IBM NALAIYA THIRAN

AI - A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

PROJECT REPORT

TEAM ID : PNT2022TMID30139

Submitted By

AKTCHAYAA A (611219104006)

DEEPIKA T (611219104017)

HARINI A (611219104028)

JAYASHREE K (611219104033)

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

KNOWLEDGE INSTITUTE OF TECHNOLOGY,

SALEM-637504

TABLE OF CONTENTS

CHAPTER - 1

INTRODUCTION

1.1 PROJECT OVERVIEW

Handwriting recognition is one of the compelling research works going on because every individual in this world has their own style of writing. It is the capability of the computer to identify and understand handwritten digits or characters automatically. Because of the progress in the field of science and technology, everything is being digitalized to reduce the human effort. Hence, MINIST 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 handwritten digit. This image is analyzed by the model and the detected result is returned on to UI.

1.2 PURPOSE

Handwritten digit recognition is one in all the much vital problems in pattern recognition applications. Handwriting are a few things that's able to describe the power of pc to translate the human writing to text writing. Handwriting recognition could be a methodology wherever the pc system will acknowledge digits. The handwriting recognition could be a technology that's accustomed establish bound things and additionally it's used on the devices. It can even acknowledge the digits written by hand that is natural handwriting. The aim of written system is to convert written into computer code. written digit recognition exploitation MNIST dataset could be a major project created with the assistance of Neural Network.

CHAPTER - 2

LITERATURE SURVEY

2.1 EXISTING PROBLEM

A hand-written digit recognition system is one that can identify digits that have been written by humans, instead of scanned from a document or some other source. A typical example would be an ATM machine, where the user enters their PIN number and the machine reads their input as they enter it on the keypad and then compares it to what is stored in its memory to verify that they are authorized to withdraw money from the account. Other examples include bank checks, credit card numbers, signatures on legal documents and other items where security or authenticity verification is desired.

2.2 SURVEY WORK

2.2.1 HANDWRITTEN DIGIT RECOGNITION USING CNN [Mayank Jain et al.,2021]

The basic target of this paper is to administer effective and solid procedures to acknowledgment of transcribed numerical by viewing totally different existing arrangement models. This paper is bothered the exhibition of Convolutional Neural Network (CNN). Written digit recognition is performed mistreatment the Convolutional neural network from Machine Learning. So, essentially to perform the model they add some libraries like NumPy, Pandas, TensorFlow, Keras. These area units the most structure on that my main project stands. MNIST knowledge contains regarding 70,000 pictures of written digits from 0-9. So, it's a category ten classification model. This dataset is split into a pair of components i.e. coaching and check dataset**.**

2.2.2 HANDWRITTEN CHARACTER RECOGNITION ON ANDROID FOR BASIC EDUCATION USING CONVOLUTIONAL NEURAL NETWORK [Thi Thi Zin et al, 2021]

Handwritten characters or words written on tablets were saved as input pictures. Then, they performed character segmentation. For the character recognition, the Convolutional Neural Network (CNN) was used for recognizing segmental characters. For building their own dataset, written information were collected from primary level students in developing countries. The network model was trained on a high-end machine to scale back the employment on the humanoid pill. Numerous varieties of classifiers were created so as to scale back the inaccurate classification. As per the experimental results, the planned system achieved 95.6% on the 1000 elite words and 98.7% for every character.

2.2.3 HAND WRITTEN DIGIT RECOGNITION USING MACHINE LEARNING [Rohan Sethi et al, 2020]

Hand-written character and digit recognition are one among the foremost exigent and absorbing field of pattern recognition and image process. The aim of this paper is to demonstrate and represent the work that is expounded to hand-written digit recognition. The hand-written digit recognition may be a terribly exigent task. During this recognition task, the numbers aren't accurately written or scripted as they dissent in form or size; thanks to that the feature extraction and segmentation of hand-written numerical script is arduous. The vertical and horizontal projections strategies square measure used for the aim of segmentation within the planned work. SVM is applied for recognition and classification, whereas lenticular hull formula is applied for feature extraction.

2.2.4 IMPROVED HANDWRITTEN DIGIT RECOGNITION USING CONVOLUTIONAL NEURAL NETWORKS [Savita Alhawat et al, 2020]

Proposed work is to explore the assorted style choices like variety of layers, stride size, receptive field, kernel size, cushioning and dilution for CNN-based written digit recognition. Additionally, their aim is to gauge numerous SGD improvement algorithms in raising the performance of written digit recognition. A network's recognition accuracy will increase by incorporating ensemble design. Here, the objective is to attain comparable accuracy by employing a pure CNN design while not ensemble design, as ensemble architectures introduce raised machine price and high testing quality. Thus, a CNN design is planned so as to attain accuracy even higher than that of ensemble architectures, beside reduced operational quality and price.

2.2.5 ARABIC HANDWRITING RECOGNITION SYSTEM USING CNN [Najwa Altwaijay et al, 2020]

Fewer studies are finished the Semitic. During this paper, they had a tendency to gift a replacement dataset of Arabic letters written solely by youngsters aged 7–12. The dataset contains 47,434 characters written by 591 participants. Additionally, they had a tendency to propose associate degree automatic handwriting recognition model supported convolutional neural networks (CNN). They also has a tendency to train our model on Hijja, also because the Arabic written Character Dataset (AHCD) dataset. Results show that our model's performance is promising, achieving accuracies of 97% and half of one mile on the AHCD dataset and therefore the Hijja dataset, severally.

2.2.6 A COMPARATIVE STUDY ON HANDWRITING DIGIT RECOGNITION USING NEURAL NETWORKS

[Mahmoud M. Abughosh et al, 2017]

This paper focuses on Neural Network approaches. The foremost 3 noted NN approaches are a unit deep neural network, deep belief network and convolutional neural network. During this paper, the 3 NN approaches area unit compared and evaluated in terms of the many factors like accuracy and performance. However there are a unit fascinating criteria like execution time. Random and customary dataset of written digit are used for conducting the experiments. The results show that among the 3 NN approaches, DNN is the most correct algorithm; it's 98.08% accuracy rate. On the opposite hand, every algorithmic rule has a slip rate of 1–2% as a result of the similarity in digit shapes, specially, with the digits (1,7) , (3,5) , (3,8) , (8,5) and (6,9)**.** This paper will have concluded that the filtered data could be applied to additional, non-tree-based classifiers to achieve enhanced classification efficiency and thus retain shape let understanding.

2.3 PROBLEM STATEMENT DEFINITION

Problem Statement (PS)	**I am (Customer)**	**I'm trying to**	**But**	**Because**	**Which makes me feel**
PS-1	Student	Convert Handwritten Digit to Digital form	Unable to convert huge amount of data	Time consumption is high	Frustrated
PS-2	Data entry operator	Uploading customer information	Unable to do repetitive work	Manual work is high	Disappointed
PS-3	Bank Employee	Process the cheques	Manual error is high	It contains delicate numbers	Anxious
PS-4	Post Office Employee	Sort the mail	Can't understand the digits	It includes different styles	Upset

CHAPTER-3

IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP & CANVAS

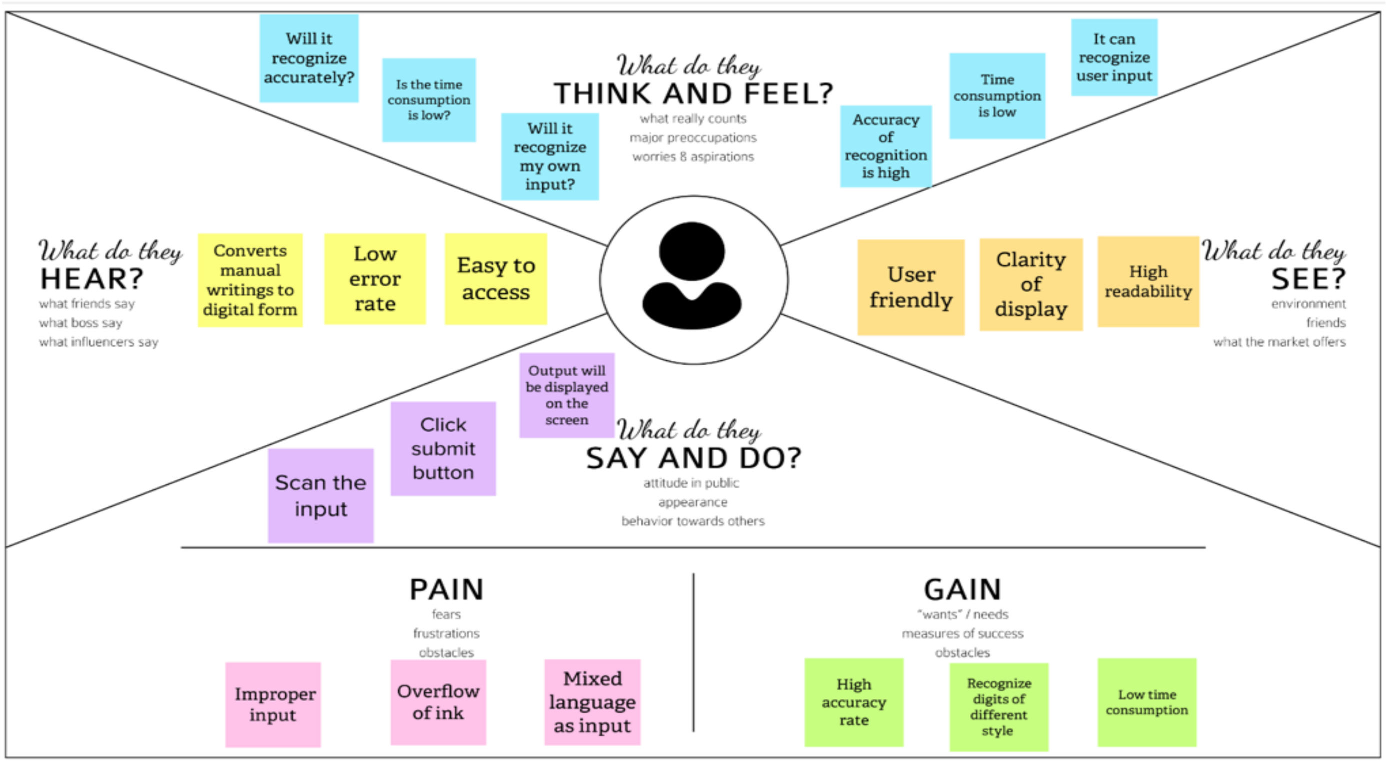


Fig 3.1 Empathy Map

3.2 IDEATION & BRAINSTORMING

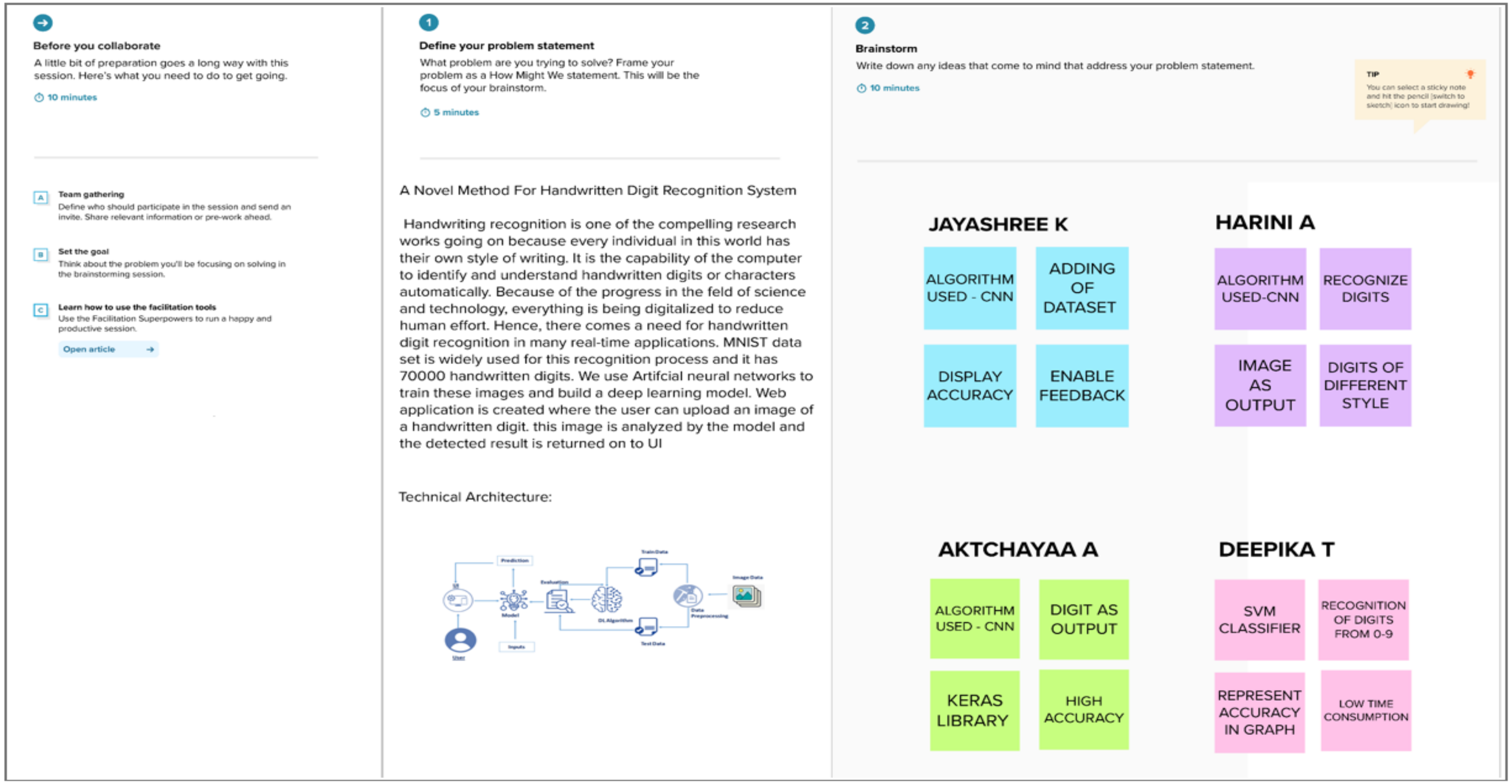


Fig 3.2.1 Brainstorming and Idea Prioritization

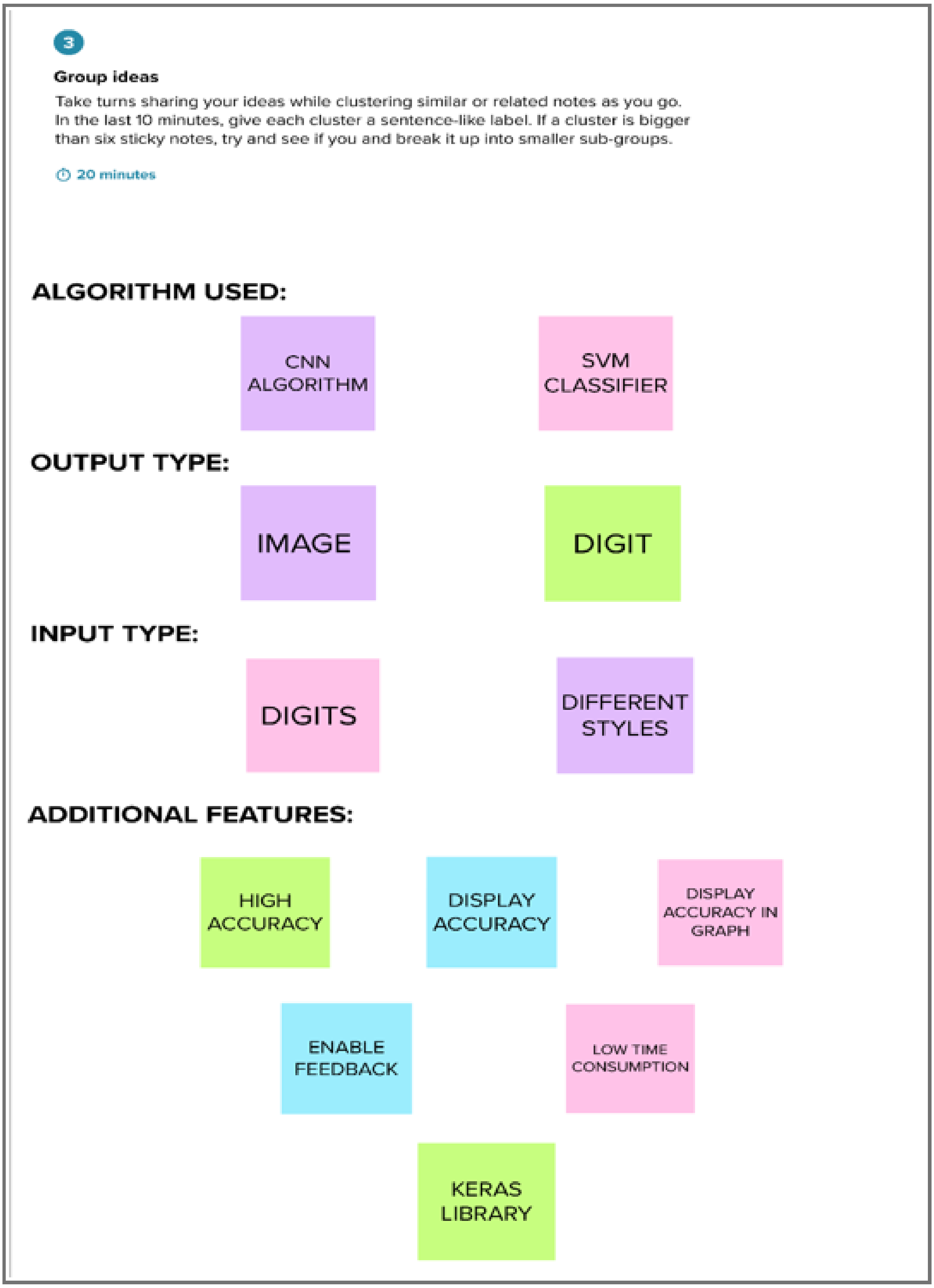


Fig 3.2.2 Brainstorming and Idea Prioritization

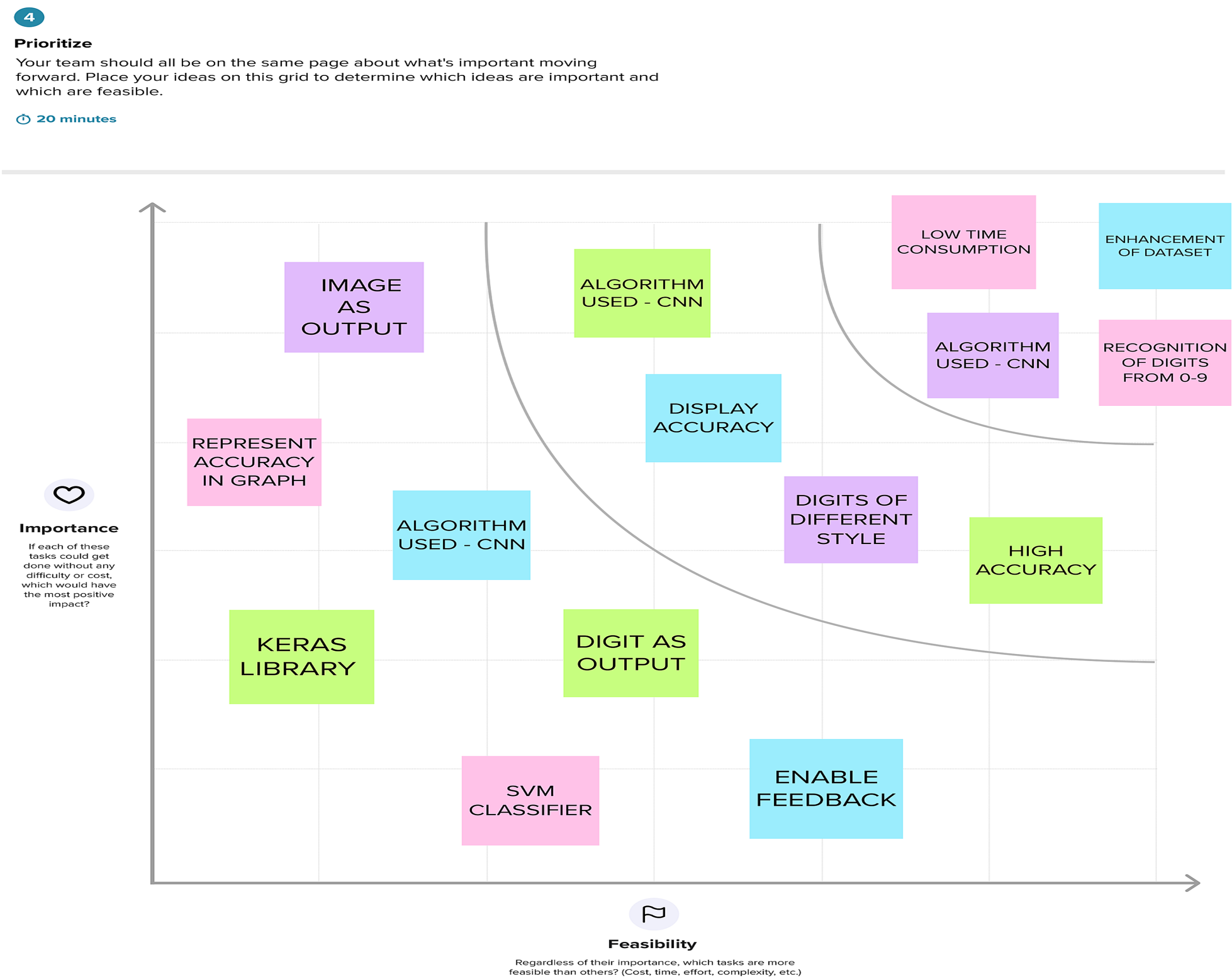


Fig 3.2.3 Brainstorming and Idea Prioritization

3.3 PROPOSED SOLUTION

Hand written digit recognition is a technology that has been around for a while. It is the process of converting hand-written digits into machine-readable data. Hand written digit recognition software is used in many different applications such as document scanning, handwriting input, and signature verification. Handwritten digit recognition is a difficult task for humans and for computer systems. The accuracy of human recognition is about 98%. This means that the error rate is about 2%. However, the error rate in machine-based recognition is much higher than that of humans.

The neural networks are a type of machine learning algorithm which has been proven to be able to solve this problem. Developers have been working on improving the performance of these neural networks by optimizing algorithms and testing new architectures. The proposed solution is an AI-based system that can recognize handwritten digits using a convolutional neural network (CNN) and a graphical user interface (GUI).

A CNN is a type of neural network that has been used for handwriting recognition. In the past, the accuracy of handwritten digit recognition systems was not high enough to be used in banking or other high-security applications. However, recent improvements in machine learning algorithms and hardware allow handwriting recognition systems to achieve higher accuracies than before.

The system is composed of a GUI, CNN and the handwritten digit recognition module. The GUI is used for inputting data, CNN for processing and classifying the inputted data and finally the handwritten digit recognition module for recognizing the inputted data.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	It is easy for the human to perform a task accurately by practicing it repeatedly and memorizing it for the next time. Human brain can processand analyse imageseasily. Also, recognize the different element present in the images. In this competition, the goal is to correctly identify digits from a dataset of tens of thousands of handwritten images and experiment with different algorithms to learn what works well and how techniques compare.
2.	Idea / Solution description	Our model converts handwritten digits into digital form using CNN algorithm as the algorithm has low time consumption compared to other neural network algorithms and it also gives high accuracy. We also produce accuracy level along with output which is represented in graph.
3.	Novelty / Uniqueness	The primary motto of our model is to convert handwritten digits into digital form. The input can be given in two forms either we can upload or draw the digits.
4.	Social Impact / Customer Satisfaction	When handwritten digits are converted into digital form it consumes more time. So we overcome that by our model.
5.	Business Model (Revenue Model)	Our system can be implemented in banking, postoffice, data entry etc..,
6.	Scalability of the Solution	It also helps many individuals to solve their problem.

3.4 PROBLEM SOLUTION FIT

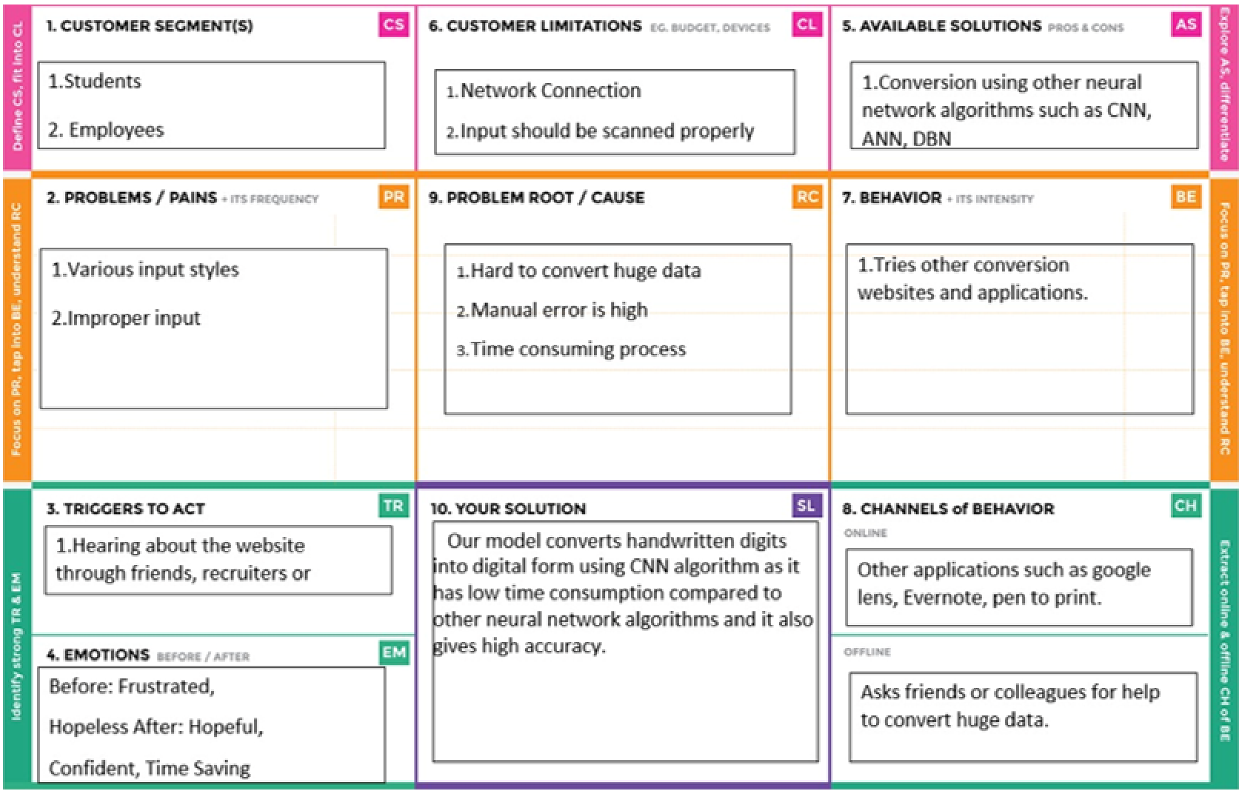


Fig 3.4 Problem Solution Fit

CHAPTER - 4

REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Input	Allow user to enter the input in GUI and recognize the input.
FR-2	Model	A file that has been trained to recognize certain types of patterns.
FR-3	Prediction	Train and Test the model and predict the user input.
FR-4	Evaluation	Checking the model whether the prediction is correct.

4.2 NON-FUNCTIONAL REQUIREMENTS

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

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	Can predict digits correctly. Our model can be used in postal mail sorting, bank check processing, data entry, etc.
NFR-2	Security	User data is secured. It ensures security in a way that the images uploaded for recognition will not be stored once the recognition is done.
NFR-3	Reliability	Can process confidential information. Recognize digits without interruption.

NFR-4	Performance	Improvement in fast prediction. We use CNN algorithm for accurate prediction.
NFR-5	Availability	Available as software for both common and professional use.
NFR-6	Scalability	It also helps many individuals to solve their problem with low time consumption and high accuracy.

CHAPTER - 5

PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS

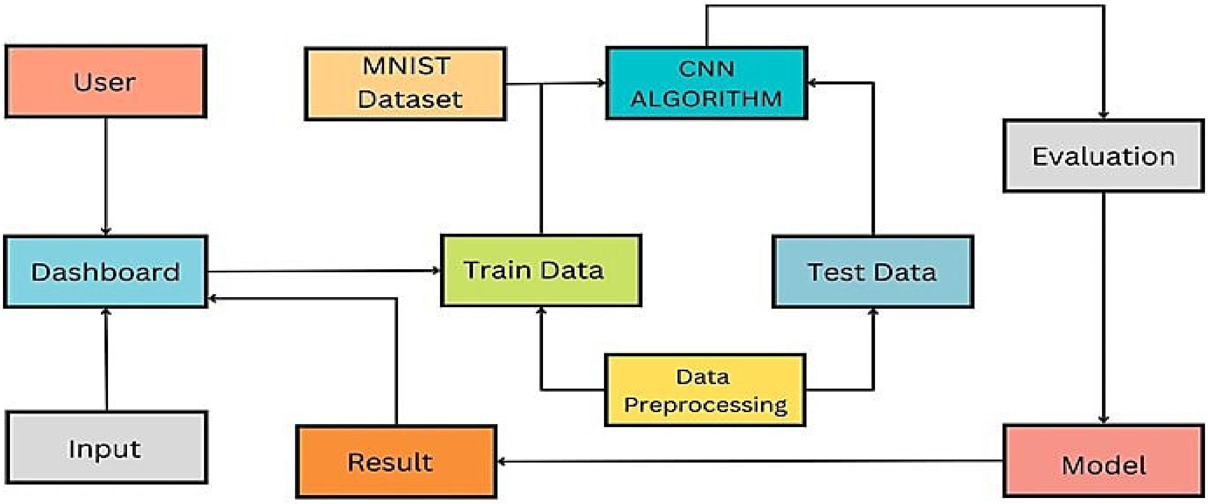


Fig 5.1.1 Data Flow Diagram

5.2 SOLUTION & TECHNICAL ARCHITECTURE

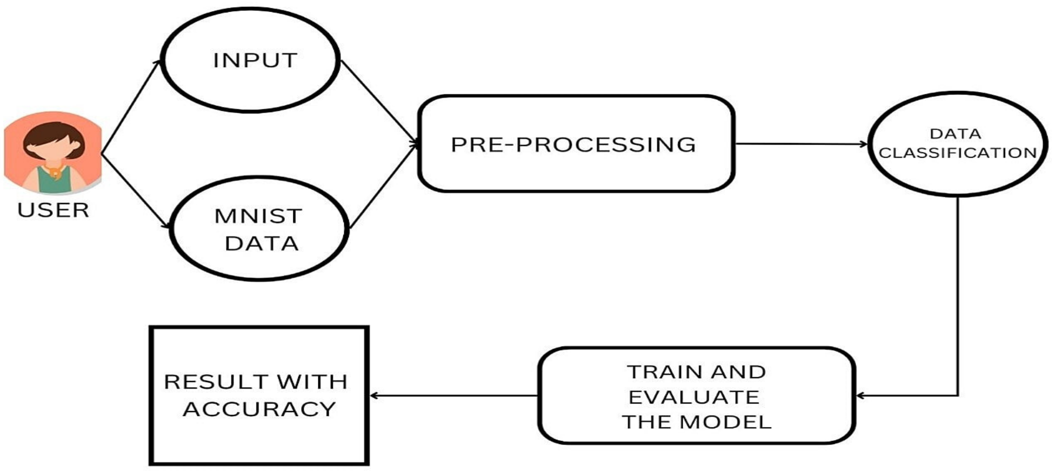


Fig 5.2.1 Technical Architecture

5.3 USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Web user)	Home	USN-1	In the Home Page, I can view the guidelines of how to use the website	I can view the guidelines	Low	Sprint 1
	Dashboard	USN-2	As a user, I can see Home Page & Prediction Page	I can access my account / dashboard	Low	Sprint 2
	Apply	USN-3	In Prediction Page, I can upload my digits from 0-9 for prediction	I can upload my input by predict button	Medium	Sprint 3
		USN-4	As a user, I can get an accuracy rate with the prediction	I have different forms of output	High	Sprint 4
	Recognition	USN-5	As a user, I can see that the GUI recognize the input	I can perform handwritten digit predictions	High	Sprint 1
	Prediction	USN-6	As a user, I can get accuracy rate by pressing the predict button	I can draw the number and click on predict button	High	Sprint 1

CHAPTER - 6

PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANNING & ESTIMATION

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Dashboard	USN-1	As a user,I can see the dashboard that contains Home, Prediction & Guidelines bars in the dashboard.	2	Low	HARINI A
Sprint-1	Home	USN-2	In the Home Page,I can view the guidelines of how to use the website.	2	Low	JAYASHREE K
Sprint-1	User Input	USN-3	In Prediction Page, I can upload the digit.	3	Medium	AKTCHAYAA A

Sprint-2	Apply	USN-4	The inputs are pre-processed and the model is created.	3	Medium	DEEPIKA T
Sprint-2		USN-5	The uploaded digits will be trained with the model.	7	High	JAYASHREE K
Sprint-2		USN-6	The inputs are evaluated with the MNIST dataset.	5	Medium	HARINI A
Sprint-3	Recognition	USN-7	The digits are tested with the algorithm.	7	High	DEEPIKA T
Sprint-4	Prediction	USN-8	As a user, I can get an accuracy rate.	7	High	JAYASHREE K
Sprint-1	GUI	USN-9	As a user, I can upload the input to predict the digits.	5	Medium	DEEPIKA T
Sprint-3	API	USN-10	Fetch the data from dataset to the model.	3	Medium	HARINI A

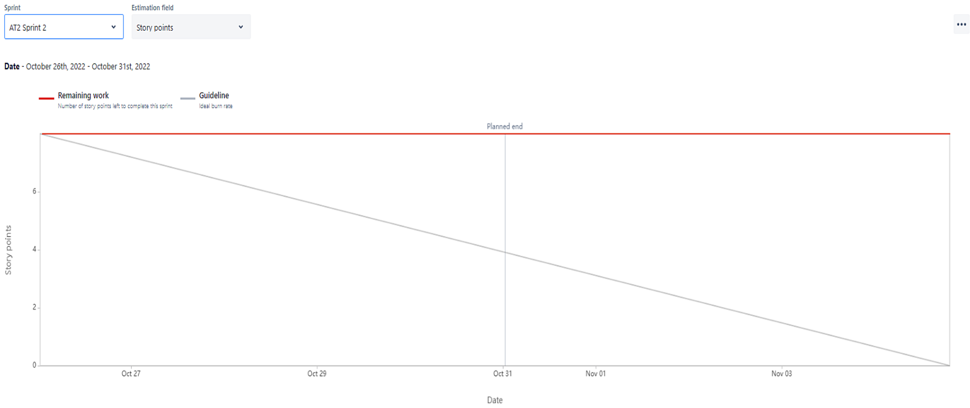
6.2 SPRINT DELIVERY SCHEDULE

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	17	6 Days	24 Oct 2022	29 Oct 2022	17	30 Oct 2022
Sprint-2	18	6 Days	31 Oct 2022	05 Nov 2022	18	06 Nov 2022
Sprint-3	10	6 Days	07 Nov 2022	12 Nov 2022	10	13 Nov 2022
Sprint-4	14	6 Days	14 Nov 2022	19 Nov 2022	14	19 Nov 2022

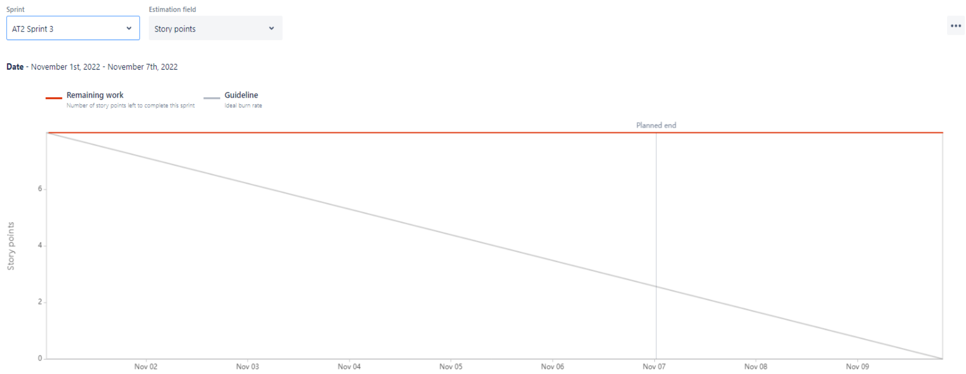
6.3 REPORT FROM JIRA

SPRINT 1

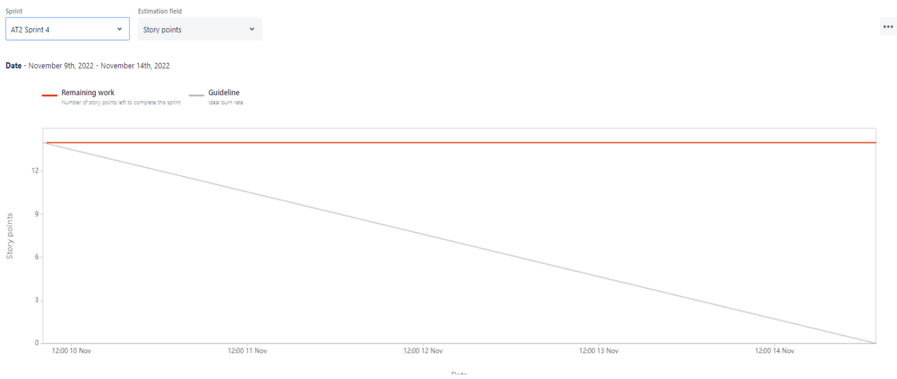
SPRINT 2



SPRINT 3



SPRINT 4



CHAPTER - 7

CODING & SOLUTIONING

7.1 FEATURE 1

Handwritten digit recognition is a technology that can be used to recognize handwritten digits. The system can work with a variety of inputs, as long as the handwriting is clear enough for the system to recognize it. It is used in various fields like banking, education and healthcare. It is also used as a security measure. For example, it can be used to prevent identity theft and stop fraudulent transactions. Features are based on shape analysis of the digit image and extract slant or slope information. They are effective in obtaining good recognition accuracies.

```
@app.route('/Predict',methods=['POST'])
def upload_image_file():
    if request.method=='POST':
      if request.files["file"]:
        img = Image.open(request.files['file'].stream).convert("L")
        img = img.resize((28,28))
        im2arr = np.array(img)
        im2arr = im2arr.reshape(1,28,28,1)
        y_pred = np.argmax(model.predict(im2arr))
        if(y_pred >= 0 and y_pred < 10):
         return render_template("Upload.html", showcase = str(y_pred))
      else :
         return render_template("Upload.html")
```

Fig 7.2.1 Coding & Solutioning

7.2 FEATURE 2

The training process is time consuming and requires a lot of manual work. In this research, we propose a novel approach for digit recognition that is based on deep convolutional neural networks (CNNs) and achieves high accuracy by using an end-to-end training framework. Handwritten digit recognition is an emerging technology that is being used to recognize the handwritten numbers.

The main features of a handwritten digit recognition system are as follows:

- It can identify numbers from 0 to 9.
- It is fast and accurate.

```
@app.route('/')
def upload_file():
    return render_template('Home.html')
@app.route('/Upload')
def upload_file1():
    return render_template('Upload.html')
@app.route('/Draw')
def upload_file2():
    return render_template('Draw.html')
@app.route('/Guidelines')
def upload_file3():
    return render_template('Guidelines.html')
```

Fig 7.2.2 Coding & Solutioning

CHAPTER - 8

TESTING

8.1 TEST CASES

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status
HomePage_TC_OO1	UI	Home Page	Verify the user is able to see and access the Home page clearly		1.Enter URL and press enter	http://192.168.202.145:8000	Home page should display	Working as expected	Pass
Dropdown_TC_OO2	UI	Home Page	Verify the UI elements in HomePage popup		1.Enter URL and click go 2.Click on Prediction dropdown button 3.Drop down should popup	http://192.168.202.145:8000	Prediction drop down should display	Working as expected	Pass
PredictionPage_TC_OO3	Functional	Prediction page	Verify user is able to choose image from local storage		1.Enter URL(http://127.0.0.1.8000) and click go 2.Click on Prediction dropdown button 3.Choose Upload bar/Draw bar	http://192.168.202.145:8000	User should navigate to prediction	Working as expected	Pass
UploadPage_TC_OO4	Functional	Upload page	Verify user is able to insert the image		1.Enter URL(http://127.0.0.1.8000) and click go 2.Click on prediction dropdown button 3. Choose Upload bar 4. Choose the image from local storage 5.Click Predict	http://192.168.202.145:8000	Application should show the Predicted Result	Working as expected	Pass
DrawPage_TC_OO4	Functional	Draw page	Verify user is able to draw the digits in the black canvas		1.Enter URL (http://127.0.0.1.8000) and click go 2.Click on prediction dropdown button 3. Choose Draw bar 4. Draw the digit in the black canvas	http://192.168.202.145:8000	Application should show the Predicted Result	Working as expected	Pass
Guidelines_TC_OO5	UI	Guidelines	Verify user is able to see the Guidelines		1.Enter URL (http://127.0.0.1.8000) and click go 2.Click on Guidelines bar 3. User can see the Guidelines for Upload and Draw	http://192.168.202.145:8000	Application should show the Guidelines for Upload and Draw	Working as expected	Pass

8.2 USER ACCEPTANCE TESTING

UAT Execution & Report Submission

8.2.1 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).

8.2.2 Defect Analysis

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	7	3	1	2	13
Duplicate	1	0	2	0	3
External	3	2	0	0	5
Fixed	8	3	2	10	23
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	3	2	1	6
Totals	19	11	9	14	53

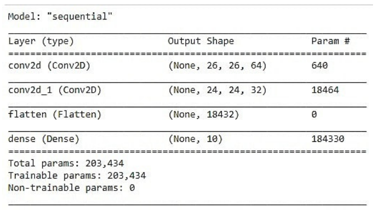
8.2.3 Test Case Analysis

Section	Total Cases	Not Tested	Fail	Pass
PrintEngine	5	0	0	5
ClientApplication	45	0	0	45
Security	2	0	0	2
Outsource Shipping	2	0	0	2
Exception Reporting	8	0	0	8
FinalReport Output	5	0	0	5
VersionControl	2	0	0	2

CHAPTER - 9

RESULT

9.1 PERFORMANCE METRICS

<table>
<tr><th>S.No.</th><th>Parameter</th><th>Values</th><th>Screenshot</th></tr>
<tr><td>1.</td><td>Model Summary</td><td>CNN, Neural network, Machine learning, Keras.</td><td>
Model: "sequential"
<table><tr><th>Layer (type)</th><th>Output Shape</th><th>Param #</th></tr><tr><td>conv2d (Conv2D)</td><td>(None, 26, 26, 64)</td><td>640</td></tr><tr><td>conv2d_1 (Conv2D)</td><td>(None, 24, 24, 32)</td><td>18464</td></tr><tr><td>flatten (Flatten)</td><td>(None, 18432)</td><td>0</td></tr><tr><td>dense (Dense)</td><td>(None, 10)</td><td>184330</td></tr></table>Total params: 203,434
Trainable params: 203,434
Non-trainable params: 0</td></tr>
<tr><td>2.</td><td>Accuracy</td><td>Training Accuracy - 97.8
Validation Accuracy - 98.9</td><td></td></tr>
</table>

CHAPTER - 10

ADVANTAGES & DISADVANTAGES

ADVANTAGES

It is not necessary to capture the entire image for successful recognition, so it can be used when only a small part of an image is available. It does not need a special environment or background, so it can be used in various situations. The accuracy rate for handwritten digit recognition is high because it does not depend on external factors such as lighting or background. It does not require any special hardware because it can be implemented with low cost devices such as cameras and computers.

DISADVANTAGES

The main disadvantage of our model is that the accuracy rate is low when the input digits are inappropriate. They can't recognize the handwriting correctly, which can lead to mistakes in the conversion process. The disadvantages of this system are that it can be difficult for some people to write legibly and the system may not be able to recognize some numbers, such as handwritten fractions or Roman numerals.

CHAPTER - 11

CONCLUSION

The goal of this project is to create a system that can recognize handwritten digits. The system will be implemented in Python and use the Matplotlib, NumPy, Pillow library for image processing. It will be done by implementing a system that can recognize handwritten digits. It is an important part of the digitization process. The system has many applications, including in banking and security, where handwritten signatures are required to authenticate transactions. The accuracy of this system is dependent on two things: the quality of the handwriting and how well it can be analyzed. This system has been used in many fields such as banking, law enforcement and education.

CHAPTER - 12

FUTURE SCOPE

In Future, Handwritten digit recognition systems are becoming more and more popular in our lives. They are used for a variety of purposes, from logging into smartphones to authenticating credit card purchases. This type of system is usually composed of an input device, a processing unit and an output device. The input device captures the handwritten digit, the processing unit converts it into a digital form and the output device displays the recognized number.

CHAPTER - 13

APPENDIX

13.1 SOURCE CODE

```
#Importing required libraries
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 the data
(x_train, y_train), (x_test, y_test) = mnist.load_data()
print(x_train.shape)
print(x_test.shape)

#Analyzing the data
x_train[0]
y_train[0]
import matplotlib.pyplot as plt
plt.imshow(x_train[0])

#Reshaping the data
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
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)

#Add CNN Layers
model = Sequential()
model.add(Conv2D(64, (3,3), input_shape=(28,28,1), activation='relu'))
model.add(Conv2D(32, (3,3), activation='relu'))
model.add(Flatten())
model.add(Dense(number_of_classes, activation='softmax'))

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

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

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

#Test the model
prediction=model.predict(x_test[:4])
print(prediction)

#Observing the Metrics
import numpy as np
print(np.argmax(prediction,axis=1))
print(y_test[:4])
```

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

Web (GUI)

```
from flask import Flask, render_template, request
from PIL import Image
import numpy as np
from tensorflow.keras.models import load_model
import tensorflow as tf

app = Flask(__name__, template_folder="templates/templates")

model = load_model('models/mnistCNN.h5')
print("Loaded model from disk")

@app.route('/')
defupload_file():
    return render_template('Home.html')
@app.route('/Upload')
defupload_file1():
    return render_template('Upload.html')
@app.route('/Draw')
defupload_file2():
    return render_template('Draw.html')
@app.route('/Guidelines')
defupload_file3():
    return render_template('Guidelines.html')

@app.route('/Predict',methods=['POST'])
defupload_image_file():
   if request.method=='POST':
     if request.files["file"]:
       img = Image.open(request.files['file'].stream).convert("L")
```

```
        img = img.resize((28,28))
        im2arr = np.array(img)
        im2arr = im2arr.reshape(1,28,28,1)
        y_pred = np.argmax(model.predict(im2arr))
        if(y_pred >= 0 and y_pred < 10):
         return render_template("Upload.html", showcase = str(y_pred))
      else :
         return render_template("Upload.html")

if__name__ == '__main__':
    app.run(host = '0.0.0.0', port = 8000, debug = True)
```

13.2 GITHUB & PROJECT DEMO LINK

Github - https://github.com/IBM-EPBL/IBM-Project-7579-1658892763

Project Demo - https://youtu.be/gz7kGh5FuZU

CHAPTER - 14

REFERENCES

[1] Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR ’05), San Diego, CA, USA, 20–26 June 2005; Volume 1, pp. 886–893.

[2] Lowe, D.G. Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. 2004, 60, 2. [CrossRef]

[3] Xiao, J.; Xuehong, Z.; Chuangxia, H.; Xiaoguang, Y.; Fenghua, W.; Zhong, M. A new approach for stock price analysis and prediction based on SSA and SVM. Int. J. Inf. Technol. Decis. Making 2019, 18, 287–310. [CrossRef]

[4] Wang, D.; Lihong, H.; Longkun, T. Dissipativity and synchronization of generalized BAM neural networks with multivariate discontinuous activations. IEEE Trans. Neural Netw. Learn. Syst. 2017, 29, 3815–3827. [PubMed]

[5] Kuang, F.; Siyang, Z.; Zhong, J.; Weihong, X. A novel SVM by combining kernel principal component analysis and improved chaotic particle swarm optimization for intrusion detection. Soft Comput. 2015, 19, 1187–1199. [CrossRef]

[6] Choudhary, A.; Ahlawat, S.; Rishi, R. A binarization feature extraction approach to OCR: MLP vs. RBF. In Proceedings of the International Conference on Distributed Computing and Technology ICDCIT, Bhubaneswar, India, 6–9 February 2014; Springer: Cham, Switzerland, 2014; pp. 341–346.

[7] Fukushima, K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biol. Cybern. 1980, 36, 193–202. [CrossRef]