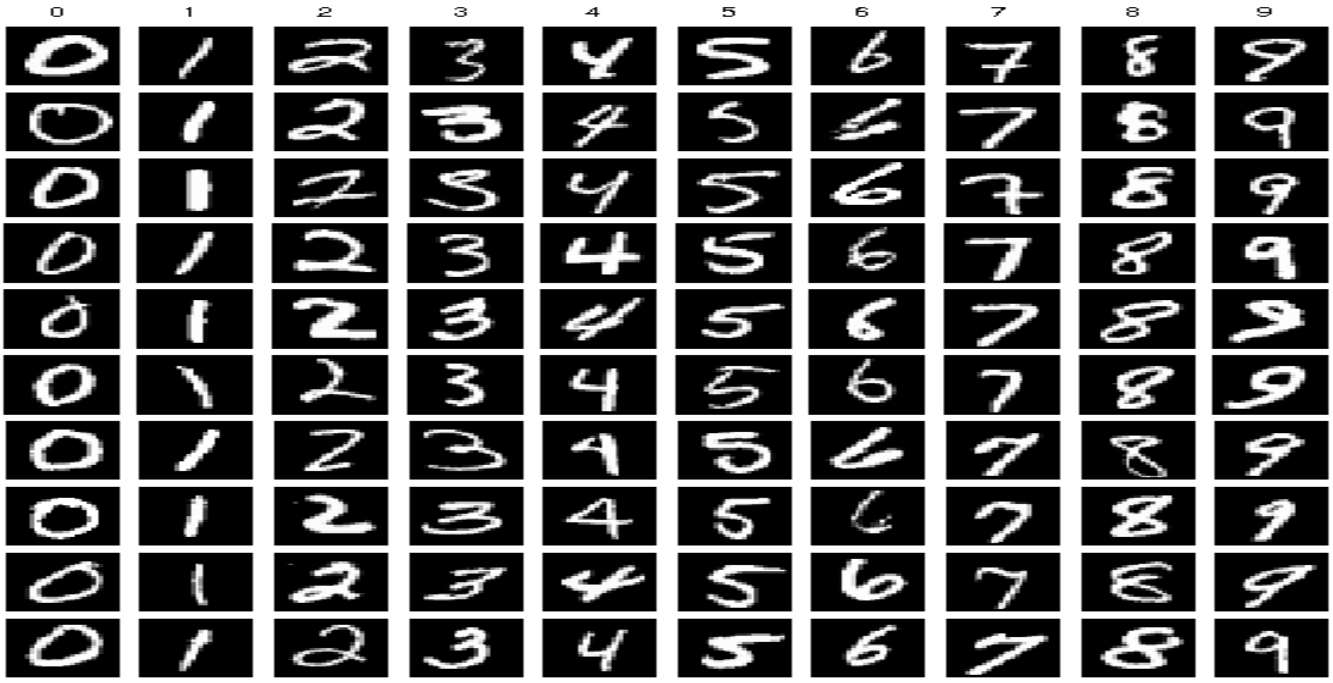


5.2 SOLUTION AND TECHNICAL ARCHITECTURE:



DATASET:

The MNIST data collection, which contains 70000 handwritten digits, is frequently utilized for this recognition method. In order to train these photos and create a deep learning model, we use artificial neural networks. A web application is developed that allows users to upload pictures of handwritten numbers. The model examines this picture. The 60,000 training and 10,000 testing labeled handwritten digit images in the MNIST Handwritten Digit Recognition Dataset.



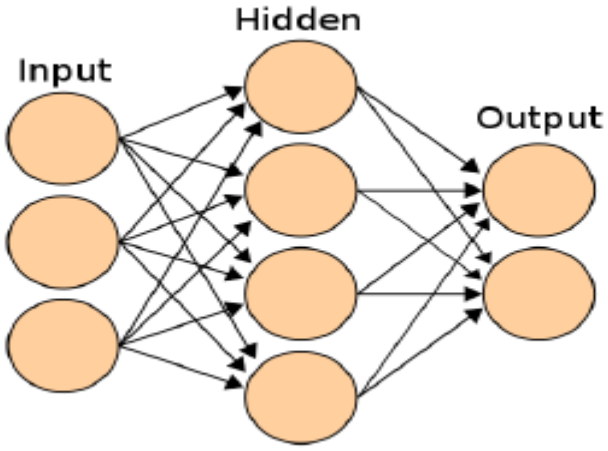
APPROACH:

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

● The input layer: The input layer transfers the information from our example systems to the following layer so that the latter can compute its activations.

● The hidden layer: The network's nonlinear ties are provided by hidden units termed activations that make up the hidden layer. Depending on our needs, there can be a variety of concealed layers.

● The output layer: The nodes in this stratum are referred to as output units. It gives us access to the neural network's final prediction, which may be used to make final predictions.



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

METHODOLOGY:

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

WORKING:

After receiving an input, neural networks change it using a number of hidden layers. Each group of neurons in a hidden layer is completely linked to every other neuron in the layer above it. One layer of neurons have perfect independence from one another. The "output layer" is the final layer to be fully connected.

CONVOLUTION LAYER :

The foundational component of a CNN is the convolutional layer. The parameters of the layer are a set of learnable filters (or kernels) that cover the entire depth of the input volume but have a narrow receptive field. Each filter is convolved across the width and height of the input volume during the forward pass, computing the dot product between each filter entry and the input to create a two-dimensional activation map of the filter. As a result, the network picks up filters that turn on when it detects a certain kind of feature at a particular spatial location in the input.

FEATURE EXTRACTION :

All neurons in a feature share the same weights .In this way all neurons detect the same feature at different positions in the input image. Reduce the number of free parameters.

TENSORFLOW :

An open-source machine learning library for both research and production is called TensorFlow. TensorFlow provides developers of all skill levels with APIs for desktop, mobile, web, and cloud applications. To get started, refer to the sections below. We can achieve text output and sound output by scanning the number digit and converting it to PNG format using the python3 command in the terminal.

5.3 USER STORIES:

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Home	USN-1	As a user, I can view the guide and awareness to use this application.	I can view the awareness to use this application and its limitations.	Low	Sprint-1
		USN-2	As a user, I'm allowed to view the guided video to use the interface of this application.	I can gain knowledge to use this application by a practical method.	Low	Sprint-1
		USN-3	As a user, I can read the instructions to use this application.	I can read instructions also to use it in a user-friendly method.	Low	Sprint-2
	Recognize	USN-10	As a user, I can use the web application virtually anywhere.	I can use the application portably anywhere.	High	Sprint-1
		USN-11	As it is an open source, can use it cost freely.	I can use it without any payment to be paid for it to access.	Medium	Sprint-2
		USN-12	As it is a web application, it is installation free	I can use it without the installation of the application or any software.	Medium	Sprint-4
	Predict	USN-13	As a user, I'm Allowed to upload and choose the image to be uploaded	I can upload and choose the image from the system storage and also in any virtual storage.	Medium	Sprint-3

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Home	USN-1	As a user, I can view the guide and awareness to use this application.	I can view the awareness to use this application and its limitations.	Low	Sprint-1
		USN-2	As a user, I'm allowed to view the guided video to use the interface of this application.	I can gain knowledge to use this application by a practical method.	Low	Sprint-1
		USN-3	As a user, I can read the instructions to use this application.	I can read instructions also to use it in a user-friendly method.	Low	Sprint-2
	Recognize	USN-4	As a user, In this prediction page I get to choose the image.	I can choose the image from our local system and predict the output.	High	Sprint-2
	Predict	USN-6	As a user, I'm Allowed to upload and choose the image to be uploaded	I can upload and choose the image from the system storage and also in any virtual storage.	Medium	Sprint-3
		USN-7	As a user, I will train and test the input to get the maximum accuracy of output.	I can able to train and test the application until it gets maximum accuracy of the result.	High	Sprint-4
		USN-8	As a user, I can access the MNIST data set	I can access the MNIST data set to produce the accurate result.	Medium	Sprint-3
Customer (Web user)	Home	USN-9	As a user, I can view the guide to use the web app.	I can view the awareness of this application and its limitations.	Low	Sprint-1

PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANNING & ESTIMATION:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Point	Priority	Team Members
Sprint-1	Data Collection	USN-1	As a user, I can collect the dataset fromvarious resources with different handwritings.	10	Low	PRATHISHA A MAGESWARI S J
Sprint-1	Data Preprocessing	USN-2	As a user, I can load the dataset, handling the missing data, scaling and split data intotrain and test.	10	Medium	PRATHISHA A MAGESWARI S J
Sprint-2	Model Building	USN-3	As a user, I will get an application with MLmodel which provides high accuracy of recognized handwritten digit.	5	High	GAYATHIRI N DIVYA R
Sprint-2	Add CNN layers	USN-4	Creating the model and adding the input,hidden, and output layers to it.	5	High	GAYATHIRI N DIVYA R

Sprint-2	Compiling the model	USN-5	With both the training data defined and model defined, it's time to configure thelearning process.	2	Medium	GAYATHIRI N DIVYA R
Sprint-2	Train & test the model	USN-6	As a user, let us train our model with ourimage dataset.	6	Medium	GAYATHIRI N DIVYA R
Sprint-2	Save the model	USN-7	As a user, the model is saved & integratedwith an android application or web application in order to predict something.	2	Low	GAYATHIRI N DIVYA R
Sprint-3	Building UI application	USN-8	As a user, I will upload the handwritten digitimage to the application by clicking a uploadbutton.	5	High	DIVYA R PRATHISHA A GAYATHIRI N
Sprint-3		USN-9	As a user, I can know the details of thefundamental usage of the application.	5	Low	DIVYA R PRATHISHA A GAYATHIRI N
Sprint-3		USN-10	As a user, I can see the predicted / recognized digits in the application.	5	Medium	DIVYA R PRATHISHA A GAYATHIRI N
Sprint-4	Train the model on IBM	USN-11	As a user, I train the model on IBM and integrate flask/Django with scoring endpoint.	10	High	MAGESWARI S J GAYATHIRI N
Sprint-4	Cloud Deployment	USN-12	As a user, I can access the web applicationand make the use of the product from anywhere.	10	High	MAGESWARI S J GAYATHIRI N

Project Tracker, Velocity & Burndown Chart:

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:

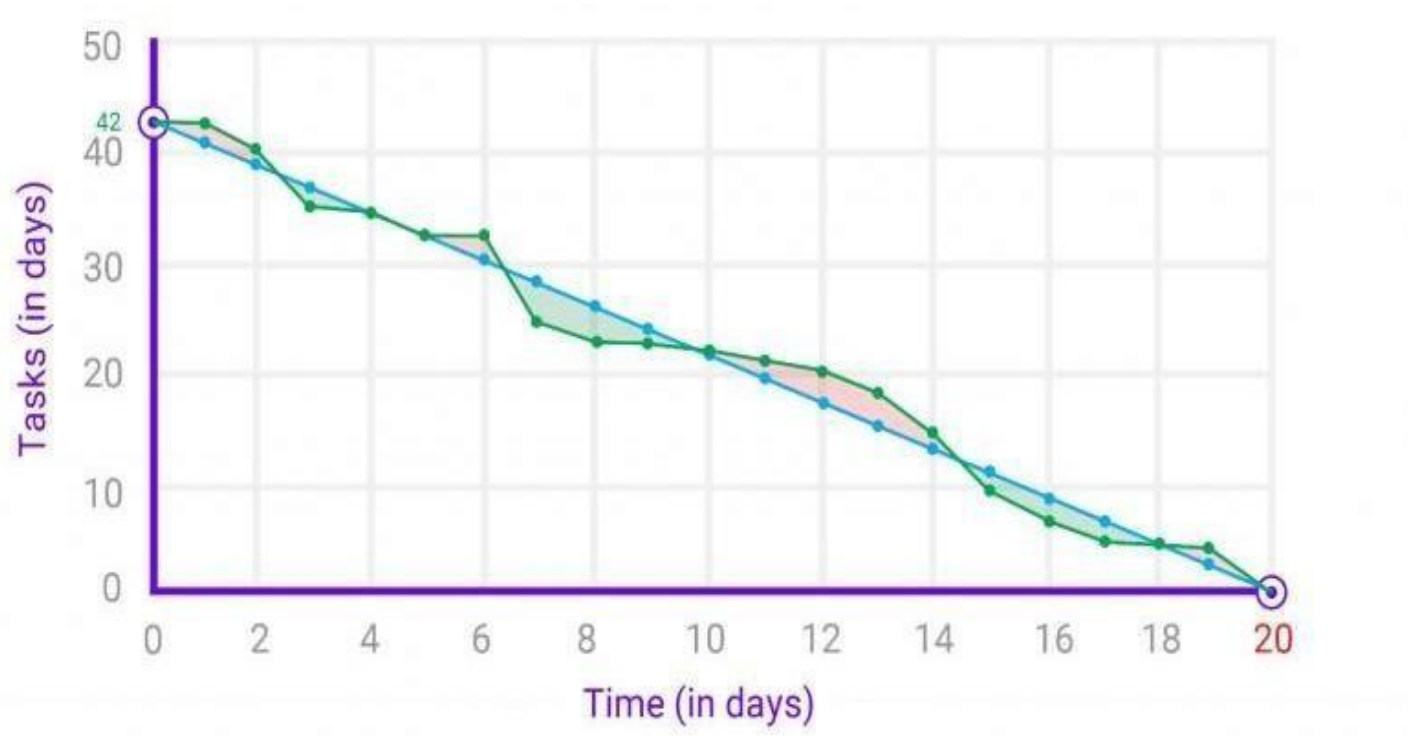
Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

Average Velocity = 20 / 6 = 3.33

Burndown Chart:

A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time

.



6.2 SPRINT DELIVERY SCHEDULE:

S.No.	Milestones	Activities	Timeline
1	Literature Survey	Literature survey on the handwritten digit recognition system and information gathering.	10 September 2022
2.	Empathy Map	Prepared empathy map canvas to capture the user gains and pains.	10 September 2022
3.	Brainstorming and Ideation	Ideas are listed and top 3 ideas are prioritized based on the feasibility and importance.	17 September 2022
4.	Proposed Solution Document	Proposed solution document is prepared which includes Novelty, feasibility of idea, social impact, scalability of solution, etc.	24 September 2022
5.	Problem Solution Fit	Includes customer segments and customer constraints, the problem root cause and jobs to be done.	1 October 2022
6.	Solution Architecture	From data collection to digit recognition by the web application are represented in architectural diagram.	1 October 2022
7.	Customer Journey Map	Prepare Customer Journey maps to understand user interactions and	8 October 2022

		experiences with the application	
8.	Functional Requirement Document	Functional requirements and non functional requirements alike scalability and accuracy are described.	15 October 2022
9.	Data Flow Diagrams	Data flow diagram and user stories are prepared and four sprint phases are described.	15 October 2022
10.	Technology Architecture	Technical flow graphs are created and the functions of technical stacks are defined.	15 October 2022

CODING AND SOLUTIONING

7.1 FEATURE 1:

IMPORTING REQUIRED LIBRARIES:

Importing Required Libraries

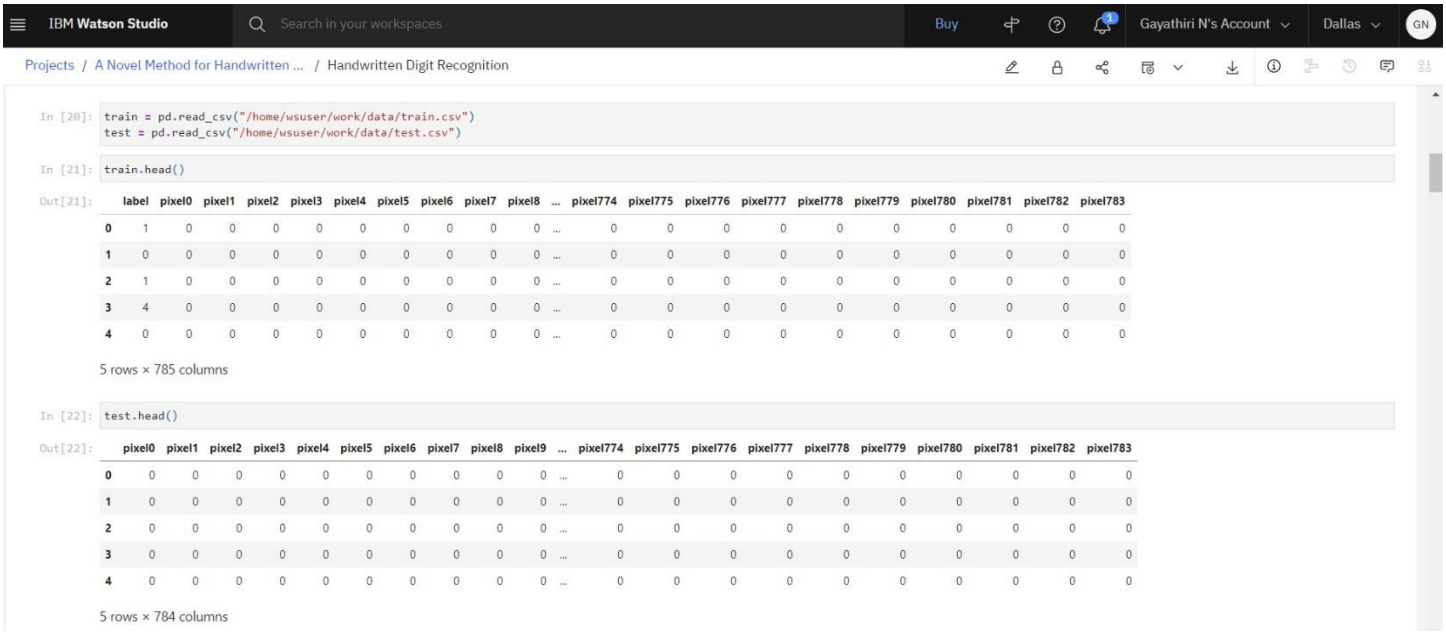
In [1]:
```
import numpy as np
import pandas as pd
import random
import tensorflow as tf
import keras
from keras import layers
keras.backend.set_image_data_format('channels_last')
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
```

LOADING DATA:

In [2]:
```
import os, types
import pandas as pd
from botocore.client import Config
import ibm_boto3

def __iter__(self): return 0

# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.
# You might want to remove those credentials before you share the notebook.
cos_client = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='l1SLNKuyTOTvyxuymCK02Yj3GQDrlAxjSLgrZRYD5R2G',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.private.us.cloud-object-storage.appdomain.cloud')

bucket = 'handwrittendigitrecognition-donotdelete-pr-yvxxug6r0c2f8y'
object_key = 'data.zip'

streaming_body_1 = cos_client.get_object(Bucket=bucket, Key=object_key)['Body']

# Your data file was loaded into a botocore.response.StreamingBody object.
# Please read the documentation of ibm_boto3 and pandas to learn more about the possibilities to load the data.
# ibm_boto3 documentation: https://ibm.github.io/ibm-cos-sdk-python/
# pandas documentation: http://pandas.pydata.org/
```

In [3]:
```
from io import BytesIO
import zipfile
unzip=zipfile.ZipFile(BytesIO(streaming_body_1.read()),'r')
file_paths=unzip.namelist()
for path in file_paths:
    unzip.extract(path)
```

DATA PREPROCESSING:



MODEL BUILDING:

```
def CNN():
    model = keras.Sequential()
    # CONV > CONV > BN > RELU > MAXPOOLING > DROPOUT
    model.add(layers.Conv2D(32, (3, 3), (1, 1), padding='valid', input_shape=(28, 28, 1), name='conv2d_1_1'))
    model.add(layers.Conv2D(32, (3, 3), (1, 1), padding='same', name='conv2d_1_2'))
    model.add(layers.BatchNormalization(name='bn_1'))
    model.add(layers.Activation('relu', name='relu_1'))
    model.add(layers.MaxPooling2D((2, 2), (2, 2), padding='valid', name='mp2d_1'))
    model.add(layers.Dropout(0.2, name='drop_1'))
    # CONV > CONV > BN > RELU > MAXPOOLING > DROPOUT
    model.add(layers.Conv2D(64, (3, 3), (1, 1), padding='valid', name='conv2d_2_1'))
    model.add(layers.Conv2D(64, (3, 3), (1, 1), padding='same', name='conv2d_2_2'))
    model.add(layers.BatchNormalization(name='bn_2'))
    model.add(layers.Activation('relu', name='relu_2'))
    model.add(layers.MaxPooling2D((2, 2), (2, 2), padding='valid', name='mp2d_2'))
    model.add(layers.Dropout(0.2, name='drop_2'))
    # FLATTEN > DENSE > CLASSIFICATION
    model.add(layers.Flatten())
    model.add(layers.Dense(100, activation='relu'))
    model.add(layers.Dense(10, activation='softmax'))

    return model
```

```
model = CNN()
```

```
model.compile(optimizer = "Adam", loss = "CategoricalCrossentropy", metrics = "accuracy")
```

```
model.summary()
```

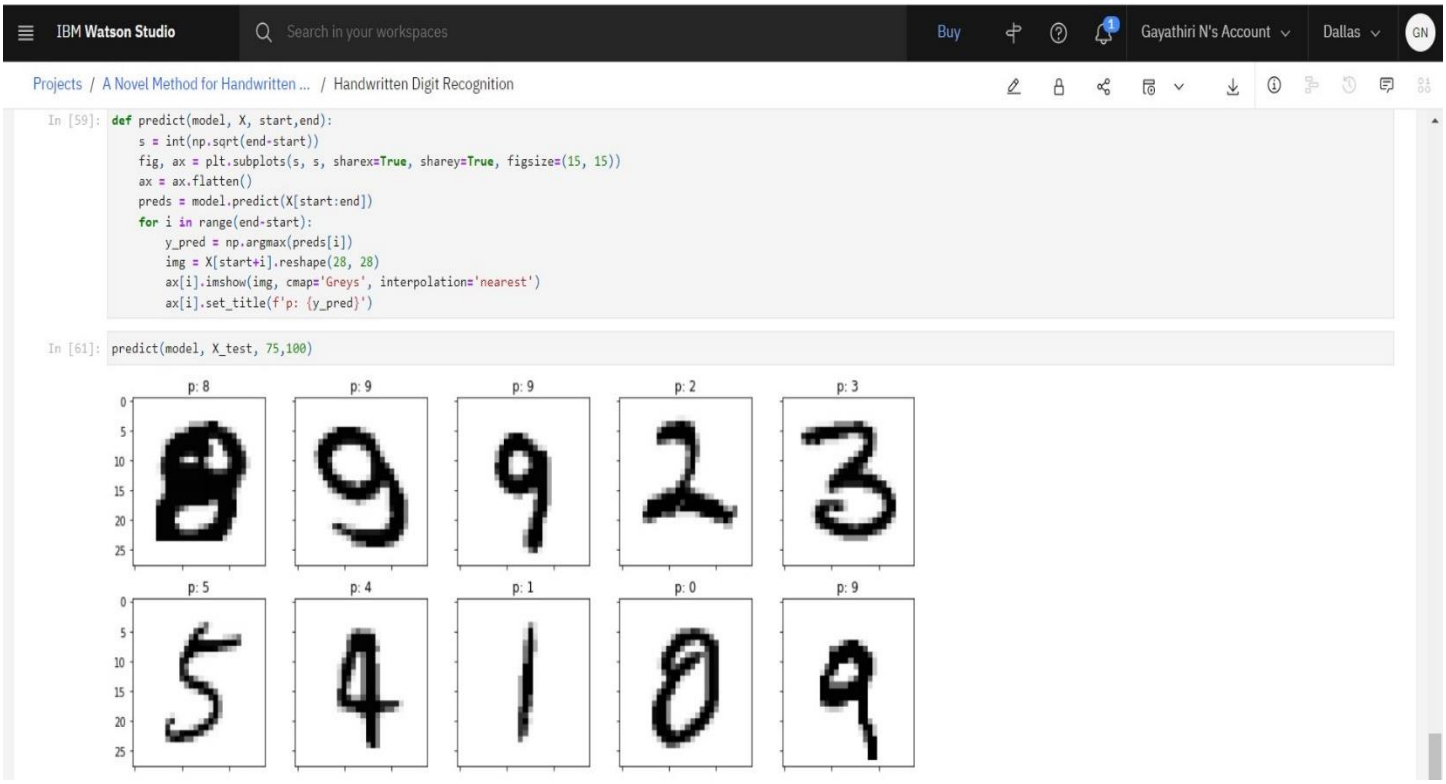
7.2 FEATURE 2:

TRAINING THE MODEL:

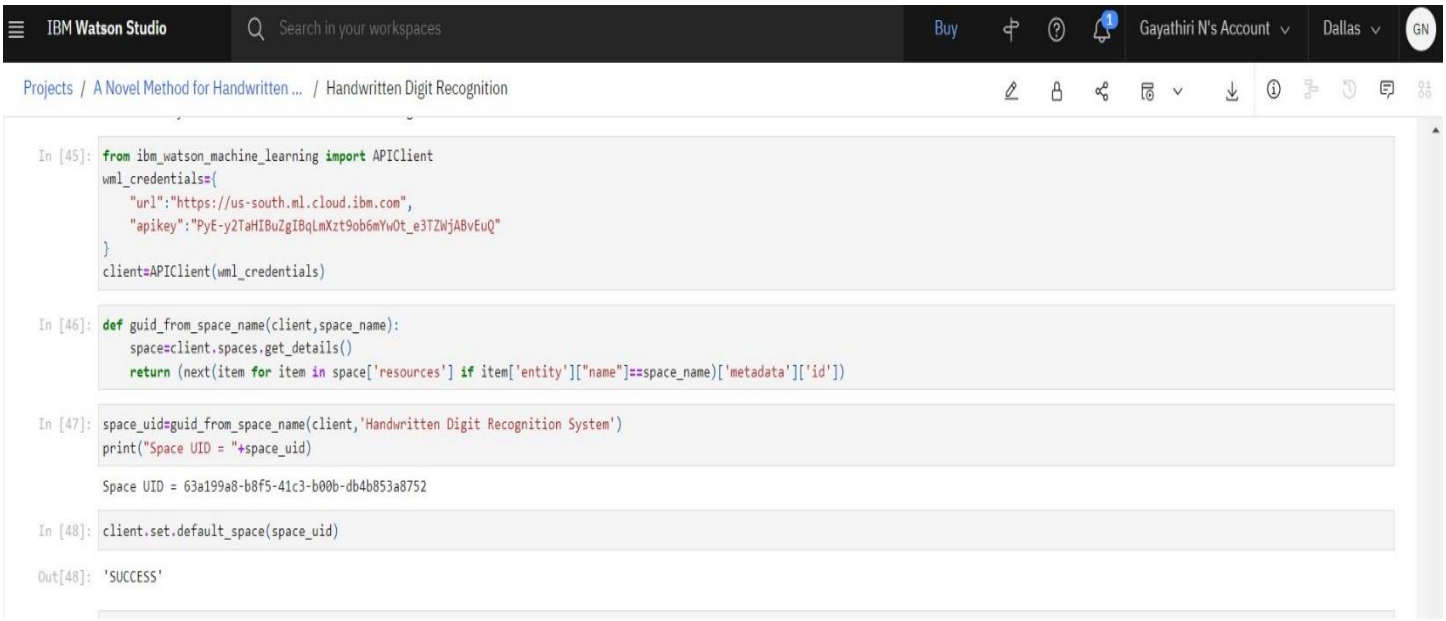
```
training = model.fit(X_train, Y_train, validation_data = (X_val, Y_val), batch_size = 64, epochs = 5, verbose = 1 )
```

```
Epoch 1/5
525/525 [==============================] - 90s 170ms/step - loss: 0.2151 - accuracy: 0.9339 - val_loss: 0.0720 - val_accuracy: 0.9771
Epoch 2/5
525/525 [==============================] - 89s 170ms/step - loss: 0.0715 - accuracy: 0.9778 - val_loss: 0.0762 - val_accuracy: 0.9767
Epoch 3/5
525/525 [==============================] - 88s 167ms/step - loss: 0.0538 - accuracy: 0.9829 - val_loss: 0.0660 - val_accuracy: 0.9805
Epoch 4/5
525/525 [==============================] - 89s 169ms/step - loss: 0.0458 - accuracy: 0.9850 - val_loss: 0.0416 - val_accuracy: 0.9889
Epoch 5/5
525/525 [==============================] - 89s 169ms/step - loss: 0.0388 - accuracy: 0.9878 - val_loss: 0.0578 - val_accuracy: 0.9832
```

TESTING THE MODEL:



IBM CLOUD DEPLOYMENT:



TESTING

8.1 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** 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	13	4	2	5	24
Duplicate	1	0	0	2	3
External	2	3	1	1	7
Fixed	4	6	4	10	24
Not Reproduced	1	1	1	1	4
Skipped	1	2	0	1	4
Won't Fix	0	5	2	4	11
Totals	22	21	10	24	77

3.Test Case Analysis

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

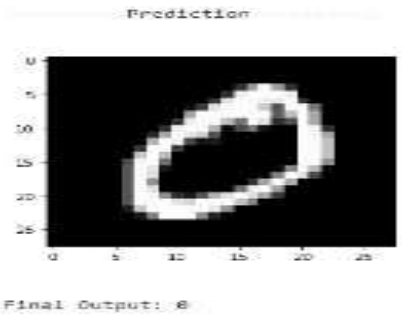
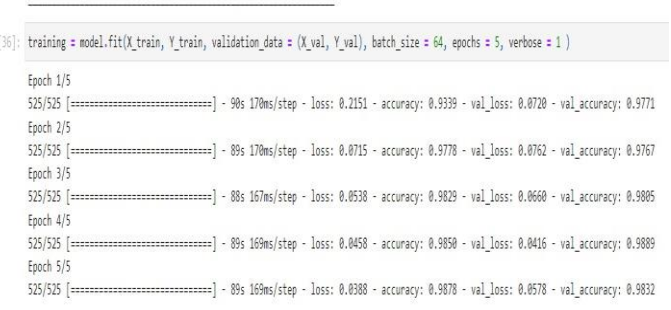
Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Security	2	0	0	2
Performance	3	0	0	3
Exception Reporting	9	0	0	9

RESULTS

9.1 PERFORMANCE METRICS:

DATE	10 TH NOVEMBER 2022
TEAM ID	PNT2022TMID35466
PROJECT	A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION
MAX MARKS	10 MARKS

Model Performance Testing:

	Parameter	Values	Screenshot
1.	**Model Summary**	Loss: 0.0181 Val_loss:0.1560	
2.	**Accuracy**	Training Accuracy - 97.87 Validation Accuracy -95.48	

ADVANTAGES AND DISADVANTAGES

ADVANTAGES:

- Reduces manual work
- Backups
- More accurate than average human
- Capable of handling a lot of data
- Can be used anywhere from any device

DISADVANTAGES:

- Cannot handle complex data
- Low retention
- All the data must be in digital format
- Requires a high performance server for faster predictions
- Prone to occasional errors

CONCLUSION:

This project demonstrated a web application that uses machine learning to recognise handwritten numbers. Flask, HTML, CSS, JavaScript, and a few other technologies were used to create this project. The model predicts the handwritten digit using a CNN network. During testing, the model achieved a 99.61% recognition rate. The proposed project is scalable and can easily handle a huge number of users. Since it is a web application, it is compatible with any device that can run a browser. This project is extremely useful in real-world scenarios such as recognizing number plates of vehicles, processing bank cheque amounts, numeric entries in forms filled up by hand (tax forms) and so on. There is so much room for improvement, which can be implemented in subsequent versions.

APPENDIX

GITHUB: https://github.com/IBM-EPBL/IBM-Project-1795-1658413978

PROJECT DEMO LINK: Demo video- Novel method for handwritten digit recognition