PNT2022TMID35898

A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

PROJECT REPORT

HX5001

NALAIYA THIRAN PROJECT BASED LEARNING ON PROFESSIONAL READINESS FOR INNOVATION, EMPLOYMENT AND ENTERPRENEURSHIP

Submitted By

TEAM ID: PNT2022TMID35898

PREM KUMAR K 2019504563
MEDIKONDURU PRAVALLIKA 2019504548
SELVAGANAPATHY C 2019504583
GOWTHAM K 2019504523

Under the Guidance of

INDUSTRY MENTOR : PRADEEPTHI
FACULTY MENTOR : Dr. C SRIDEVI
FACULTY EVALUATOR : Dr. G.KAVITHA

NOVEMBER 2022

INDEX

1. INTRODUCTION

1.1 Project Overview

Handwriting recognition is one of the compelling researches works going on becauseevery individual in this world has their own style of writing. Because everyone in the world has a unique writing style, handwriting identification is one of the fascinating research projects now being conducted. It is the ability of a computer to automaticallyrecognize and comprehend handwritten numbers or letters.

Everything is being digitized to minimize human labour as a result of advancementsin science and technology. Thus, handwritten digit recognition is required in many real-time applications. The MNIST data collection, which contains 70000 handwrittendigits, 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 applicationis developed that allows users to upload pictures of handwritten numbers. The model examines this picture and the detected result is returned to the UI

1.2 Purpose

Digit recognition systems can recognize digits from a variety of sources. It is the classification ability of a computer to detect human handwritten digits from various sources such as photographs, papers, touch screens and classify them among one of the digits from 0-9.

• Handwritten digit recognition system can be used in processing bank check amounts, numeric entries in forms filled up by hand.

• It can be used to handles formatting, performs correct segmentation into characters, and finds the most plausible words.

• It can be used by postal services to identify postal codes, recognize zip codes on mail for postal mail sorting.

• It can be used to read form data.

• It can be used to help visually impaired people.

2. LITERATURE SURVEY

2.1 Existing Problem

Handwriting varies from person to person, which differs widely in terms of texture, size, width, orientation and also due to similarity in writing style of numbers makes it even more challenging to recognize. The uniqueness and diversity of each person's handwriting affects the digit's structure and look.

2.2 References

Salvador España-Boquera et al, in this paper hybrid Hidden Markov Model (HMM) model is proposed for recognizing unconstrained offline handwritten texts. In this, the structural part of the optical model has been modelled with Markov chains, and a Multilayer Perceptron is used to estimate the emission probabilities. In this paper, different techniques are applied to remove slope and slant from handwritten text and to normalize the size of text images with supervised learning methods. The key features of this recognition system were to develop a system having high accuracy in preprocessing and recognition, which are both based on ANNs.

R. Bajaj, L. Dey, S. Chaudhari et al, employed three different kinds of features, namely, the density features, moment features and descriptive component features for classification of Devanagari Numerals. They proposed multi classifier connectionist architecture for increasing the recognition reliability and they obtained 89.6% accuracy for handwritten Devanagari numerals.

Renata F. P. Neves has proposed SVM based offline handwritten digit recognition. Author's claim that SVM outperforms the Multilayer perceptron classifier. Experiment is carried out on NIST SD19 standard dataset. Advantage of MLP is that it is able to segment non-linearly separable classes. However, MLP can easily fall into a region of local minimum, where the training will stop assuming it has achieved an optimal point in the error surface. Another hindrance is defining the best network architecture to solve the problem, considering the number of layers and the number of perceptrons in each hidden layer. Because of these disadvantages, a digit recognizer using the MLP structure may not produce the desired low error rate.

Yoshimasa Kimura presented a work on how to select features for Character Recognition Using Genetic Algorithms. The author proposes a novel method of feature selection for character recognition using genetic algorithms (GA). The proposed method selects only the genes for which the recognition rate of training samples exceeds the predetermined threshold as a candidate of the parent gene and adopts a reduction ratio in the number of features used for recognition as the fitness value.

M. Hanmandlu, O.V. Ramana Murthy has presented in their study the recognition of handwritten Hindi and English numerals by representing them in the form of exponential membership functions which serve as a fuzzy model. The recognition is carried out by modifying the exponential membership functions fitted to the fuzzy sets. These fuzzy sets are derived from features consisting of normalized distances obtained using the Box approach. The membership function is modified by two structural parameters that are estimated by optimizing the entropy subject to the attainment of membership function to unity. The overall recognition rate is found to be 95% for Hindi numerals and 98.4% for English numeral.

Ragha & Sasikumar describes a system for Kannada characters. In this paper, the moment features are extracted from the Gabor wavelets of preprocessed images of 49 characters. The comparison of moments features of 4 directional images with original images are tested on Multi-Layer Perceptron with Back Propagation Neural Network. The average performance of the system with these two features together is 92%.

2.3 Problem Statement Definition

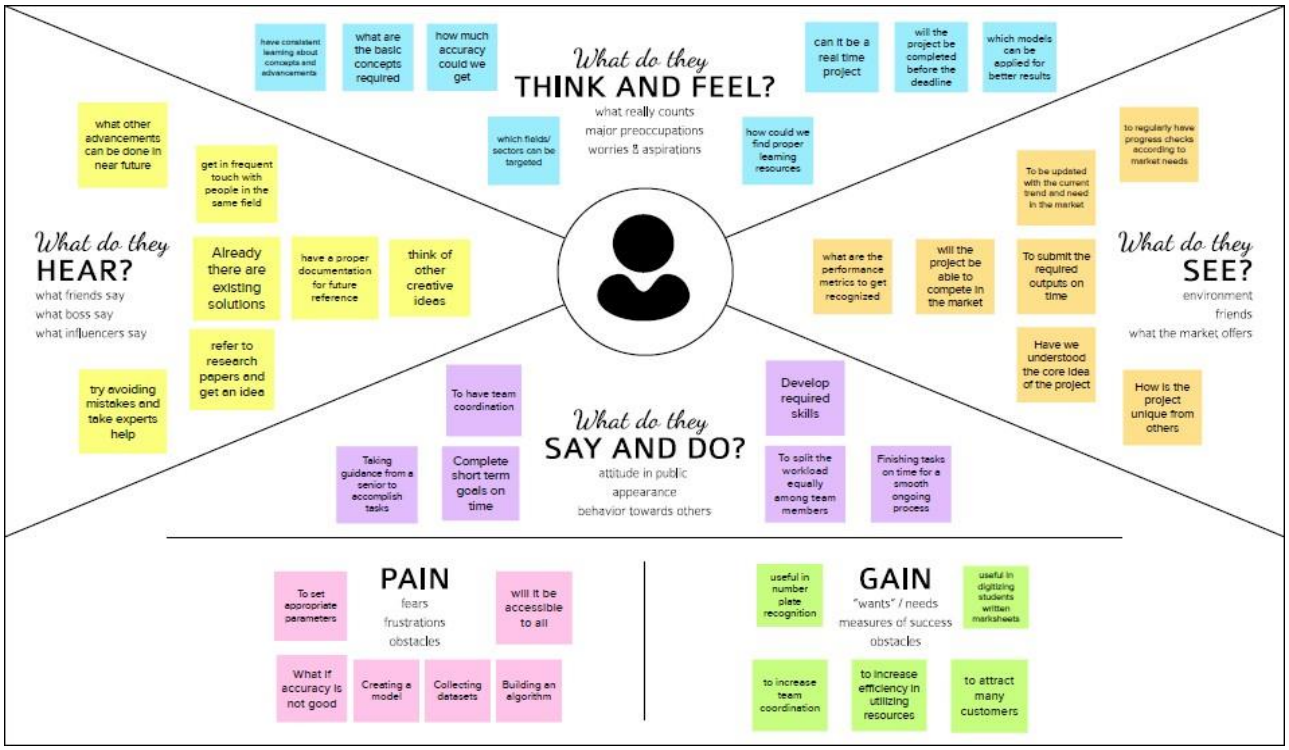
The handwritten digit recognition is the capability of computer applications to recognize the human handwritten digits. It is a hard task for the machine because handwritten digits are not perfect and can be made with many different shapes and sizes. The handwritten digit recognition system is a way to tackle this problem which uses the image of a digit and recognizes the digit present in the image. Convolutional Neural Network model created using PyTorch library over the MNIST dataset to recognize handwritten digits.

The task is to classify a given image of a handwritten digit into one of 10 classes representing integer values from 0 to 9, inclusively. It is a widely used and deeply understood dataset and, for the most part, is “solved.”

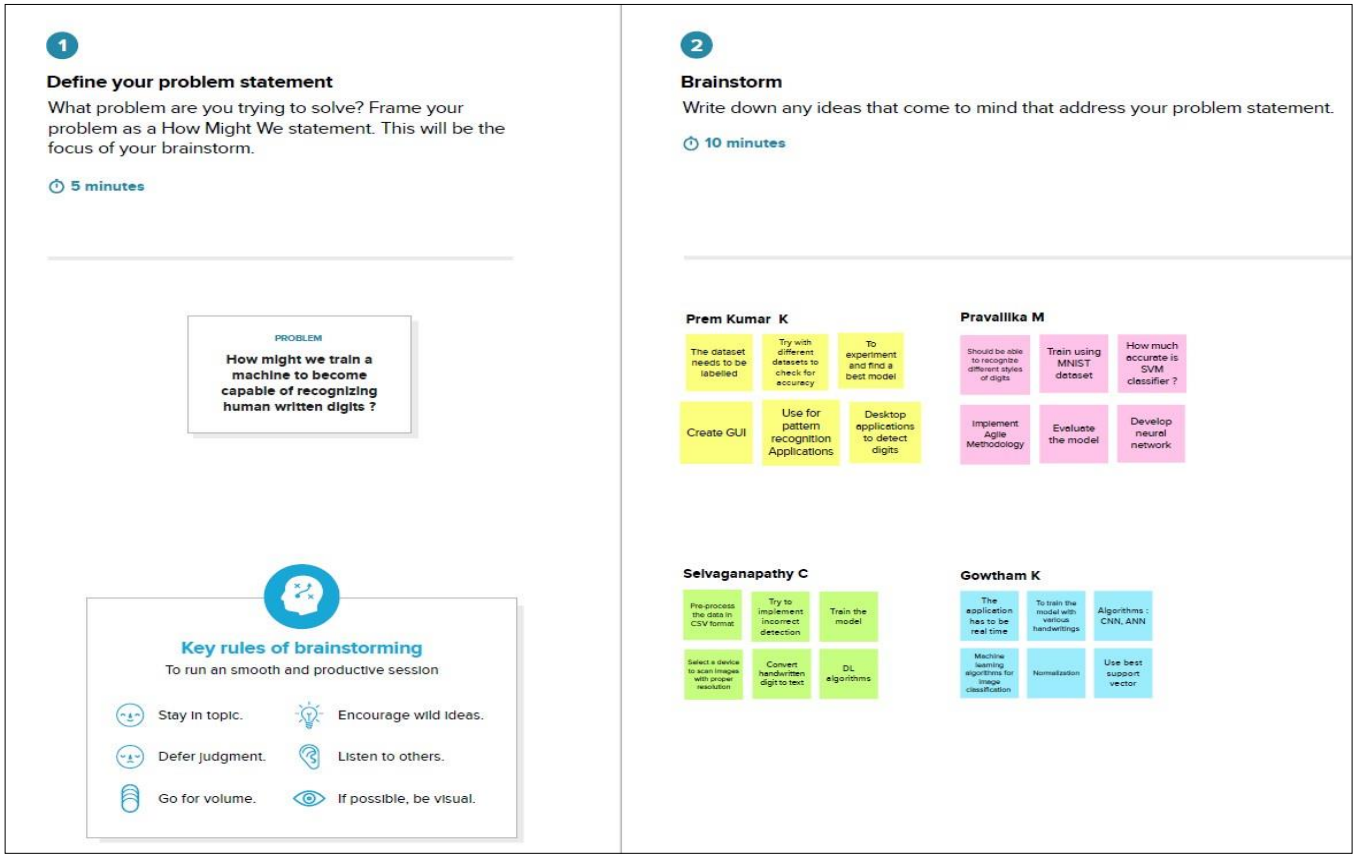
3. IDEATION AND PROPOSED SOLUTION

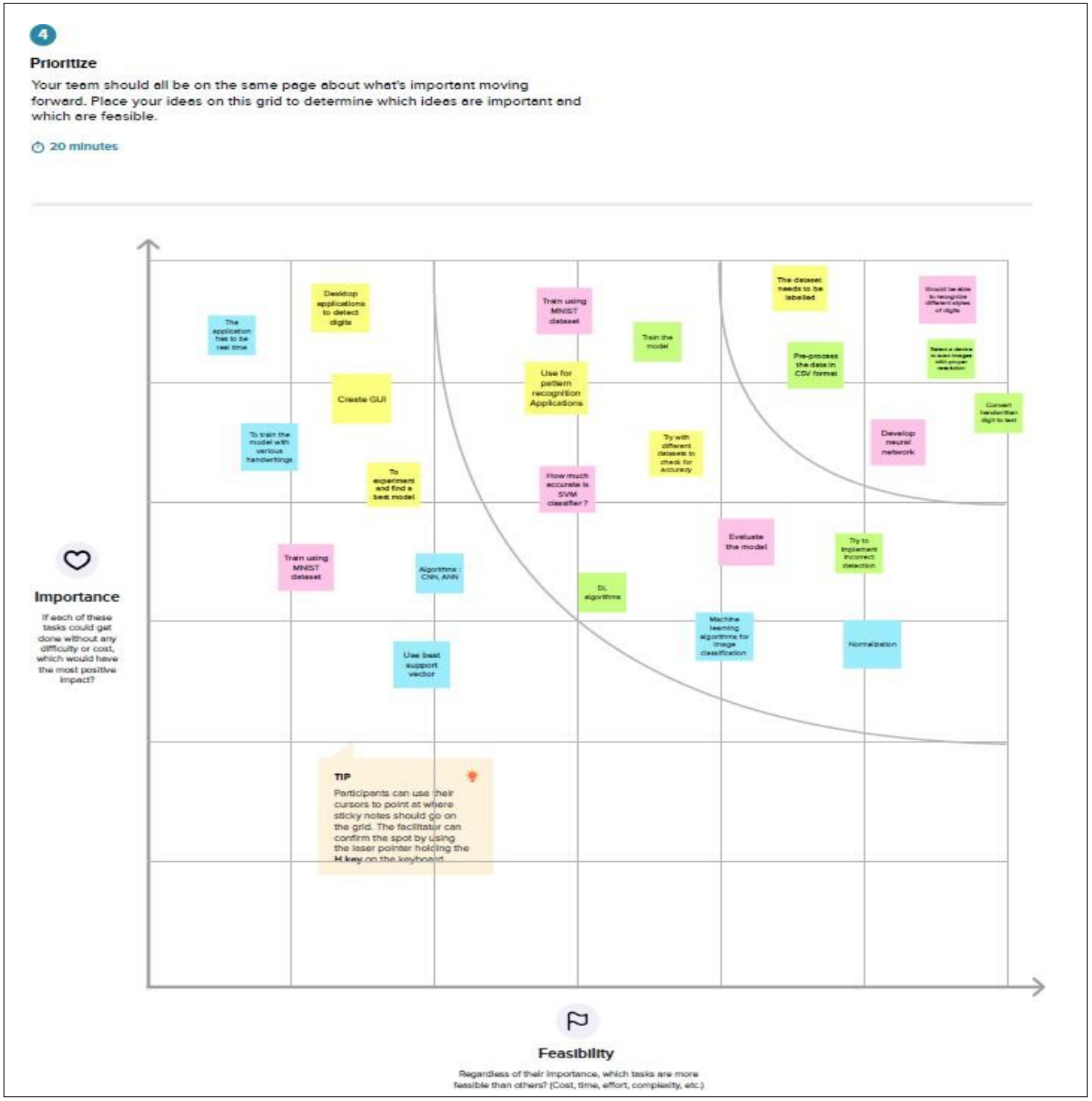
3.1 Empathy Map Canvas

An empathy map is a simple and easy to understand chart that helps to know about the insights of the customer.



3.2 Ideation and Brain Storming





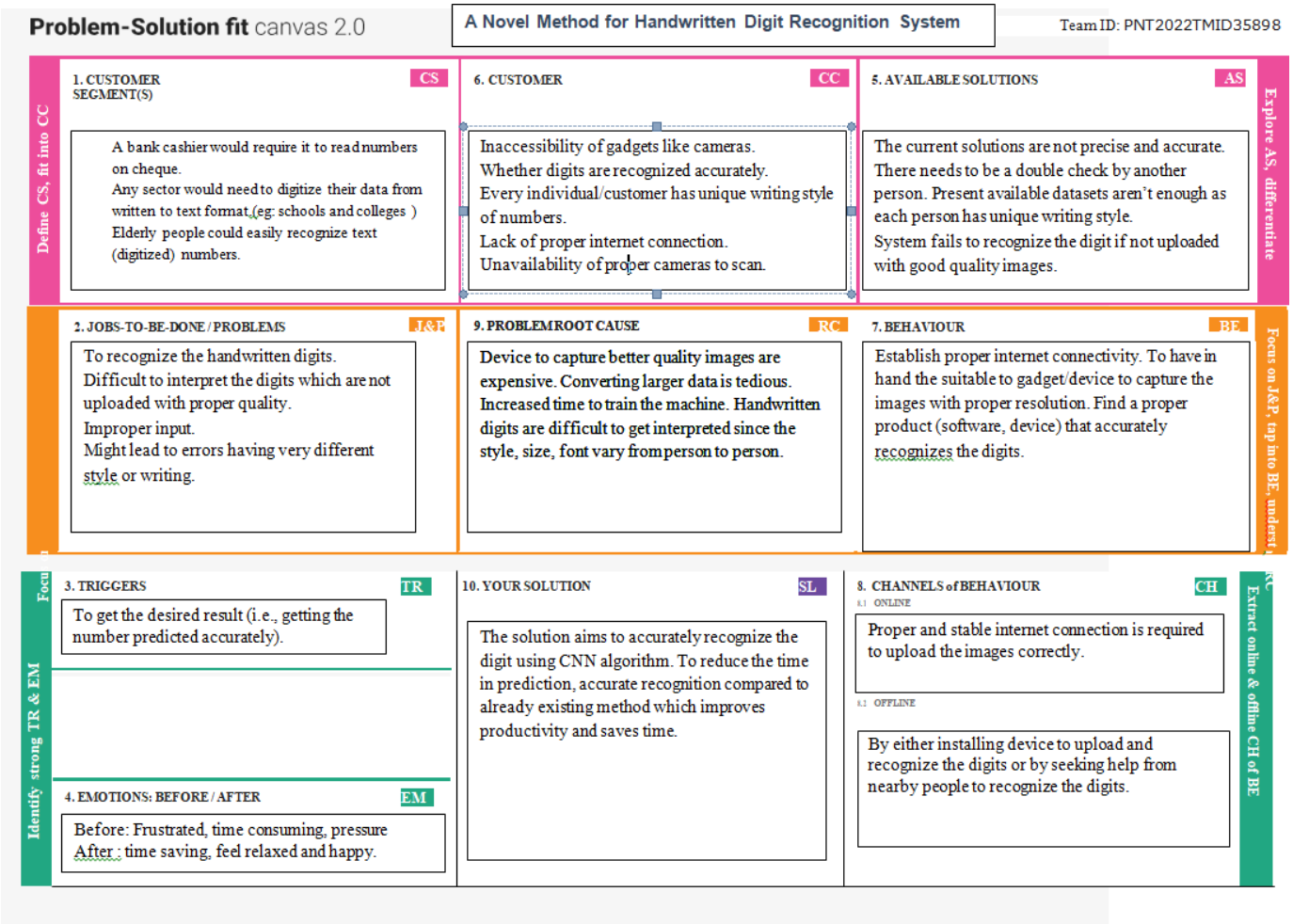
3.3 Proposed Solution

S.No	Parameter	Description
1.	Problem Statement (Problem to be solved)	**Statement:** The handwritten digit recognition is the capability of computer applications to recognize the human handwritten digits. **Description:** It is a challenging task for the machine because handwritten digits are not perfect and can be made with many different shapes and sizes.

2.	Idea / Solution description	1. It is the capability of a computer to fetch the mortal handwritten integers from various sources like images, papers and touch defenses. 2. It allows the user to translate all those signatures and notes into electronic words in atext document format and this data only requires far less physical space than the storage of the physical copies. 3. Build a machine learning model using neural networks and CNN that captures similar patterns from image dataset.
3.	Novelty / Uniqueness	1. Accurately recognize the digits rather than recognizing all the characters like OCR. 2. GAN layers can be used for better accuracy in the handwritten digit recognition system. Normalization can be used for better efficiency.
4.	Social Impact / Customer Satisfaction	1. Artificial Intelligence developed the app called handwritten digit Recognizer. 2. It converts the written word into digital approximations and utilizes complex algorithms to identify characters before churning out a digital approximation. 3. Old people who have eye sight issues with handwritten digits can use this system to recognize the handwritten digits correctly.
5.	Business Model (Revenue Model)	1. This system can be integrated with traffic surveillance cameras to recognize the vehicle's number plates for effective traffic management. 2. Can be integrated with Postal system to identify and recognize the pin-code details easily. 3. In banking sectors handwritten numbers are involved like account numbers, figures of cash and checks. By this system we can avoid human mistakes.

6.	Scalability of the Solution	1. Ability to recognize digits in more noisy environments. 2. There is no limit to the number of digits it can be recognized.

3.4 Proposed Solution Fit



4. REQUIREMENT ANALYSIS

4.1 Functional Requirements

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirements	Description
FR-1	User input	Handwritten Digit as a input.
FR-2	Pre-processing	The role of the Pre- processing steps is it performs various tasks on input image and train the model.
FR-3	Analysis	With the help of pre- trained model, Analysis the current hand written digitwith the help of DL and CNN algorithm.
FR-4	Prediction	With the power of pre- trained data and model the output becomes more accurate.

4.2 Non-Functional Requirements

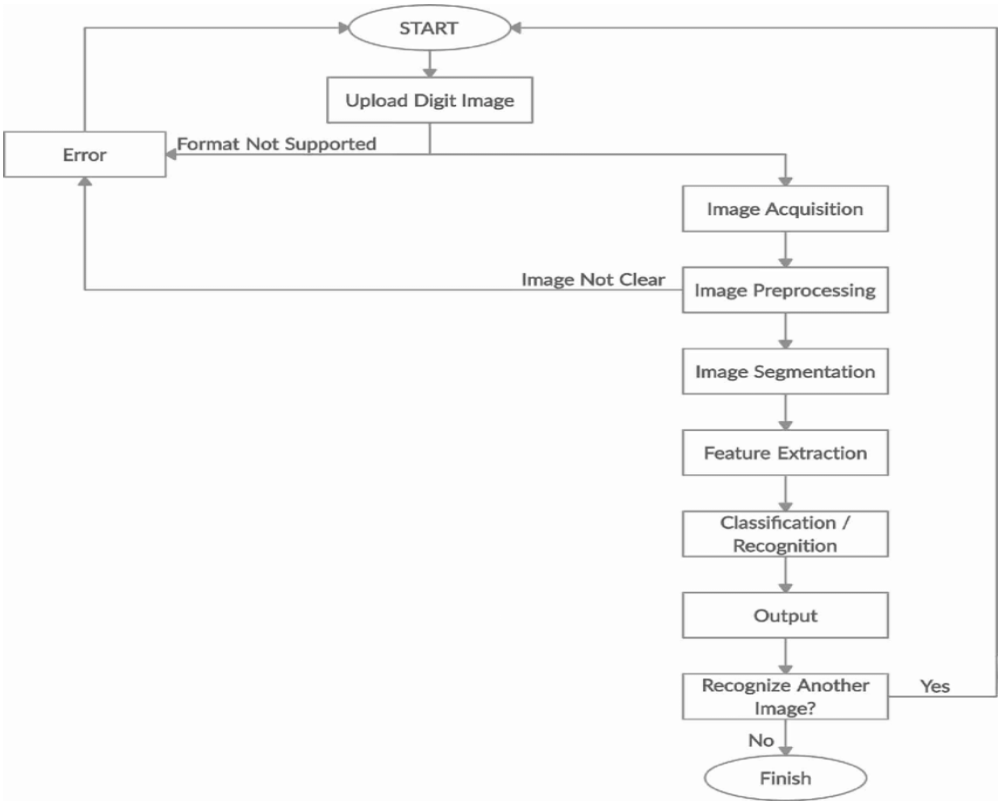
Following are the Non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirements	Description
NFR-1	Usability	Handwritten digit recognition is the ability of acomputer system to recognize the handwritten inputs like digits, characters etc.
NFR-2	Security	The applications where this handwritten digit recognition can be used are Banking sector whereit can be used to maintain the security pin numbers, it can be also used for blind peoples by using sound as output.

NFR-3	Reliability	This software will work reliably for low resolution images and not for graphical images.
NFR-4	Performance	Most standard implementations of neural networks achieve an accuracy of ~ (98– 99) percent in correctly classifying the handwritten digits. Beyond this number, every single decimal increase in the accuracy percentage is hard.
NFR-5	Availability	This is probably one of the most popular datasets among machine learning and deep learning enthusiasts. The MNIST dataset contains 60,000 training images of handwritten digits from zero to nine and 10,000 images for testing. So, the MNIST dataset has 10 different classes.
NFR-6	Scalability	Hand written digit Recognition helps thousands and thousands of users to help them the digit better.

5. PROJECT DESIGN

5.1 Data Flow Diagram



5.2 Solution and Technical Architecture

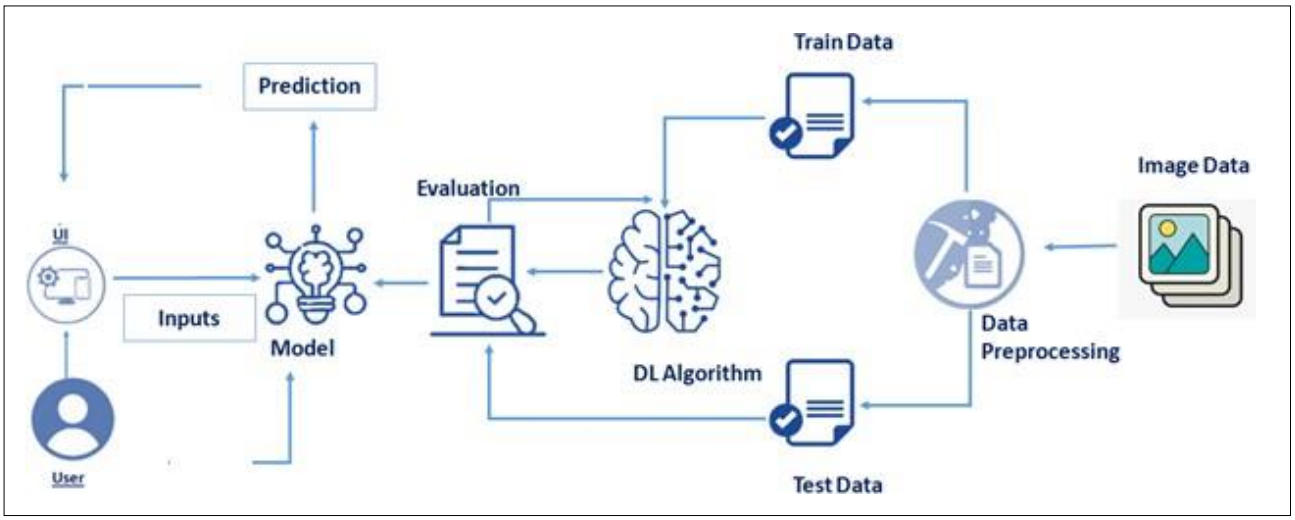


Table-1: Components & Technologies

S.No	Component	Description	Technology
1.	Website	User interacts with the prediction model through website to predict the fuel consumption	HTML, CSS, JavaScript / Angular JS / React JSetc.
2.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc.
3.	API	Used to extend the service to other applications	Flask Application

4.	DL and CNN algorithm	Using Neural networks to compute large amount data that trains the model	Deep Learning, Machine Learning, CNN.
5.	Data Pre-processing	Process of converting the raw data set into a clean data set	Dimensionality reduction
6.	Machine learning Model	This model is developed to predict the fuel consumption using ML algorithms	Sklearn, Algorithms -SVM & MLR
7.	Database	Data is pre-processed and used for training the model which is then used for prediction	Pandas, NumPy, Matplotlib

Table-2: Application Characteristics

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Used to freely access the public code	Angular JS / React JS
2.	Scalable Architecture	Justify the scalability of architecture (3 – tier, Micro-services)	Technology used
3.	Security Implementations	List all the security / access controls implemented and use of firewalls etc.	e.g., SHA-256, Encryptions, IAM Controls, OWASP etc.
4.	Availability	The application will be available in all regions	Distributed servers
5.	Performance	Higher efficiency of performance. The application can give response to requests within 5 sec	Technology used

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 the application with Gmail	Medium	Sprint-2
	Login	USN-5	As a user, I can log into the application By entering email & password	I can login to the application	High	Sprint-1
	Home	USN-6	As a user, I can view the application's home page where I can read the instructions to use this application	I can read instructions also and the home Page is user-friendly.	Low	Sprint-1
	Upload Image	USN-7	As a user, I can able to input the images of digital documents to the application	As a user, I can able to input the images of digital documents tothe application	High	Sprint-3
	Predict	USN-8	As a user I can able to get the recognized digit as output from the images of digital documents or images	I can access the recognized digits from digital document or images	High	Sprint-3
		USN-9	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.	Medium	Sprint-4
Customer (Web user)	Accessibility	USN-10	As a user, I can use the web application virtually anywhere.	I can use the application in any device with a browser	Medium	Sprint-4

6. PROJECT PLANNING AND SCHEDULING

6.1. Sprint Planning and Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story/Task	Story Points	Priority	Team Members
Sprint-1	Data Collection	USN-1	As a user, I can collect the dataset from MNIST.	5	High	Selvaganapathy. C Gowtham. K
Sprint-1	Data Preprocessing	USN-2	As a user, I can load the data set, handling the missing data, scaling and split data into train and test.	5	High	Prem Kumar. k Pravallika. M Selvaganapathy. C Gowtham. K
Sprint-2	Model Building	USN-3	As a user, I can initialize the model, by adding the layers and output layer, train, evaluate, save and test the model.	10	High	Prem Kumar. k Pravallika. M
Sprint-3	Application Building	USN-4	As a user, I create a Html file, build a python code and run the app and show casting the prediction.	10	High	Prem Kumar. k Pravallika. M Selvaganapathy. C Gowtham. K
Sprint-4	Train the model on IBM	USN-5	As a user, I train the model on IBM and integrate flask with scoring end point.	10	Medium	Prem Kumar. k Pravallika. M Selvaganapathy. C Gowtham. K

6.2 Sprint Delivery Schedule

Sprint	Total story Points	Duration	Sprint start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	10	5 Days	24 Oct 2022	29 Oct 2022	10	29 Oct 2022
Sprint-2	10	5 Days	31 Oct 2022	05 Nov 2022	10	05 Nov 2022
Sprint-3	10	5 Days	07 Nov 2022	12 Nov 2022	10	12 Nov 2022
Sprint-4	10	5 Days	14Nov 2022	19 Nov 2022	10	19 Nov 2022

6.3 Reports from JIRA

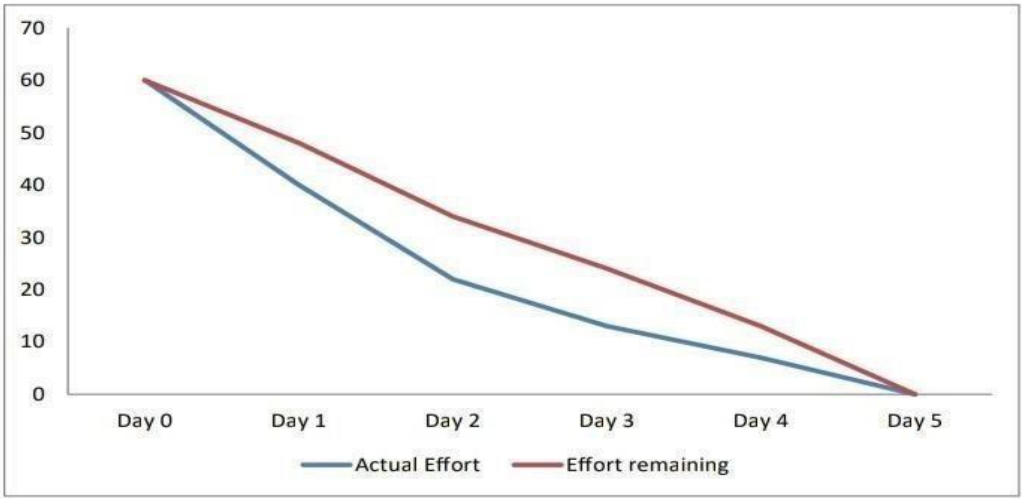
Velocity:

Imagine we have 5-day sprint duration, and the velocity of the team is 10 (points per sprint). Let's calculate the team's average velocity (AV)per iteration unit (story points per day)

AV=Sprint Duration/Velocity=10/5=2

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 suchas Scrum. However, burn down charts can be applied to any project containing measurable progress over time.



7. CODING AND SOLUTIONING

[i] Feature 1

(i) Importing Libraries

In [1]:
```
#Import all Necessary Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from tensorflow.keras.datasets import mnist
```

(ii) Loading Data

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
```

(iii) Analyzing the data

In [3]:
```
x_train
```

Out[3]:
```
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, 0, 0],
        [0, 0, 0, ..., 0, 0, 0]]], dtype=uint8)
```

(iv) Apply one hot encoding

```
In [14]: y_cat_test = to_categorical(y_test,num_classes=10)

In [15]: y_cat_train = to_categorical(y_train,10)

In [16]: one_img.max(),one_img.min()

Out[16]: (255, 0)
```

(v) Reshaping the data

```
In [19]: #reshape the scaled data
         x_train = x_train.reshape(60000,28,28,1)
         x_test = x_test.reshape(10000,28,28,1)

In [20]: x_train.shape,x_test.shape

Out[20]: ((60000, 28, 28, 1), (10000, 28, 28, 1))
```

[ii] Feature 2

(i) Adding CNN layers

```
In [22]: #Adding Layers
         model = Sequential()
         model.add(Conv2D(filters=32, kernel_size=(4,4),activation='relu',input_shape=(28,28,1)))
         model.add(MaxPool2D(pool_size=(2,2)))
         model.add(Flatten())
         model.add(Dense(128,activation='relu'))
         model.add(Dense(10,activation='softmax'))
```

(ii) Compile Model

```
In [23]: #Compilation of the model
         model.compile(loss='categorical_crossentropy',optimizer='Adadelta',metrics=['accuracy'])
```

(iii) Train Model

```
model.fit(x_train,y_cat_train,
          epochs=15,
          validation_data=(x_test,y_cat_test),
          callbacks=[early_stop])
```

```
Epoch 1/15
1875/1875 [==============================] - 7s 4ms/step - loss: 0.1093 - accuracy: 0.9695 - val_loss: 0.1068 - val_accuracy: 0.9707
Epoch 2/15
1875/1875 [==============================] - 6s 3ms/step - loss: 0.1089 - accuracy: 0.9700 - val_loss: 0.1064 - val_accuracy: 0.9709
Epoch 3/15
1875/1875 [==============================] - 6s 3ms/step - loss: 0.1085 - accuracy: 0.9700 - val_loss: 0.1062 - val_accuracy: 0.9707
Epoch 4/15
1875/1875 [==============================] - 7s 4ms/step - loss: 0.1081 - accuracy: 0.9701 - val_loss: 0.1060 - val_accuracy: 0.9706
Epoch 5/15
1875/1875 [==============================] - 6s 3ms/step - loss: 0.1078 - accuracy: 0.9703 - val_loss: 0.1056 - val_accuracy: 0.9706
Epoch 6/15
1875/1875 [==============================] - 6s 3ms/step - loss: 0.1074 - accuracy: 0.9702 - val_loss: 0.1053 - val_accuracy: 0.9713
Epoch 7/15
1875/1875 [==============================] - 6s 3ms/step - loss: 0.1070 - accuracy: 0.9704 - val_loss: 0.1048 - val_accuracy: 0.9711
Epoch 8/15
1875/1875 [==============================] - 7s 4ms/step - loss: 0.1066 - accuracy: 0.9704 - val_loss: 0.1045 - val_accuracy: 0.9710
Epoch 9/15
1875/1875 [==============================] - 7s 4ms/step - loss: 0.1063 - accuracy: 0.9707 - val_loss: 0.1042 - val_accuracy: 0.9712
Epoch 10/15
1875/1875 [==============================] - 6s 3ms/step - loss: 0.1059 - accuracy: 0.9708 - val_loss: 0.1039 - val_accuracy: 0.9708
Epoch 11/15
1875/1875 [==============================] - 6s 3ms/step - loss: 0.1056 - accuracy: 0.9708 - val_loss: 0.1035 - val_accuracy: 0.9711
Epoch 12/15
1875/1875 [==============================] - 6s 3ms/step - loss: 0.1052 - accuracy: 0.9709 - val_loss: 0.1033 - val_accuracy: 0.9709
Epoch 13/15
1875/1875 [==============================] - 7s 4ms/step - loss: 0.1048 - accuracy: 0.9711 - val_loss: 0.1031 - val_accuracy: 0.9713
Epoch 14/15
1875/1875 [==============================] - 6s 3ms/step - loss: 0.1045 - accuracy: 0.9712 - val_loss: 0.1027 - val_accuracy: 0.9712
Epoch 15/15
1875/1875 [==============================] - 7s 4ms/step - loss: 0.1041 - accuracy: 0.9713 - val_loss: 0.1025 - val_accuracy: 0.9712
```

(iv) Test Model

In [48]:
```
model.evaluate(x_test,y_cat_test,verbose=0)
      #loss          |       #accuracy
```

Out[48]: `[0.10245479643344879, 0.9711999893188477]`

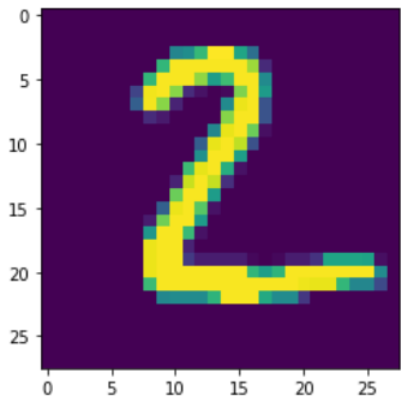
In [54]:
```
my_num = x_test[1]
```

In [55]:
```
classes_x
```

Out[55]: `array([7, 2, 1, ..., 4, 5, 6])`

In [56]:
```
plt.imshow(my_num.reshape(28,28))
```

Out[56]:



(v) Observing Metrics

In [51]:
```
print(classification_report(y_test,classes_x))
```

```
              precision    recall  f1-score   support

           0       0.96      0.99      0.98       980
           1       0.98      0.99      0.99      1135
           2       0.97      0.96      0.97      1032
           3       0.97      0.98      0.97      1010
           4       0.98      0.98      0.98       982
           5       0.98      0.97      0.98       892
           6       0.97      0.97      0.97       958
           7       0.97      0.96      0.97      1028
           8       0.96      0.96      0.96       974
           9       0.97      0.95      0.96      1009

    accuracy                           0.97     10000
   macro avg       0.97      0.97      0.97     10000
weighted avg       0.97      0.97      0.97     10000
```

In [52]:
```
print(confusion_matrix(y_test,classes_x))
```

```
[[ 972    0    1    0    0    1    2    1    3    0]
 [   0 1120    3    1    0    0    3    2    6    0]
 [   5    2  994    8    4    0    4    5    9    1]
 [   1    0    2  985    0    6    0    6    6    4]
 [   1    1    5    0  960    0    5    1    2    7]
 [   5    1    0    5    0  867    5    2    3    4]
 [   8    3    1    1    2    7  932    1    3    0]
 [   2    4   17    5    1    0    0  988    2    9]
 [   6    1    4    8    5    3    6    6  931    4]
 [   8    7    1    5   10    1    0    7    7  963]]
```

8. TESTING

8.1 Test Cases

Test Case ID	Feature Type	Component	Test Scenario	Steps to Execute	Test Data	Expected Result	Actual Result	Status	BUG ID	Executed By
HP_TC_001	UI	Home Page	Verify the UI elements in the Home Page	1)Open the page 2) Check if all theUI elements are displayed in the home page	127.0.0.1:5000	The Home page must be displayed perfectly	Working as expected	Pass		Prem Kumar K Pravallika M Selvganapathy C Gowtham K
HP_TC_002	UI	Home Page	Check if the elements are displayed properly in different screen sizes	1) Open the page in a specific device 2) Check if all the UI elements are displayed properly 3) Repeat the above steps with different device sizes	Screen Sizes- 2560 x 1801	The Home page must be displayed properly in all sizes	The UI is not displayed properly in screen size 2560 x 1801 and 768 x 630	Fail	BUG-HP_TC_001	Prem Kumar K Pravallika M Selvganapathy C Gowtham K
HP_TC_003	Functional	Home Page	Check if user can upload their file	1) Open the page 2) Click on select button 3) Select the input image	Sample1.png	The input image should be uploaded to the application successfully	Working as expected	Pass		Prem Kumar K Pravallika M Selvganapathy C Gowtham K
HP_TC_004	Functional	Home Page	Check if user cannot upload unsupported files	1)Open the page 2) Click on selectbutton 3)Select a random input file	Installer.exe	The application should not allow user to select a non-image file	User is able to upload any file	Fail	BUG-HP_TC_002	Prem Kumar K Pravallika M Selvganapathy C Gowtham K
HP_TC_004	Functional	Home Page	Check if the page redirects to the result page once the input is given	1)Open the page 2) Click on selectbutton 3)Select the input image 4) Check if the page redirects	Sample1.png	The page should redirect to the results page	Working as expected	Pass		Prem Kumar K Pravallika M Selvganapathy C Gowtham K
HP_TC_001	Functional	Backend	Check if all the routes are working properly	1)Go to Home Page 2) Upload the input image 3)Check the results page	Sample1.png	All the routes should properly work	Working as expected	Pass		Prem Kumar K Pravallika M Selvganapathy C Gowtham K
HP_TC_001	Functional	Model	Check if the model can handle various image sizes	1) Open the page in a specific device 2) Upload the input image 3) Repeat the above steps with different input image	Sample1.png Sample1 X5.png Sample1 XL.png	The model should rescale the image and predict the results	Working as expected	Pass		Prem Kumar K Pravallika M Selvganapathy C Gowtham K
HP_TC_002	Functional	Model	Check if the model predicts the digit	1)Open the page 2) Click on selectbutton 3)Select the input image 4) Check the results	Sample1.png	The model should predict the number	Working as expected	Pass		Prem Kumar K Pravallika M Selvganapathy C Gowtham K
HP_TC_003	Functional	Model	Check if the model can handle complex input image	1)Open the page 2) Click on selectbu tton 3)Select the input image 4) Check theresults	Complex Sample.png	The model should predict the number in the complex image	The model fails to identify the digit since the model is not built to handle such data	Fail		Prem Kumar K Pravallika M Selvganapathy C Gowtham K
HP_TC_001	UI	Result Page	Verify UI elements in the Result Page	1)Open the page 2) Click on selectbutton 3)Select the input image 4) Check if all the UI elements are displayed properly	Sample1.png	The result page must be displayed properly	Working as expected	Pass		Prem Kumar K Pravallika M Selvganapathy C Gowtham K
HP_TC_002	UI	Result Page	Check if the result is displayed properly	1)Open the page 2) Click on selectbutton 3)Select the input image 4) Check if theresult is displayed	Sample1.png	The result page should be displayed properly	Working as expected	Pass		Prem Kumar K Pravallika M Selvganapathy C Gowtham K
HP_TC_004	UI	Result Page	Check if the other predictions are displayed properly	1)Open the page 2) Click on selectbutton 3)Select the input image 4) Check if all the other predictions are displayed	Sample1.png	The other predictions should be displayed properly	Working as expected	Pass		Prem Kumar K Pravallika M Selvganapathy C Gowtham K

8.2 User Acceptance Testing

(i) 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	Sub total
By Design	10	8	6	11	35
Duplicate	1	0	3	0	4
External	2	3	2	7	14
Fixed	8	2	4	20	34
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	21	18	19	40	98

(ii) 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
Client Application	20	0	3	17
Security	2	0	0	2
Outsource Shipping	4	0	1	3
Exception Reporting	9	0	0	9
Final Report Output	4	0	2	2
Version Control	2	0	0	2

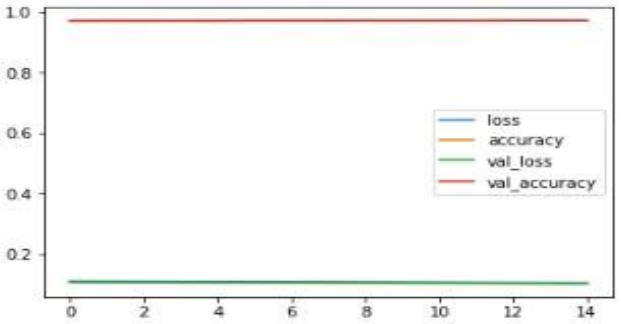
9. RESULTS

9.1 Performance metrics

(i) Model metrics

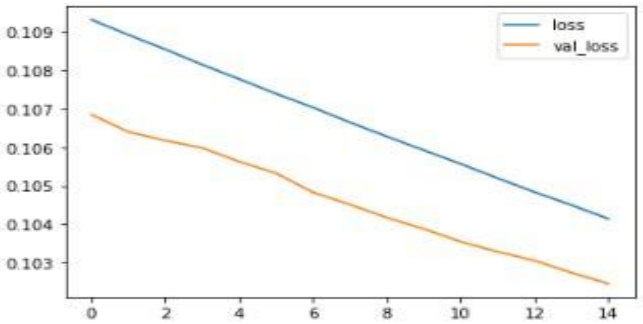
In [45]:
```
metrics.plot()
```

Out[45]:



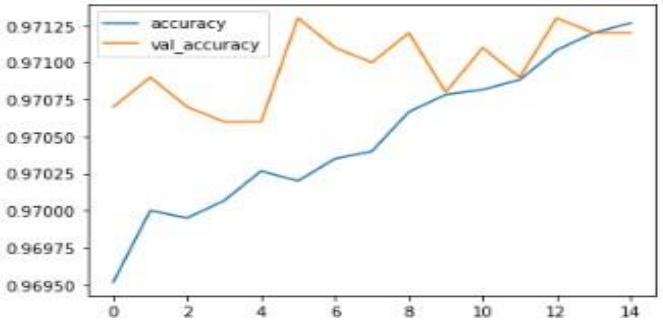
In [46]:
```
metrics[['loss','val_loss']].plot()
```

Out[46]:



In [47]:
```
metrics[['accuracy','val_accuracy']].plot()
```

Out[47]:



(ii) Overall application performance

Metrics:

In [43]:
```
metrics = pd.DataFrame(model.history.history)
```

In [44]:
```
metrics
```

Out[44]:

	loss	accuracy	val_loss	val_accuracy
0	0.109312	0.969517	0.106847	0.9707
1	0.108919	0.970000	0.106399	0.9709
2	0.108544	0.969950	0.106174	0.9707
3	0.108145	0.970067	0.105983	0.9706
4	0.107771	0.970267	0.105627	0.9706
5	0.107390	0.970200	0.105325	0.9713
6	0.107032	0.970350	0.104828	0.9711
7	0.106646	0.970400	0.104511	0.9710
8	0.106278	0.970667	0.104172	0.9712
9	0.105923	0.970783	0.103879	0.9708
10	0.105567	0.970817	0.103546	0.9711
11	0.105196	0.970883	0.103284	0.9709
12	0.104834	0.971083	0.103056	0.9713
13	0.104498	0.971200	0.102740	0.9712
14	0.104140	0.971267	0.102455	0.9712

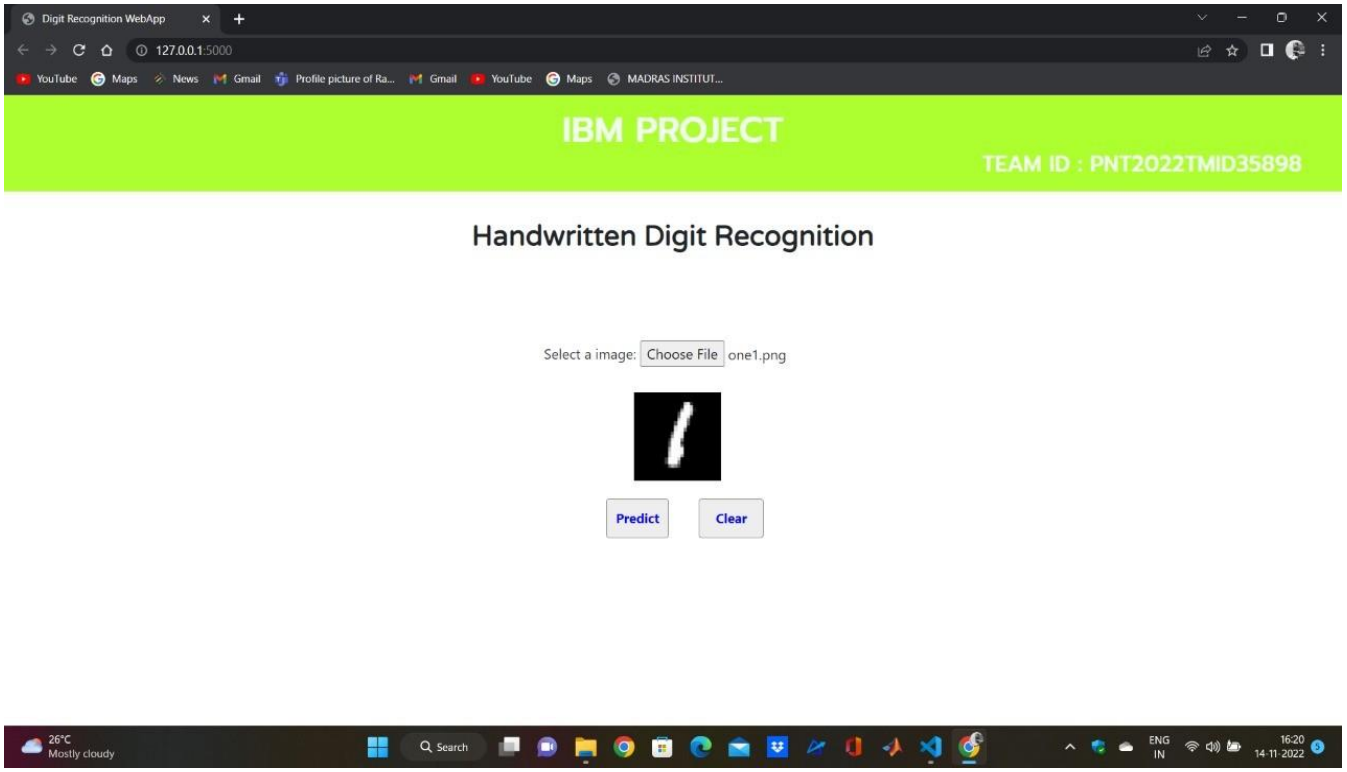
In [51]:
```
print(classification_report(y_test,classes_x))
```

```
              precision    recall  f1-score   support

           0       0.96      0.99      0.98       980
           1       0.98      0.99      0.99      1135
           2       0.97      0.96      0.97      1032
           3       0.97      0.98      0.97      1010
           4       0.98      0.98      0.98       982
           5       0.98      0.97      0.98       892
           6       0.97      0.97      0.97       958
           7       0.97      0.96      0.97      1028
           8       0.96      0.96      0.96       974
           9       0.97      0.95      0.96      1009

    accuracy                           0.97     10000
   macro avg       0.97      0.97      0.97     10000
weighted avg       0.97      0.97      0.97     10000
```

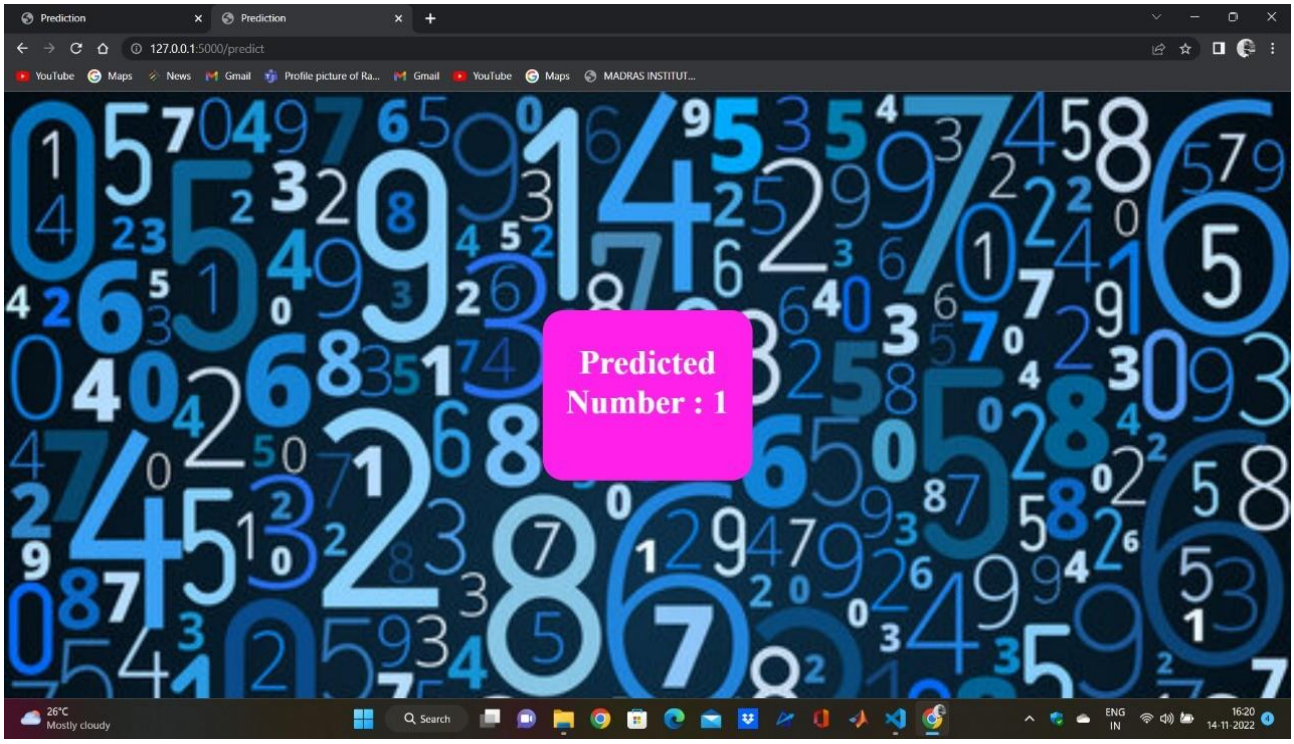
In [52]:
```
print(confusion_matrix(y_test,classes_x))
```

```
[[ 972    0    1    0    0    1    2    1    3    0]
 [   0 1120    3    1    0    0    3    2    6    0]
 [   5    2  994    8    4    0    4    5    9    1]
 [   1    0    2  985    0    6    0    6    6    4]
 [   1    1    5    0  960    0    5    1    2    7]
 [   5    1    0    5    0  867    5    2    3    4]
 [   8    3    1    1    2    7  932    1    3    0]
 [   2    4   17    5    1    0    0  988    2    9]
 [   6    1    4    8    5    3    6    6  931    4]
 [   8    7    1    5   10    1    0    7    7  963]]
```

(i) Index of Application



(ii) Prediction Page



10. ADVANTAGES AND DISADVANTAGES

Any system of its own has both advantages and disadvantages. The advantages and disadvantages of this model are as follows.

ADVANTAGES:

Using this system has many advantages. Some of them are:

- Ability to handle a lot of data.
- Less manual labour.
- Better accuracy than an average person.
- User friendly interface makes the user to navigate easily to other pages.
- The quick prediction will save the time of the users.
- It provides results in graphical representation for easy understanding.
- Can be used anywhere from any device.

DISADVANTAGES:

Using this system has many disadvantages. Some of them are:

- This is only for single digit recognition.
- The persons who have the knowledge about this only can use this.
- Cannot handle complex data.
- Low retention.
- All the data must be in digit format.
- For quicker predictions, a server with high performance is required.
- Prone to occasional errors.

11. CONCLUSION

An implementation of handwritten recognition using deep learning has been implemented. In this handwritten recognition system high accuracy is achieved. We have used the Machine Learning algorithm CNN for accuracy.

There are many real-world applications for this project, which includes reading bank cheques for amounts, processing license plate recognition data, and manually inputting data on forms like tax returns. There is a great deal of room for development that can be included into later iterations.

Accuracy can alter as it depends on the splitting of training and testing data, and this can further be improved if the number of training and testing data is provided. There is always a chance to improve accuracy if the size of data increases. Hence, utilizing these deep learning techniques, a high amount of accuracy can be obtained.

12. FUTURE SCOPE

There is a tone of room for improvement in this project. Some ways this project could be enhanced include the following:

1. Provide capabilities for saving the results of repeated digit image detection.

2. Include support to identify different digits.

3. Improve the model so it can spot numbers in a range of photos.

4. Adding support for additional languages will help users everywhere.

This project has limitless potential and is always up for improvement. This strategy's implementation will help many people and various sectors because it will lighten workloads and increase overall job effectiveness. The potential for this project is endless, and it can always be improved. Implementing this strategy will benefit numerous industries, as well as many people, as it will reduce workloads and improve overall job efficiency.

It is impossible to foresee the future of handwriting recognition, but one unlikely scenario predicts that as keyboard use rises, handwriting abilities are beginning to deteriorate. However, this trend will most likely develop gradually, and handwriting will probably continue to hold some significancefor the foreseeable future.

13. APPENDIX

SOURCE CODE

app.py

```
import numpy as np
import os
from PIL import Image
from flask import Flask, request, render_template, url_for
from werkzeug.utils import secure_filename, redirect
from gevent.pywsgi import WSGIServer
from keras.models import load_model
from keras.preprocessing import image
from flask import send_from_directory

UPLOAD_FOLDER = r"D:\IBM\run the application\project\uploads"


app = Flask(__name__)
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER

model = load_model(r"D:\IBM\run the application\project\CNN-MNIST.h5")


@app.route('/')
def index():
    return render_template('index.html')


@app.route('/predict', methods=['GET', 'POST'])
def upload():
    if request.method == "POST":
        f = request.files["image"]
        filepath = secure_filename(f.filename)
        f.save(os.path.join(app.config['UPLOAD_FOLDER'], filepath))

        upload_img = os.path.join(UPLOAD_FOLDER, filepath)
        img = Image.open(upload_img).convert("L") # convert image to monochrome
        img = img.resize((28, 28))  # resizing of input image

        im2arr = np.array(img)  # converting to image
        im2arr = im2arr.reshape(1, 28, 28, 1) # reshaping according to our requirement

        pred = model.predict(im2arr)
```

```
        num = np.argmax(pred, axis=1) # printing our Labels

        return render_template('predict.html', num=str(num[0]))


if__name__== '__main__':
    app.run(debug=True, threaded=False)
```

index.html

```
<html>

<head>
  <title>Digit Recognition WebApp</title>

  <meta name="viewport" content="width=device-width">
  <link
href="https://fonts.googleapis.com/css2?family=Prompt:wght@600&display=swap"
rel="stylesheet">
  <link href="https://fonts.googleapis.com/css2?family=Varela+Round&display=swap"
rel="stylesheet">
  <link
href="https://fonts.googleapis.com/css2?family=Source+Code+Pro:wght@500&display=
swap" rel="stylesheet">
  <link
href="https://fonts.googleapis.com/css?family=Calistoga|Josefin+Sans:400,700|Pacifico
&display=swap" rel="stylesheet">
  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-
ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">
  <link rel="stylesheet" type= "text/css" href= "{{ url_for('static',filename='css/style.css')
}}">
  <script src="https://kit.fontawesome.com/b3aed9cb07.js"
crossorigin="anonymous"></script>
  <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-
q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
crossorigin="anonymous"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"
integrity="sha384-
UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1
" crossorigin="anonymous"></script>
  <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"
integrity="sha384-
JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM"
crossorigin="anonymous"></script>
  <script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@latest"></script>
```

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

    $(document).ready(function() {
          $('#clear_button').on('click', function() {
              $('#image').val('');
              $('#frame').attr('src',"");
            });
        });

</script>

<body>

  <h1 class="welcome">IBM PROJECT
  <div id="team_id">TEAM ID : PNT2022TMID35898</div>
  </h1>
  <section id="title">
    <h4 class="heading">Handwritten Digit Recognition</h4>
    <br><br>
  </section>

  <section id="content">

        <div class="leftside">
        <form action="D:\IBM\run the application\project\template\predict.html"
method="POST" enctype="multipart/form-data">
        <label>Select a image:</label>
        <input id="image" type="file" name="image" accept="image/png, image/jpeg"
onchange="preview()"><br><br>
          <img id="frame" src="" width="100px" height="100px"/>
          <div class="buttons_div">
            <button type="submit" class="btn btn-dark"
id="predict_button">Predict</button>
            <button type="button" class="btn btn-dark" id="clear_button">  Clear
 </button>
          </div>
        </form>
        </div>
  </section>

</body>

</html>
```

predict.html

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

<style>
    body{
     background-image: url('D:\IBM\run the
application\project\static\images\numbers.jpg');
     background-repeat: no-repeat;
     background-size: cover;
    }
     #rectangle{
     width:250px
     height:100px;
     background-color: #20e4ff;
     border-radius: 25px;
     position:absolute;
     text-align:center;
     top:50%;
     left:50%;
     transform:translate(-50%,-50%);
    }

    #ans{
  text-align: center;
  font-size: 40px;
  margin: 0 auto;
  padding: 3% 5%;
  padding-top: 15%;
  color: white;
    }

</style>
<body>
    <div id="rectangle">
        <h1 id="ans">Predicted Number : {{num}}</h1>
    </div>
</body>
</html>
```

GitHub Link

https://github.com/IBM-EPBL/IBM-Project-397-1658300230

Demo Link

https://youtu.be/0YQaFG5JER0