PROJECT REPORT

A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION

Aarthi M	**810019104001**
Aarthi V	**810019104002**
Abinayasri T	**810019104003**
Jayashree R	**810019104025**

Submitted by
PNT2022TMID32288

TABLE OF CONTENTS

CHAPTER 1

1. INTRODUCTION

1.1 PROJECT OVERVIEW

- To know fundamental concepts and techniques of the Artificial Neural Network and Convolution Neural Networks

- To gain a broad understanding of image data

- To work with Sequential type of modeling

- To work with Keras capabilities

- To work with image processing techniques

- To know how to build a web application using the Flask framework

1.2 PURPOSE

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.

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 fifilled up by hand (tax forms) and so on.

CHAPTER 2

2.LITERATURE SURVEY

2.1. EXISTING PROBLEM

The advance of handwriting processing results from a combination of various elements, for example: improvements in there cognition rates, the use of complex systems to integrate various kinds of information, and new technologies such as high quality high speed scanners and cheaper and more powerful CPUs. Some handwriting recognition system allows us to input our handwriting into the system.

Handwritten digit recognition is the ability of a computer system to recognize the handwritten inputs like digits, characters etc. from a wide variety of sources like emails, papers, images, letters etc. This has been a topic of research for decades. Some of the research areas include signature verification, bank check processing, postal address interpretation from envelopes etc.

Handwriting recognition system is the most basic and an important step towards this huge and interesting area of Computer Vision.

2.2. REFERENCES

NO: 1

TITLE: Neural Network Based Handwritten Digit Recognition for Managing Examination Score in Paper Based Test

AUTHORS: Ankit Sharma1 , Yogiraj Barole , Kaustubh Kerhalkar , Dr. Prabhu K.R.

PUBLISHING YEAR: 2016

CONTENT:

Recognition of handwritten character is a difficult task in the field of image processing, artificial intelligence since the handwriting varies from person to person. In proposed paper, we are training the neural network to recognize the off-line strategies for the isolated handwritten character (0 to 9). This work improves the character recognition and pre processing of the Character is done by image rendering, character extraction and training and testing steps. The proposed method is based on the use of linear regression algorithm to classify the characters and is used to train the given dataset. After training a network performance curve is generated along with the individual required characters. In given system, numerical character is represented by binary numbers that are used as input then they are fed to an ANN. Neural network followed by the linear regression Algorithm which compromises Training.

NO: 2

TITLE: Handwritten digit recognition by combined classifiers

AUTHORS: M. Breukelen; Robert P. W. Duin; David M. J. Tax; J. E. den Hartog**.**

PUBLISHING YEAR: 1998

CONTENT:

In practical pattern recognition problems one often tries a number of classifiers and a number of feature sets in order to find the best combination. As soon as this combination is found the other classifiers and features are no longer used. Methods for combining classifiers to reduce the number of classification errors are described in recent literature. In this paper the usefulness of combining classifiers was tested on a real data set consisting of several sets of features of handwritten digits. The questions we would like to answer are: "When does combining classifiers result in a reduction of classification errors and why?" and "If we have a large set of features, how do we divide this set into subsets in order to get the best results?" Section 2 this paper describes how classifiers can be combined, Section 3 how our classifiers estimate posterior probabilities and Section 4 describes our data. In Section 5 we describe the experiments we did and in Section 6 our conclusions.

NO: 3

TITLE: A hybrid recognition system for off-line handwritten characters

AUTHORS: Gauri Katiyar and Shabana Mehfuz

PUBLISHING YEAR: **2016**

CONTENT:

Computer based pattern recognition is a process that involves several sub-processes, including pre-processing, feature extraction, feature selection, and classification. Feature extraction is the estimation of certain attributes of the target patterns. Selection of the right set of features is the most crucial and complex part of building a pattern recognition system. In this work we have combined multiple features extracted using seven different approaches. The novelty of this approach is to achieve better accuracy and reduced computational time for recognition of handwritten characters using Genetic Algorithm which optimizes the number of features along with a simple and adaptive Multi Layer Perceptron classifier. Experiments have been performed using standard database of CEDAR (Centre of Excellence for Document Analysis and Recognition) for English alphabet.

NO: 4

TITLE: Analytical Review of Preprocessing Techniques for Offline Handwritten Character Recognition

AUTHORS: Gaurav Kumar and Pradeep Kumar Bhatia

PUBLISHING YEAR: 2013

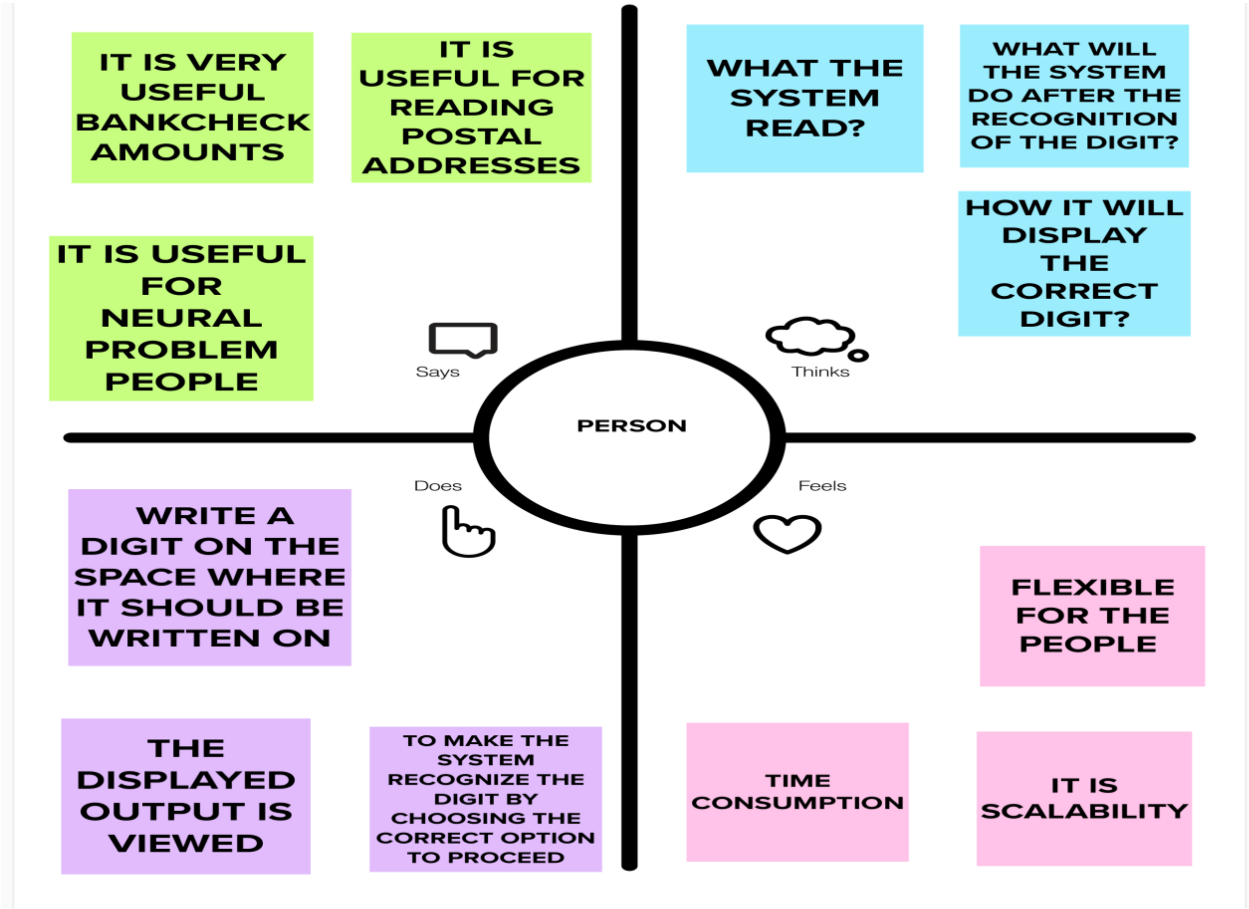
CONTENT:

Preprocessing techniques are the first step in a character recognition system. This paper deals with the various preprocessing techniques involved in character recognition system with different kind of images ranges from simple handwritten form based documents and documents containing colored and complex background and varied intensities. Here, we are going to discuss all important preprocessing techniques like skew detection and correction, image enhancement techniques of contrast stretching, binarization, noise removal techniques, normalization and segmentation, morphological processing techniques

CHAPTER 3

3.IDEATION AND PROPOSED SOLUTION

3.1. EMPATHY MAP



IT IS VERY USEFUL BANKCHECK AMOUNTS

IT IS USEFUL FOR READING POSTAL ADDRESSES

IT IS USEFUL FOR NEURAL PROBLEM PEOPLE

WHAT THE SYSTEM READ?

WHAT WILL THE SYSTEM DO AFTER THE RECOGNITION OF THE DIGIT?

HOW IT WILL DISPLAY THE CORRECT DIGIT?

WRITE A DIGIT ON THE SPACE WHERE IT SHOULD BE WRITTEN ON

THE DISPLAYED OUTPUT IS VIEWED

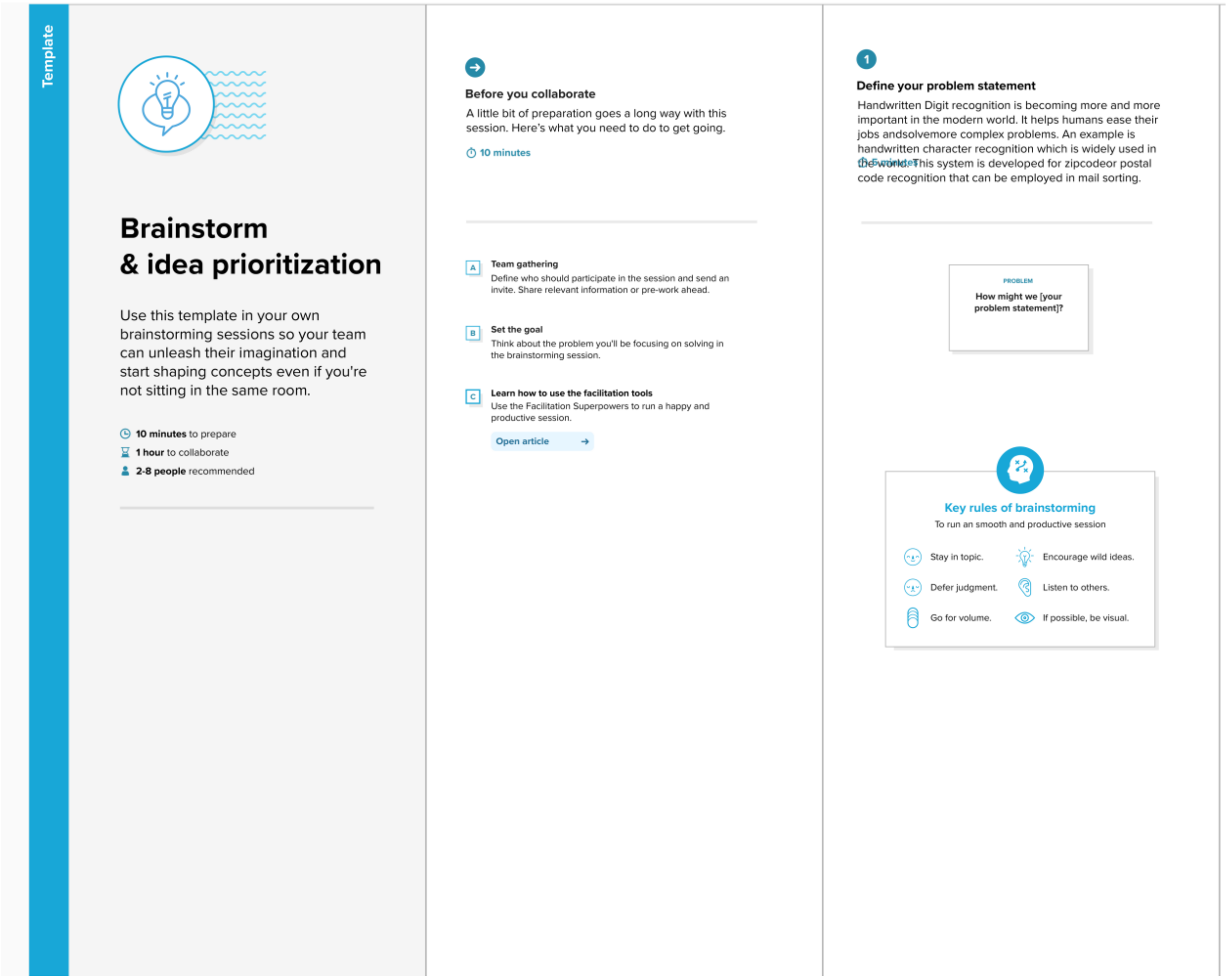
TO MAKE THE SYSTEM RECOGNIZE THE DIGIT BY CHOOSING THE CORRECT OPTION TO PROCEED

FLEXIBLE FOR THE PEOPLE

TIME CONSUMPTION

IT IS SCALABILITY

3.2. IDEATION & BRAINSTORMING



Template

Brainstorm & idea prioritization

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

10 minutes to prepare

1 hour to collaborate

2-8 people recommended

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

A **Team gathering**
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B **Set the goal**
Think about the problem you'll be focusing on solving in the brainstorming session.

C **Learn how to use the facilitation tools**
Use the Facilitation Superpowers to run a happy and productive session.

Open article →

1

Define your problem statement

Handwritten Digit recognition is becoming more and more important in the modern world. It helps humans ease their jobs andsolvemore complex problems. An example is handwritten character recognition which is widely used in the world. This system is developed for zipcodeor postal code recognition that can be employed in mail sorting.

5 minutes

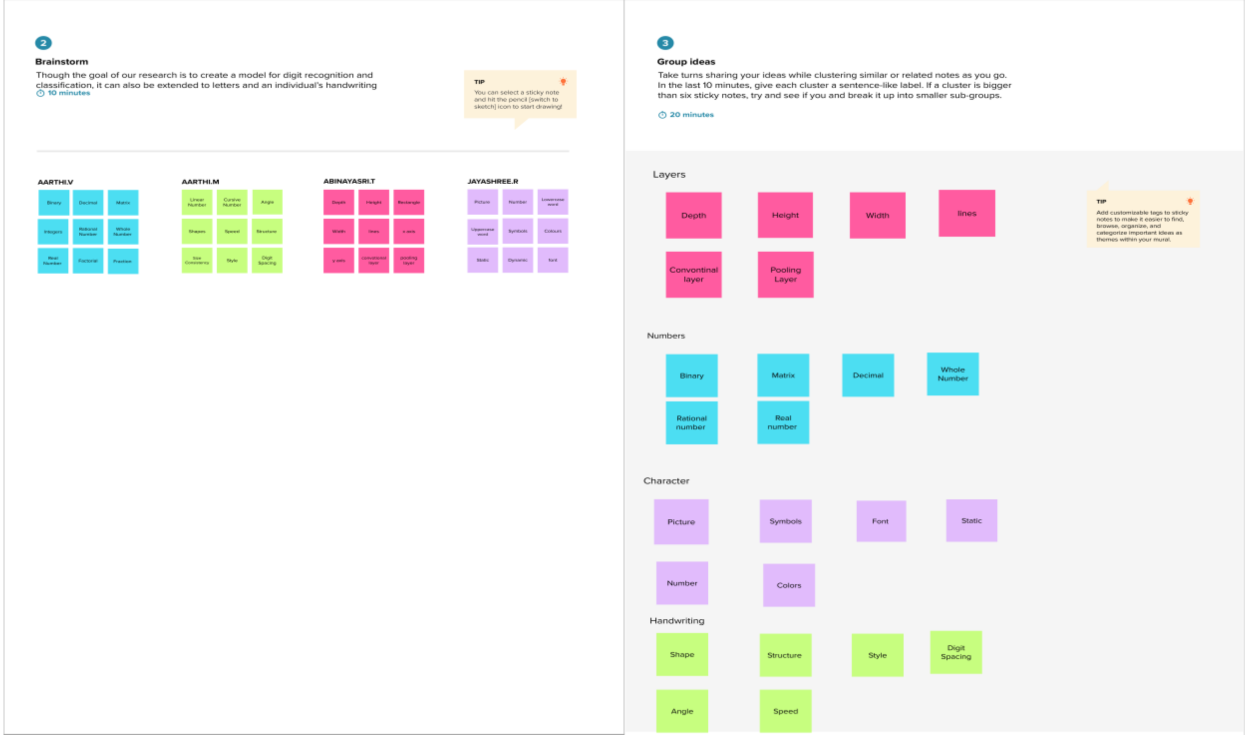
PROBLEM

How might we [your problem statement]?

Key rules of brainstorming

To run an smooth and productive session

Stay in topic.	Encourage wild ideas.
Defer judgment.	Listen to others.
Go for volume.	If possible, be visual.



2

Brainstorm

Though the goal of our research is to create a model for digit recognition and classification, it can also be extended to letters and an individual's handwriting

10 minutes

TIP

You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

AARTHI.V

- Binary
- Decimal
- Matrix
- Integers
- Rational Number
- Whole Number
- Real Number
- Factorial
- Fraction

AARTHI.M

- Linear Number
- Cursive Number
- Angle
- Shapes
- Speed
- Structure
- Size Consistency
- Style
- Digit Spacing

ABINAYASRI.T

- Depth
- Height
- Rectangle
- Width
- lines
- x axis
- y axis
- convotional layer
- pooling layer

JAYASHREE.R

- Picture
- Number
- Lowercase word
- Uppercase word
- Symbols
- Colours
- Static
- Dynamic
- font

3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10 minutes, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

TIP

Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

Layers

- Depth
- Height
- Width
- lines
- Convontinal layer
- Pooling Layer

Numbers

- Binary
- Matrix
- Decimal
- Whole Number
- Rational number
- Real number

Character

- Picture
- Symbols
- Font
- Static
- Number
- Colors

Handwriting

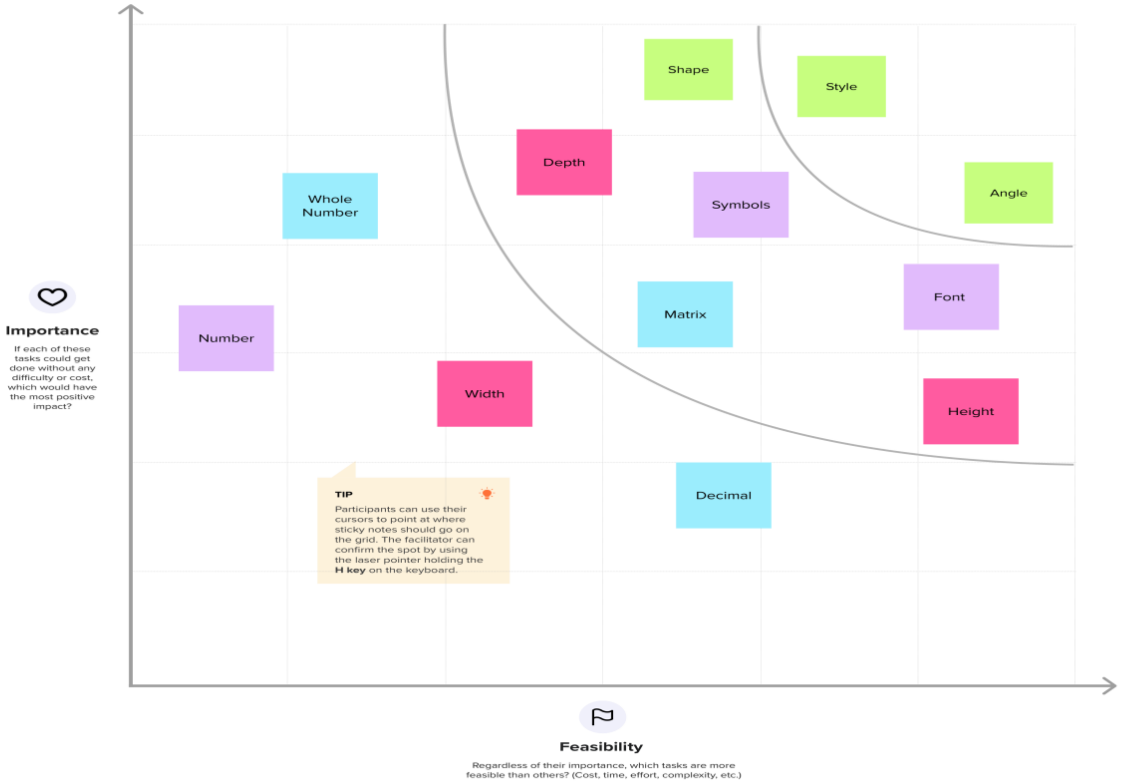
- Shape
- Structure
- Style
- Digit Spacing
- Angle
- Speed

4

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes



Importance

If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?

TIP

Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the **H key** on the keyboard.

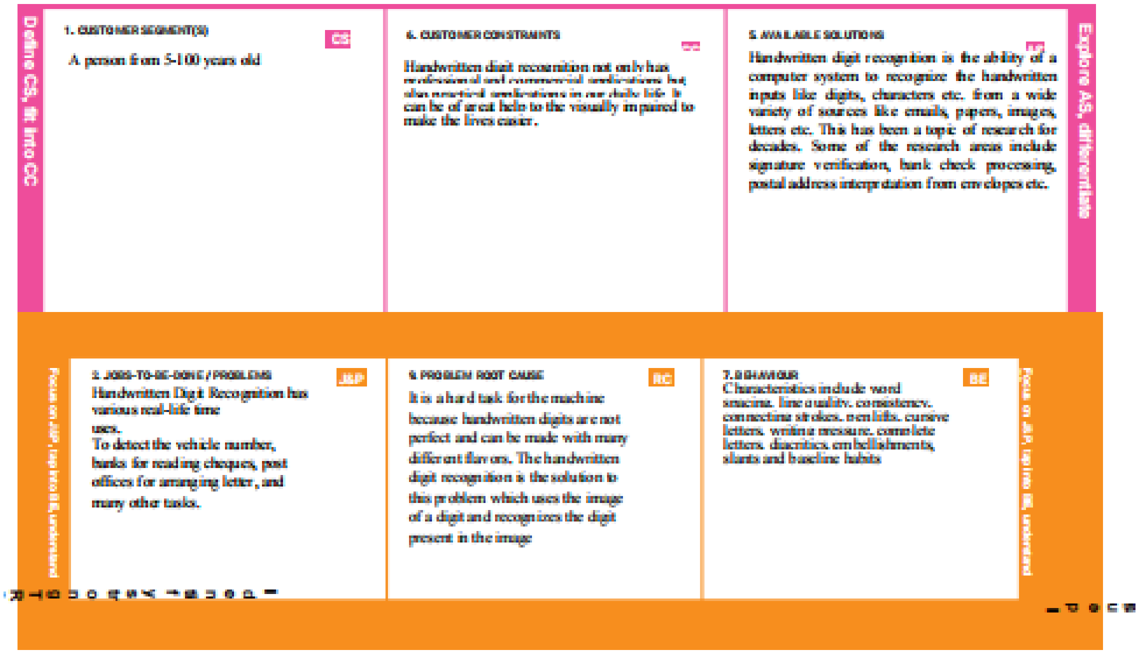
Feasibility

Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

3.3. PROPOSED SOLUTION

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Handwriting recognition is one of the compelling research works going on because every individual in this world has their own style of writing. It is the capability of the computer to identify and understand handwritten digits or characters automatically. Because of the progress in the field of science and technology, everything is being digitalized to reduce human effort.
2.	Idea / Solution description	We use Artificial neural networks to train these images and build a deep learning model. Web application is created where the user can upload an image of a handwritten digit. this image is analyzed by the model and the detected result is returned on to UI and using the tab we can also write the digit and can easily be recognized on the system.
3.	Novelty / Uniqueness	The user interacts with the UI (User Interface) to upload the image as input .The uploaded image is analyzed by the model which is integrated Once the model analyses the uploaded image, the prediction is showcased on the UI. The goal of our work is to create a model that will be able to recognize and classify the handwritten digits from images by using concepts of Convolution Neural Network.
4.	Social Impact / Customer Satisfaction	Handwritten Digit Recognition has various real-life time uses. It is used in the detection of vehicle number, banks for reading cheques, post offices for arranging letter, and many other tasks.
5.	Business Model (Revenue Model)	It is cost-efficiency but also it provides best results.
6.	Scalability of the Solution	This model can be expanded to include more attributes for more accurate detection .Training the model with even more attributes will increase the efficiency further.

3.4. PROBLEM SOLUTION FIT



Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

A person from 5-100 years old

6. CUSTOMER CONSTRAINTS CC

Handwritten digit recognition not only has professional and commercial applications but also practical applications in our daily life. It can be of great help to the visually impaired to make the lives easier.

5. AVAILABLE SOLUTIONS AS

Handwritten digit recognition is the ability of a computer system to recognize the handwritten inputs like digits, characters etc. from a wide variety of sources like emails, papers, images, letters etc. This has been a topic of research for decades. Some of the research areas include signature verification, bank check processing, postal address interpretation from envelopes etc.

Explore AS, differentiate

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

2. JOBS-TO-BE-DONE / PROBLEMS J&P

Handwritten Digit Recognition has various real-life time uses.

To detect the vehicle number, banks for reading cheques, post offices for arranging letter, and many other tasks.

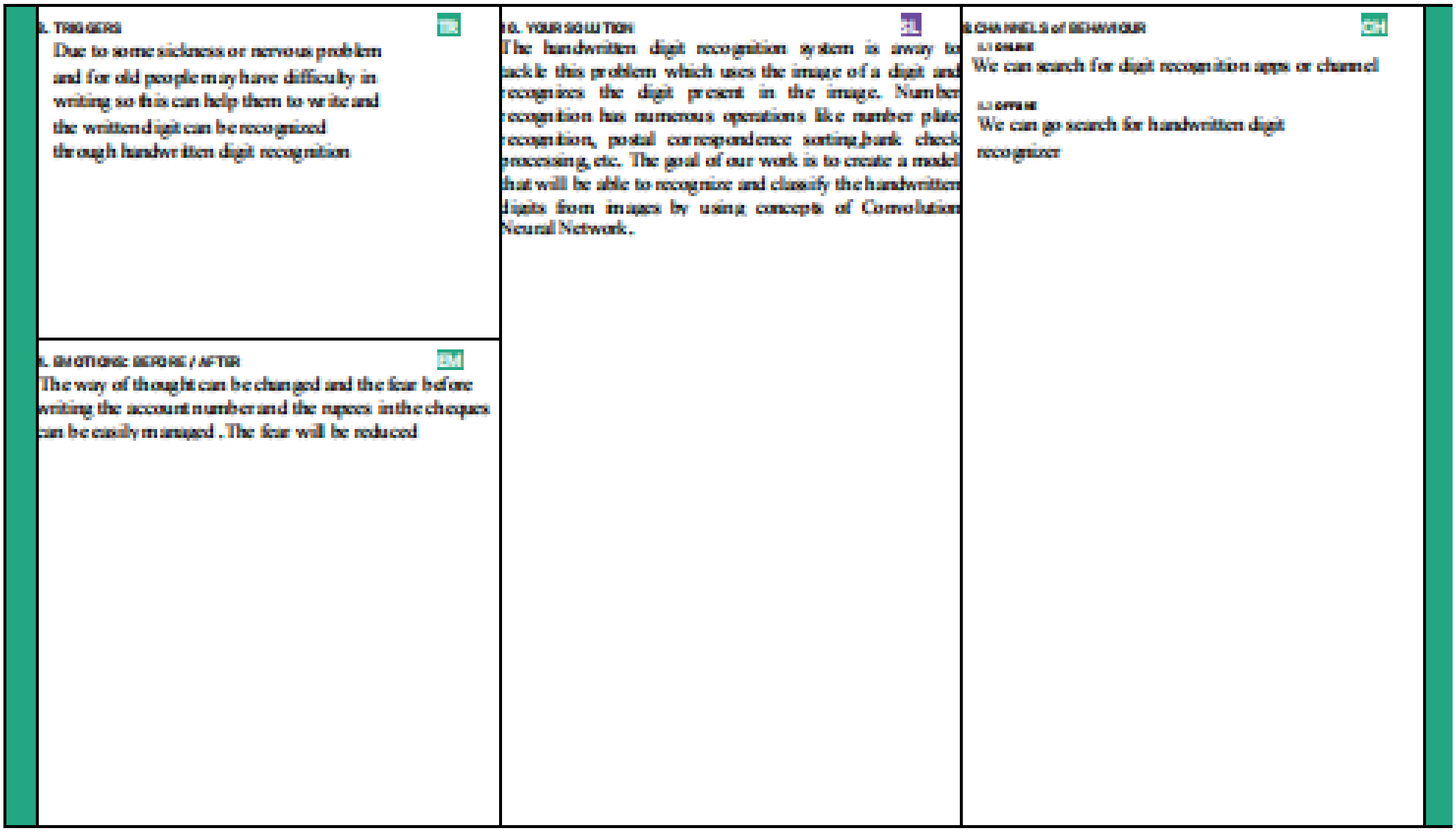
9. PROBLEM ROOT CAUSE RC

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

7. BEHAVIOUR BE

Characteristics include word spacing, line quality, consistency, connecting strokes, pen lifts, cursive letters, writing pressure, complete letters, diacritics, embellishments, slants and baseline habits

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



3. TRIGGERS TR

Due to some sickness or nervous problem and for old people may have difficulty in writing so this can help them to write and the written digit can be recognized through handwritten digit recognition

4. EMOTIONS: BEFORE / AFTER EM

The way of thought can be changed and the fear before writing the account number and the rupees in the cheques can be easily managed .The fear will be reduced

10. YOUR SOLUTION SL

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. Number recognition has numerous operations like number plate recognition, postal correspondence sorting,bank check processing, etc. The goal of our work is to create a model that will be able to recognize and classify the handwritten digits from images by using concepts of Convolution Neural Network.

8.CHANNELS of BEHAVIOUR CH

8.1 ONLINE

We can search for digit recognition apps or channel

8.2 OFFLINE

We can go search for handwritten digit recognizer

CHAPTER 4

4.REQUIREMENT ANALYSIS

4.1. FUNCTIONAL REQUIREMENTS

FR No.	**Functional Requirement (Epic)**	**Sub Requirement (Story / Sub-Task)**
FR-1	Home page	• if new user , REGISTER • if already exist , SIGN IN • Documents or numbers in that document checked correctly
FR-2	User Registration	Enter Mail Id and other personal details required for Registering
FR-3	User Confirmation	Before scanning confirm whether that is used by you or not through Otp /mail
FR-4	Scanning the Image	Scan the written digit / write the digit in tab
FR-5	Result	The scanned digit is recognized as a correct number

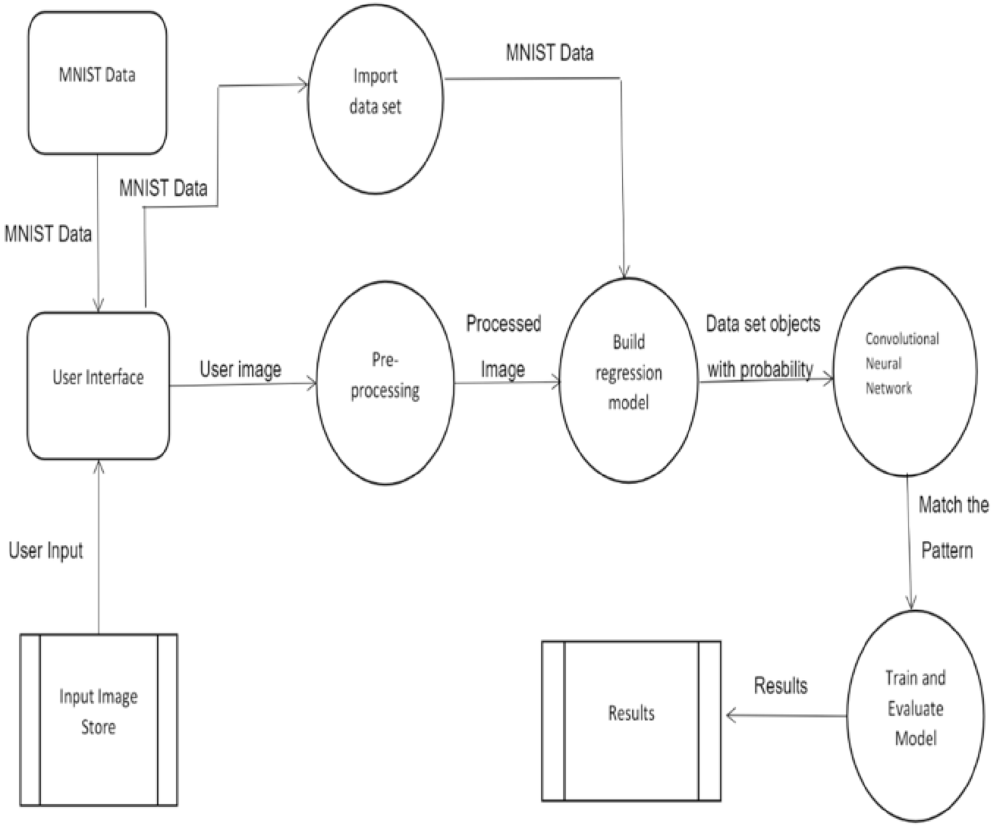
4.2. Non-Functional requirements

FR No.	**Non-Functional Requirement**	**Description**
NFR-1	**Usability**	It describes who, when, why, How this software system can be used
NFR-2	**Security**	Security functionality that ensures one of many different security properties of software is being satisfied .Security requirements are derived from industry standards, applicable laws ,and a history past vulnerabilities.
NFR-3	**Reliability**	It test whether the correct digit is recognized after scanning
NFR-4	**Performance**	It defines how well the software system accomplishes certain functions under specific condition.
NFR-5	**Availability**	It describes how long this software will be used and how long the IT system can be unavailable without impacting operations.
NFR-6	**Scalability**	It is the measure of a system ability to increase or decrease in performance and cost in response

CHAPTER 5

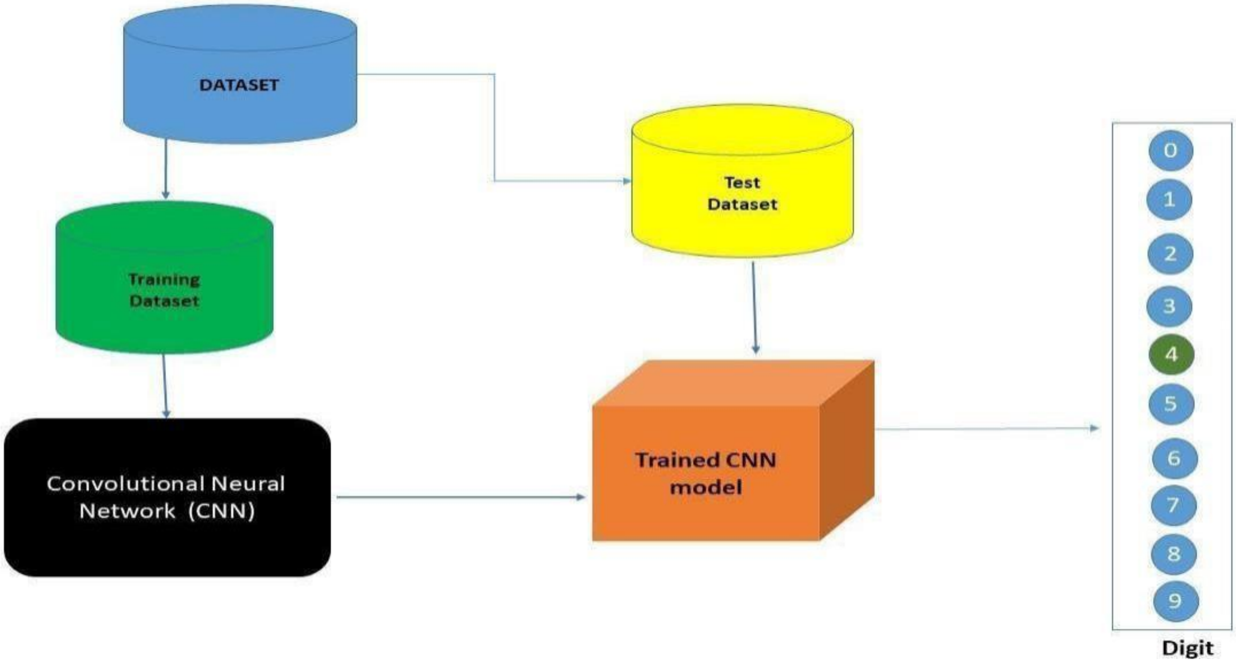
5. PROJECT DESIGN

5.1 Data Flow Diagrams



5.2 Solution & Technical Architecture

The architectural diagram of the model is as below and the Technology used is shown in table1 & table2



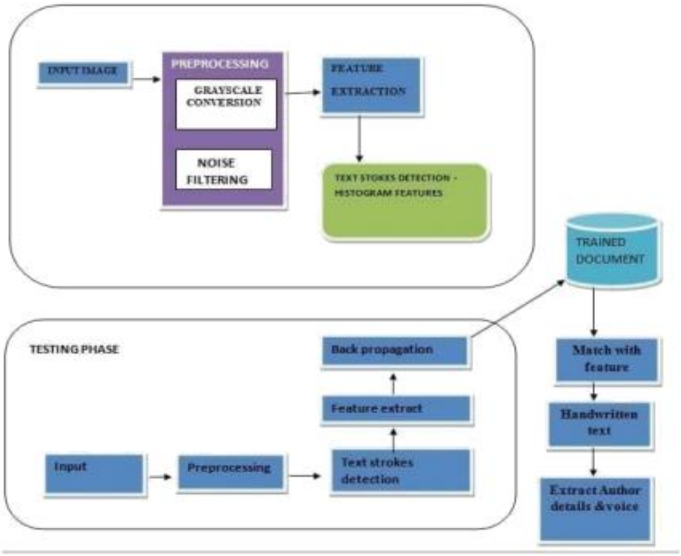
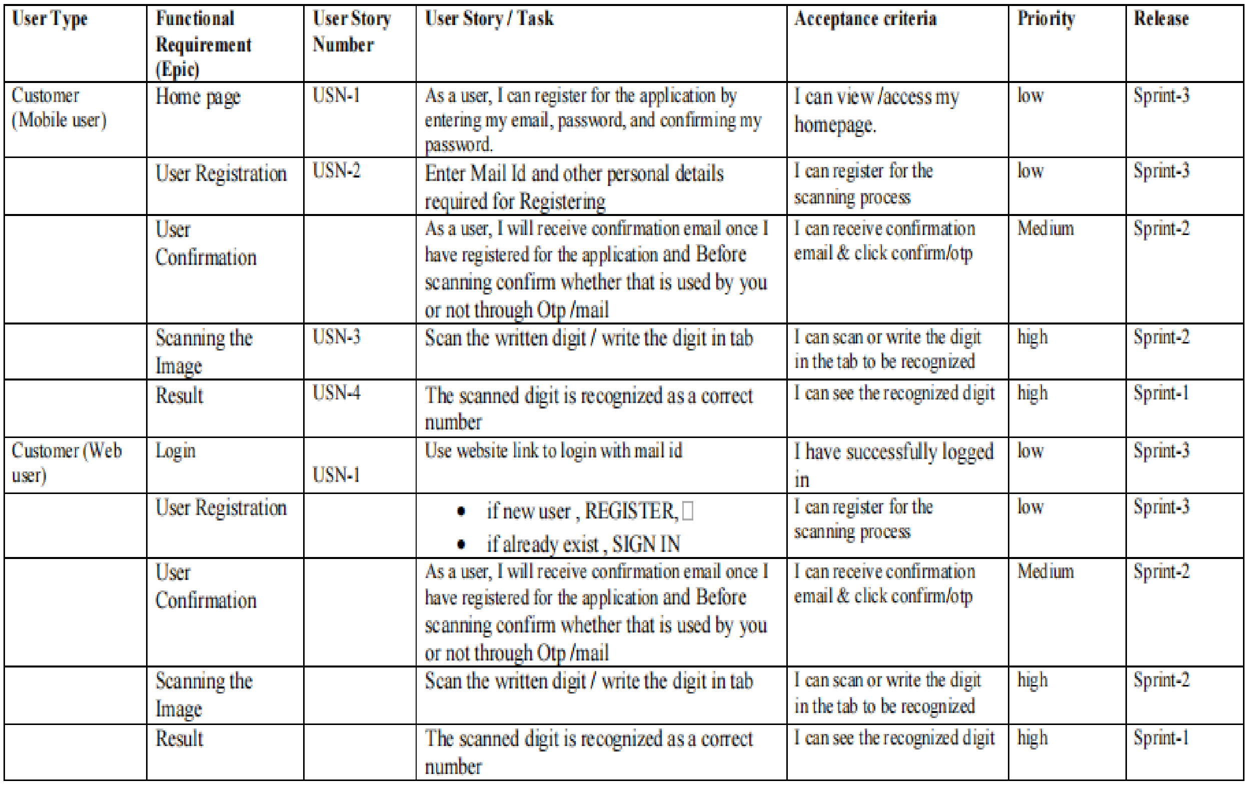


FIG. 1. BLOCK DIAGRAM

5.2 User Stories

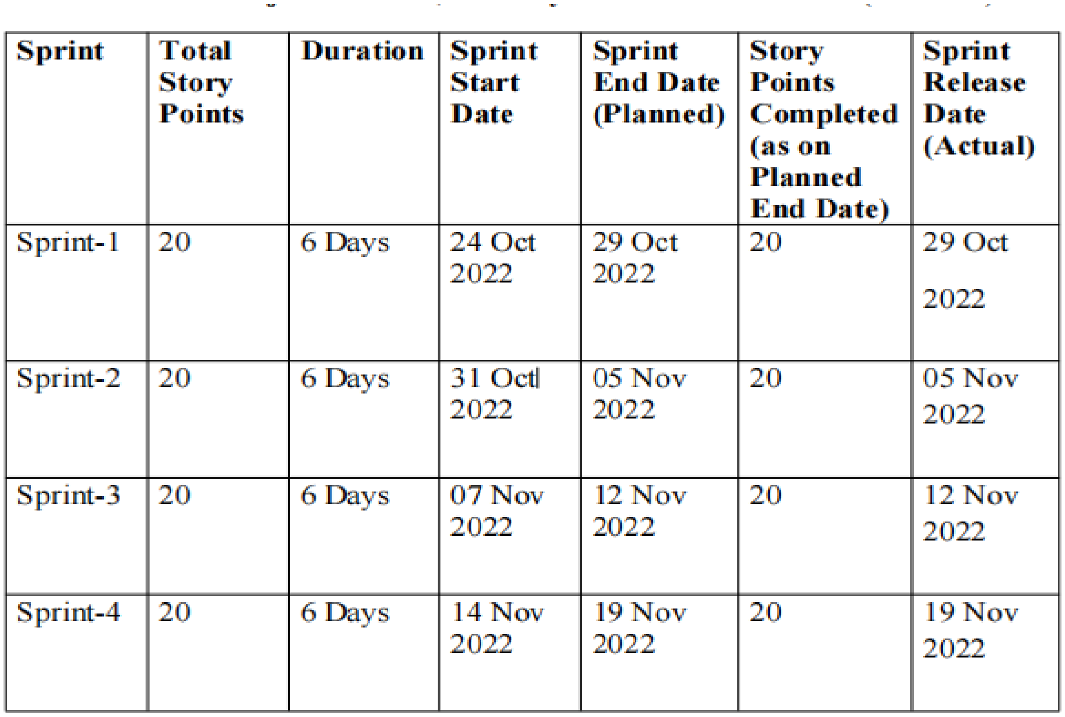
User Stories



User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Home page	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can view /access my homepage.	low	Sprint-3
	User Registration	USN-2	Enter Mail Id and other personal details required for Registering	I can register for the scanning process	low	Sprint-3
	User Confirmation		As a user, I will receive confirmation email once I have registered for the application and Before scanning confirm whether that is used by you or not through Otp /mail	I can receive confirmation email & click confirm/otp	Medium	Sprint-2
	Scanning the Image	USN-3	Scan the written digit / write the digit in tab	I can scan or write the digit in the tab to be recognized	high	Sprint-2
	Result	USN-4	The scanned digit is recognized as a correct number	I can see the recognized digit	high	Sprint-1
Customer (Web user)	Login	USN-1	Use website link to login with mail id	I have successfully logged in	low	Sprint-3
	User Registration		• if new user , REGISTER, □ • if already exist , SIGN IN	I can register for the scanning process	low	Sprint-3
	User Confirmation		As a user, I will receive confirmation email once I have registered for the application and Before scanning confirm whether that is used by you or not through Otp /mail	I can receive confirmation email & click confirm/otp	Medium	Sprint-2
	Scanning the Image		Scan the written digit / write the digit in tab	I can scan or write the digit in the tab to be recognized	high	Sprint-2
	Result		The scanned digit is recognized as a correct number	I can see the recognized digit	high	Sprint-1

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation



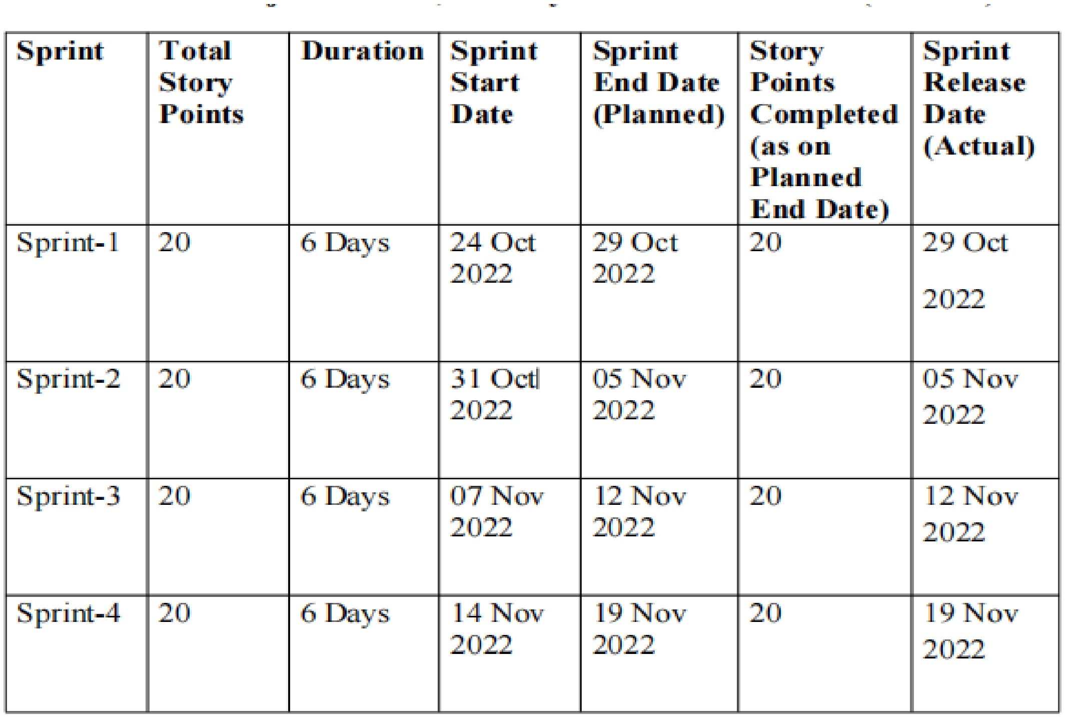
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

Velocity:

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

6.2 Sprint Delivery Schedule

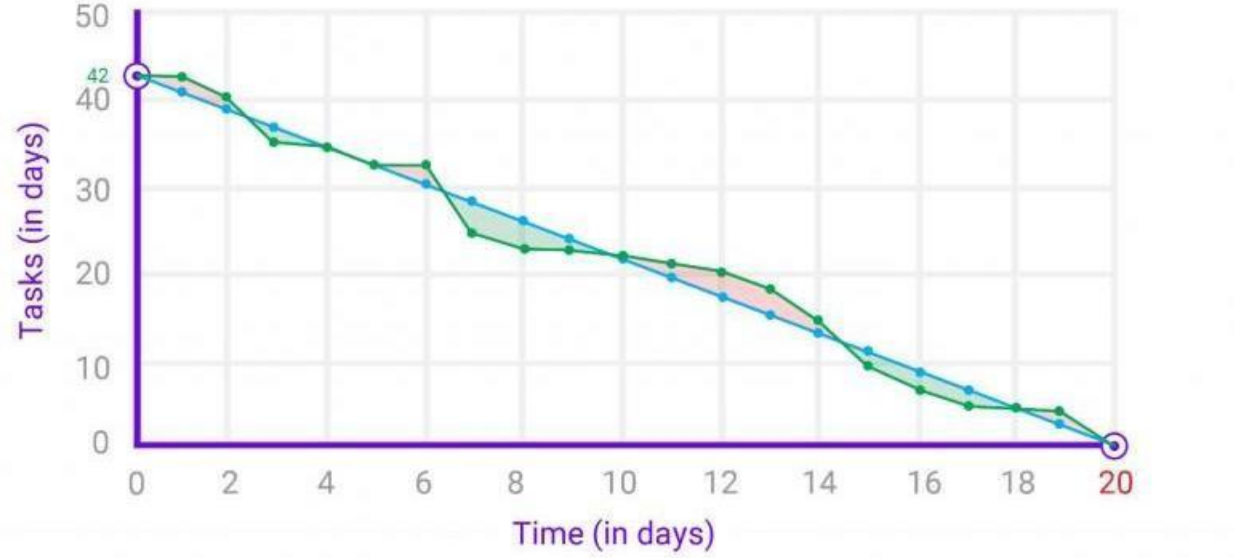


Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Octl 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

6.3 Reports from JIRA

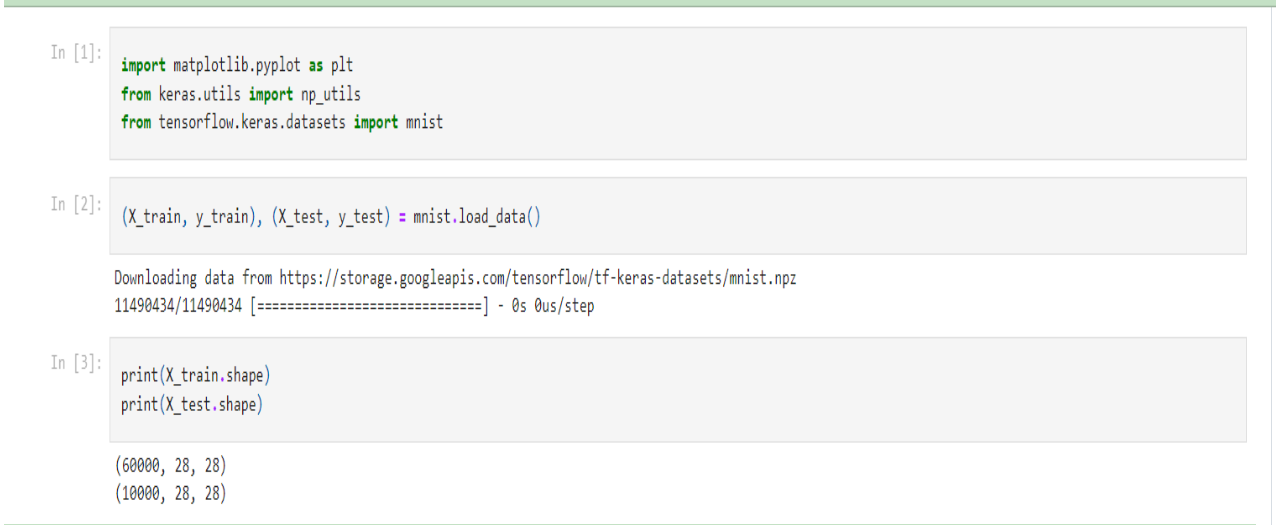
Burndown Chart:

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.

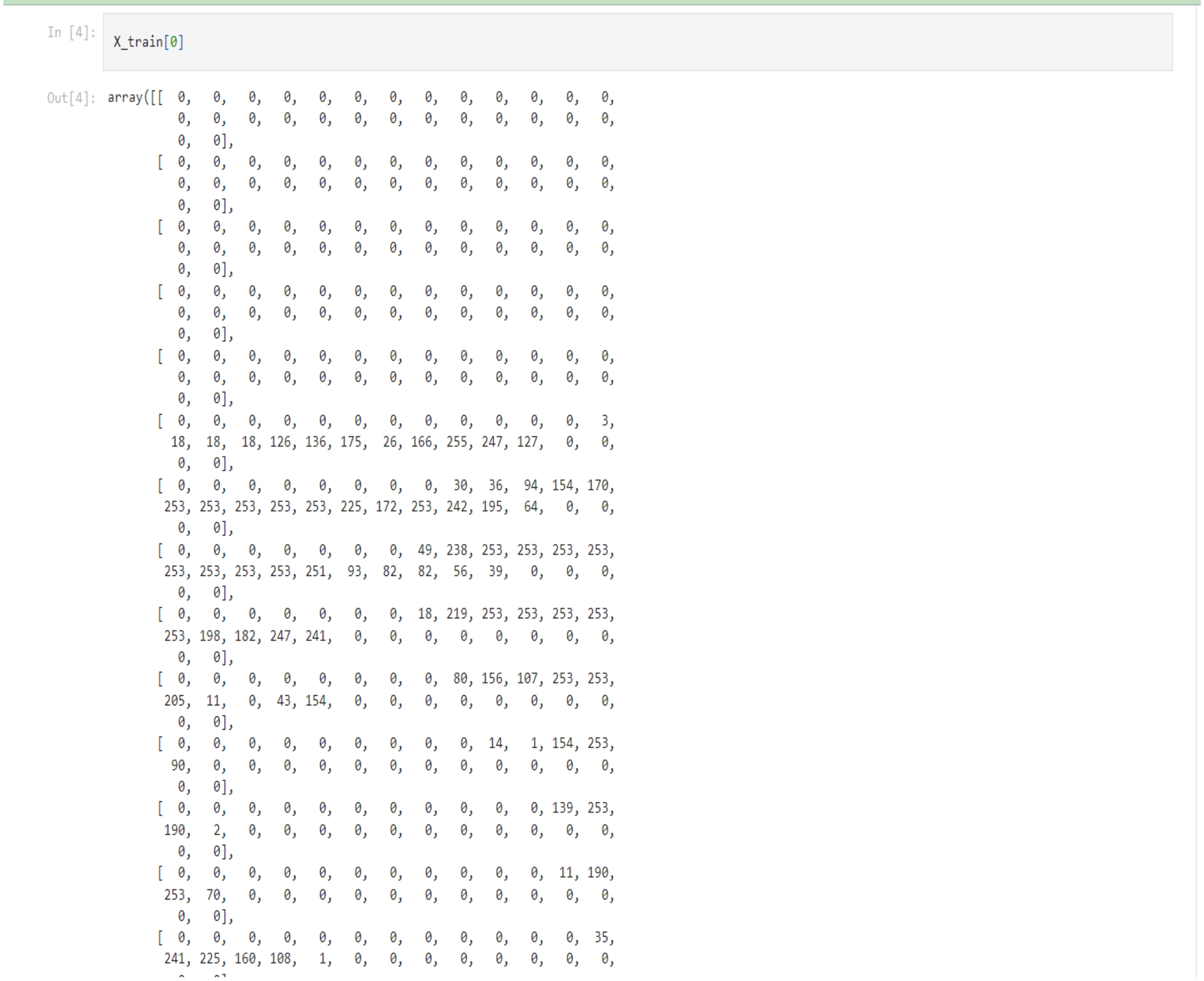


7. CODING & SOLUTIONING (Explain the features added in the project along with code)

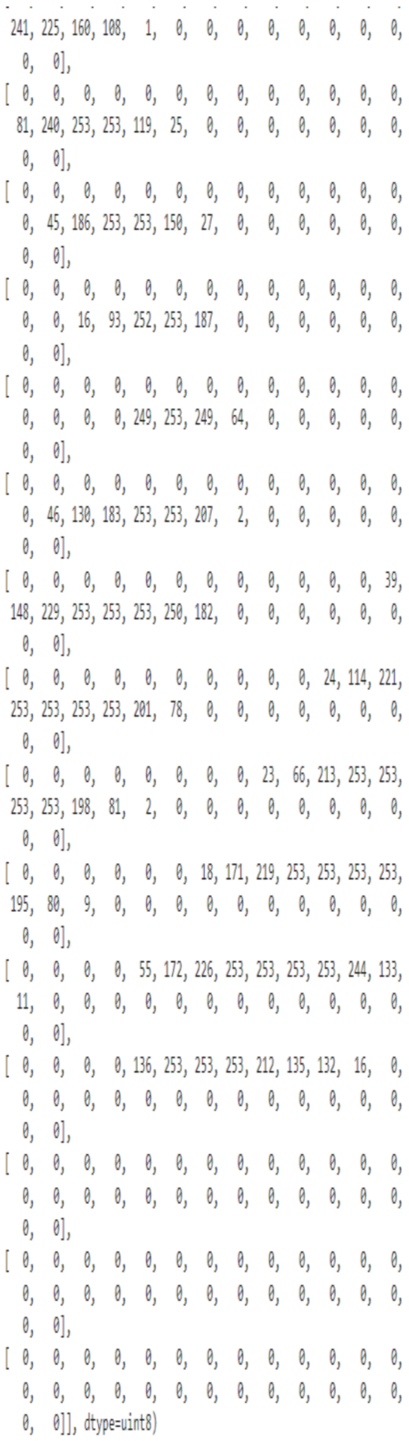
7.1 Feature 1



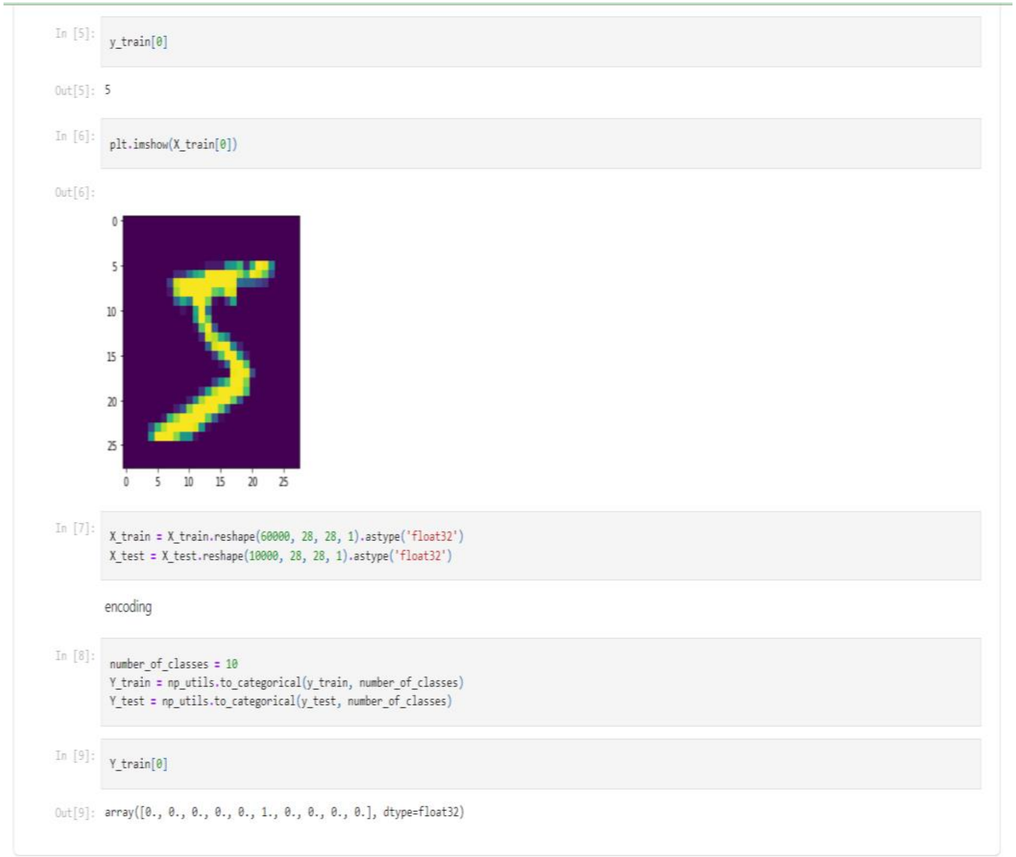
```
In [1]: import matplotlib.pyplot as plt
        from keras.utils import np_utils
        from tensorflow.keras.datasets import mnist

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

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

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

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



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

Out[4]: array([[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
                  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
                  0,   0],
               [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
                  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
                  0,   0],
               [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
                  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
                  0,   0],
               [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
                  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
                  0,   0],
               [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
                  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
                  0,   0],
               [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   3,
                 18,  18,  18, 126, 136, 175,  26, 166, 255, 247, 127,   0,   0,
                  0,   0],
               [  0,   0,   0,   0,   0,   0,   0,   0,  30,  36,  94, 154, 170,
                253, 253, 253, 253, 253, 225, 172, 253, 242, 195,  64,   0,   0,
                  0,   0],
               [  0,   0,   0,   0,   0,   0,   0,  49, 238, 253, 253, 253, 253,
                253, 253, 253, 253, 251,  93,  82,  82,  56,  39,   0,   0,   0,
                  0,   0],
               [  0,   0,   0,   0,   0,   0,   0,  18, 219, 253, 253, 253, 253,
                253, 198, 182, 247, 241,   0,   0,   0,   0,   0,   0,   0,   0,
                  0,   0],
               [  0,   0,   0,   0,   0,   0,   0,   0,  80, 156, 107, 253, 253,
                205,  11,   0,  43, 154,   0,   0,   0,   0,   0,   0,   0,   0,
                  0,   0],
               [  0,   0,   0,   0,   0,   0,   0,   0,   0,  14,   1, 154, 253,
                 90,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
                  0,   0],
               [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0, 139, 253,
                190,   2,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
                  0,   0],
               [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,  11, 190,
                253,  70,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
                  0,   0],
               [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,  35,
                241, 225, 160, 108,   1,   0,   0,   0,   0,   0,   0,   0,   0,
```



```
 241, 225, 160, 108,   1,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
  81, 240, 253, 253, 119,  25,   0,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,  45, 186, 253, 253, 150,  27,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0,  16,  93, 252, 253, 187,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0,   0,   0, 249, 253, 249,  64,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,  46, 130, 183, 253, 253, 207,   2,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,  39,
 148, 229, 253, 253, 253, 250, 182,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,  24, 114, 221,
 253, 253, 253, 253, 201,  78,   0,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,  23,  66, 213, 253, 253,
 253, 253, 198,  81,   2,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,  18, 171, 219, 253, 253, 253, 253,
 195,  80,   9,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,  55, 172, 226, 253, 253, 253, 253, 244, 133,
  11,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0, 136, 253, 253, 253, 212, 135, 132,  16,   0,
   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0]], dtype=uint8)
```



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

Out[5]: 5

In [6]: plt.imshow(X_train[0])

Out[6]:
```

```
In [7]: X_train = X_train.reshape(60000, 28, 28, 1).astype('float32')
        X_test = X_test.reshape(10000, 28, 28, 1).astype('float32')
```

encoding

```
In [8]: number_of_classes = 10
        Y_train = np_utils.to_categorical(y_train, number_of_classes)
        Y_test = np_utils.to_categorical(y_test, number_of_classes)

In [9]: Y_train[0]

Out[9]: array([0., 0., 0., 0., 0., 1., 0., 0., 0., 0.], dtype=float32)
```

7.2 Feature 2

Code.css

```
@import
url("https://fonts.googleapis.com/css2?family=Overpass:wght@200;300;400;500;600;700
;900&display=swap");
* {
  padding: 0;
  margin: 0;
}

body {
  color: black;
  font-family: "Overpass", sans-serif;
}
```

```
.container {
  width: 100%;
  height: 100%;
  display: flex;
  flex-direction: column;
  justify-content: center;
  align-items: center;
  background-color: white;
}
.container .heading {
  margin-top: -2rem;
  padding-bottom: 2rem;
  width: fit-content;
  text-align: center;
}
.container .heading h1 {
  font-size: 3rem;
  font-weight: 550;
}
.container .heading h2 {
  font-size: 1rem;
  color: rgb(90, 88, 88);
}
.container .sub_container1 {
  box-shadow: 0 0 20px rgb(172, 170, 170);
  width: 40rem;
  height: 25rem;
  padding: 1.5rem;
}
.container .sub_container2 {
  background-color: rgba(190, 190, 190, 0.5);
  width: 100%;
  height: 100%;
  display: flex;
  border: 1px dashed black;
  justify-content: center;
  align-items: center;
}
.container .sub_container2 .upload {
  display: flex;
  justify-content: center;
  align-items: center;
  width: 8rem;
  height: -webkit-fit-content;
  height: -moz-fit-content;
  height: fit-content;

  border-radius: 6px;
  color: white;
  background-color: rgb(114, 96, 182);
  box-shadow: 0 5px 10px rgb(146, 135, 247);
}

.container .sub_container2 #loading {
  display: none;
  justify-content: center;
  align-items: center;
  width: 10rem;
  height: auto;
  position: absolute;
}
```

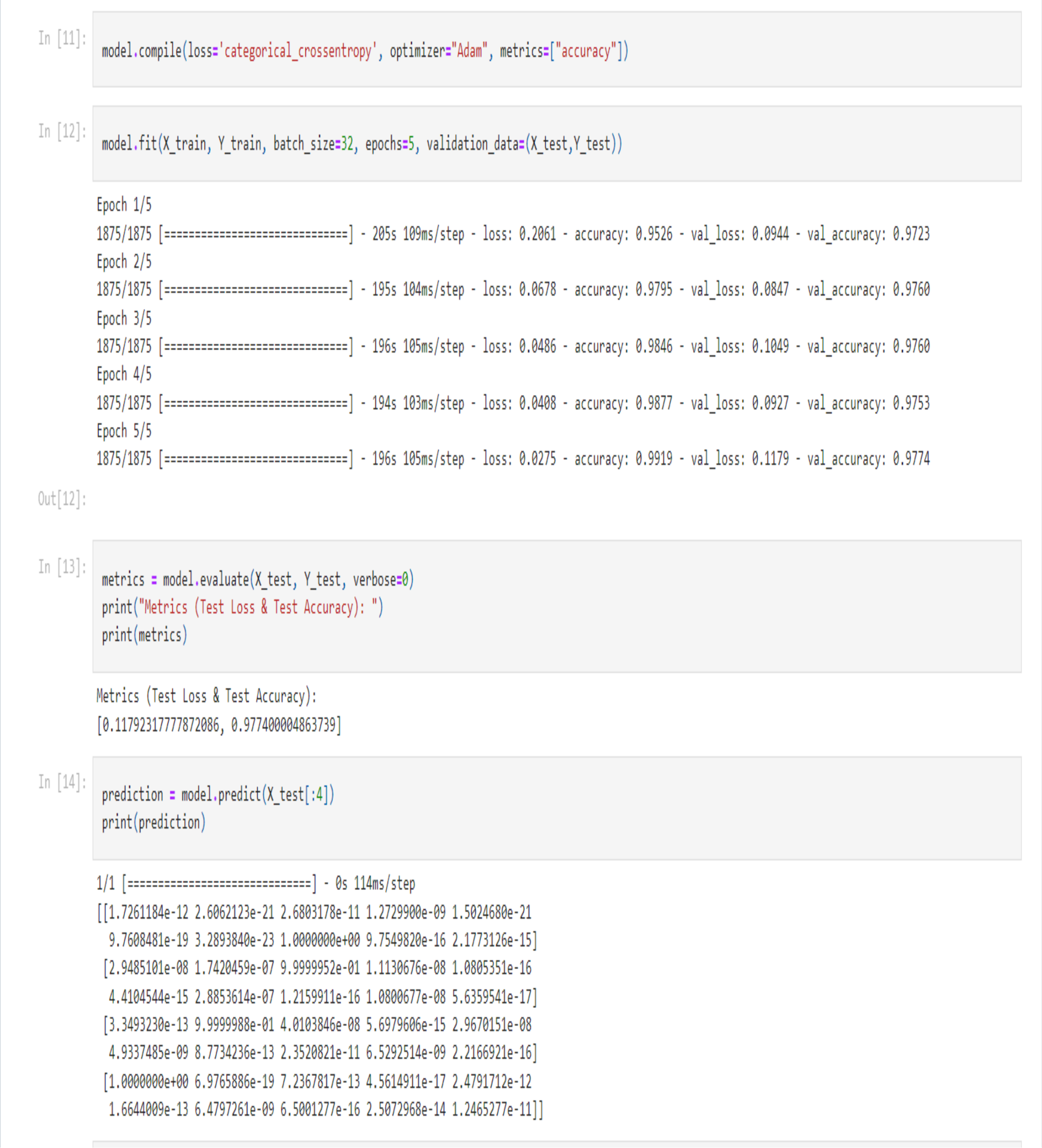
```
.container .sub_container2 .upload label {
  font-size: 1rem;
  font-weight: 600;
  color: white;
  height: 100%;
  width: 100%;
  padding: 10px;
  display: block;
}
.container .sub_container2 .upload svg
{
  height: 15px;
  width: auto;
  padding-right: 8px;
  margin-bottom: -2px;
}


@media screen and (max-width:700px) {
  .container .sub_container1 {
    height: 20rem;
    width: 18rem;
    margin-top: 3.5rem;
    margin-bottom: -8rem;


  }
  .container .heading h1 {
    margin-top: -6rem;
    font-size: 2rem;
    padding-bottom:1rem ;
  }
}
```

Code.html

```
<html>
    <head>
        <title>Handwritten Digit Recognizer</title>
        <link rel="stylesheet" href="{{url_for('static',filename='css/main.css')}}">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <script src="https://unpkg.com/feather-icons"></script>
    </head>
    <body>
        <div class="container">
            <div class="heading">
                <h1>Handwritten Digit Recognizer</h1>
                <h2>Easy analyze and detect handwritten digits</h2>
            </div>
            <div class="sub_container1">
                <div class="sub_container2">
                    <form class="upload" action="/confirm" method="post" enctype="multipart/form-data">
                        <!-- select -->
                        <label id="label" for="upload-image"><i data-feather="file-plus"></i>Select File</label>
                        <!-- upload -->
                        <input type="file" name="photo" id="upload-image" hidden>
                    </form>
                    <img id="loading" src="{{url_for('static',filename='images/loading.gif')}}">
                </div>
            </div>
        </div>
    <script rel="text/javascript">
      feather.replace();

      form = document.querySelector(".upload")
      label = document.querySelector('#label');
      loading = document.querySelector("#loading")
      select = document.querySelector("#upload-image");

      console.log("working...")
      select.addEventListener('change', (e) => {
        e.preventDefault();

        form.style.visibility = "hidden";
        form.submit()
        loading.style.display = 'flex';
      })
    </script>
  </body>
</html>
```

8. TESTING

8.1 Test Cases



```
In [11]: model.compile(loss='categorical_crossentropy', optimizer="Adam", metrics=["accuracy"])
```

```
In [12]: model.fit(X_train, Y_train, batch_size=32, epochs=5, validation_data=(X_test,Y_test))
```

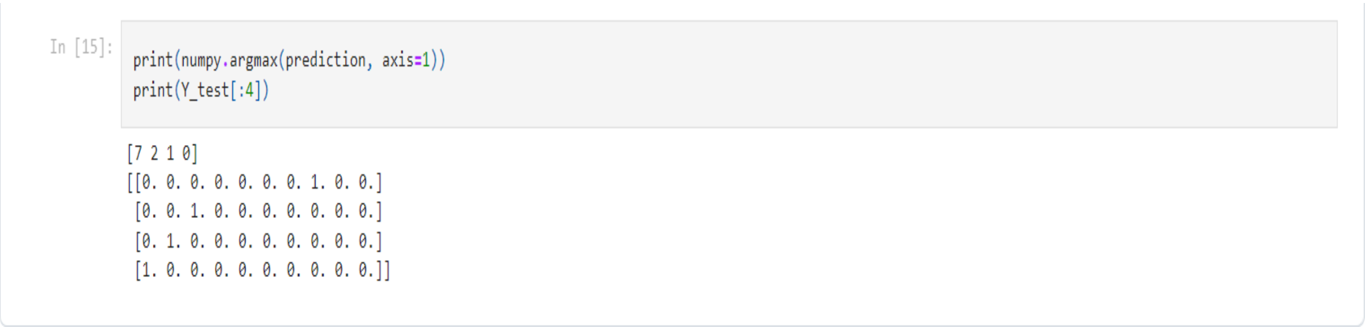
```
Epoch 1/5
1875/1875 [==============================] - 205s 109ms/step - loss: 0.2061 - accuracy: 0.9526 - val_loss: 0.0944 - val_accuracy: 0.9723
Epoch 2/5
1875/1875 [==============================] - 195s 104ms/step - loss: 0.0678 - accuracy: 0.9795 - val_loss: 0.0847 - val_accuracy: 0.9760
Epoch 3/5
1875/1875 [==============================] - 196s 105ms/step - loss: 0.0486 - accuracy: 0.9846 - val_loss: 0.1049 - val_accuracy: 0.9760
Epoch 4/5
1875/1875 [==============================] - 194s 103ms/step - loss: 0.0408 - accuracy: 0.9877 - val_loss: 0.0927 - val_accuracy: 0.9753
Epoch 5/5
1875/1875 [==============================] - 196s 105ms/step - loss: 0.0275 - accuracy: 0.9919 - val_loss: 0.1179 - val_accuracy: 0.9774
Out[12]:
```

```
In [13]: metrics = model.evaluate(X_test, Y_test, verbose=0)
         print("Metrics (Test Loss & Test Accuracy): ")
         print(metrics)
```

```
Metrics (Test Loss & Test Accuracy):
[0.11792317777872086, 0.977400004863739]
```

```
In [14]: prediction = model.predict(X_test[:4])
         print(prediction)
```

```
1/1 [==============================] - 0s 114ms/step
[[1.7261184e-12 2.6062123e-21 2.6803178e-11 1.2729900e-09 1.5024680e-21
  9.7608481e-19 3.2893840e-23 1.0000000e+00 9.7549820e-16 2.1773126e-15]
 [2.9485101e-08 1.7420459e-07 9.9999952e-01 1.1130676e-08 1.0805351e-16
  4.4104544e-15 2.8853614e-07 1.2159911e-16 1.0800677e-08 5.6359541e-17]
 [3.3493230e-13 9.9999988e-01 4.0103846e-08 5.6979606e-15 2.9670151e-08
  4.9337485e-09 8.7734236e-13 2.3520821e-11 6.5292514e-09 2.2166921e-16]
 [1.0000000e+00 6.9765886e-19 7.2367817e-13 4.5614911e-17 2.4791712e-12
  1.6644009e-13 6.4797261e-09 6.5001277e-16 2.5072968e-14 1.2465277e-11]]
```



In [15]:

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

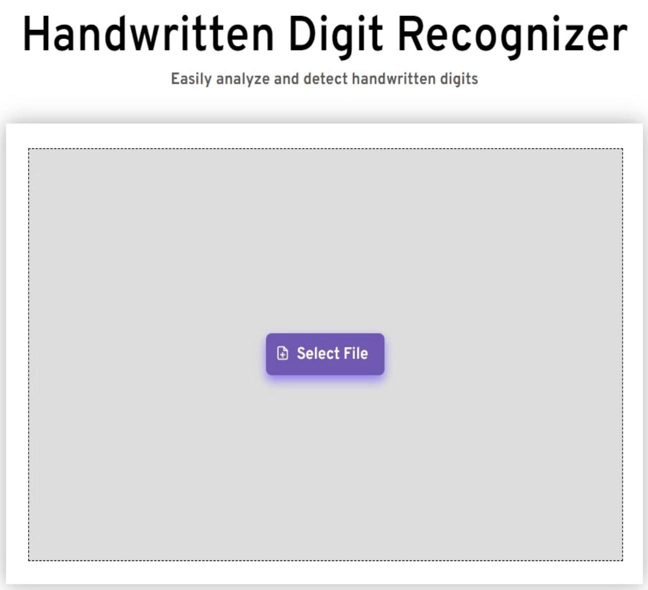
```
[7 2 1 0]
[[0. 0. 0. 0. 0. 0. 0. 1. 0. 0.]
 [0. 0. 1. 0. 0. 0. 0. 0. 0. 0.]
 [0. 1. 0. 0. 0. 0. 0. 0. 0. 0.]
 [1. 0. 0. 0. 0. 0. 0. 0. 0. 0.]]
```

8.2 User Acceptance Testing

1. DEFECT ANALYSIS

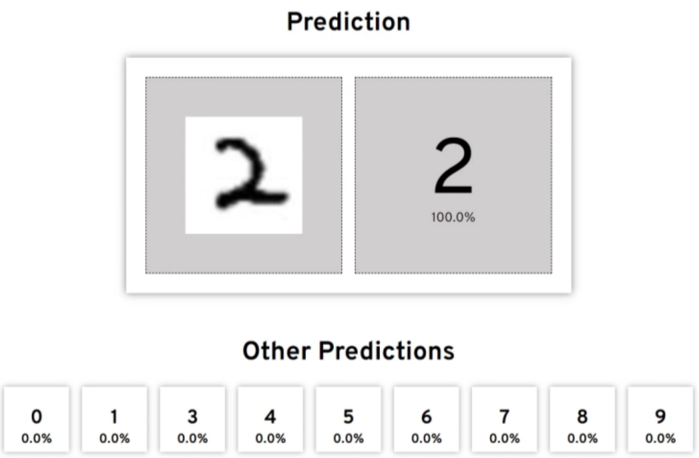
Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Total
By Design	1	0	1	0	2
Duplicate	0	0	0	0	0
External	0	0	2	0	2
Fixed	4	1	0	1	6
Not Reproduced	0	0	0	1	1
Skipped	0	0	0	1	1
Won’t Fix	1	0	1	0	2
Total	6	1	4	3	14

9. RESULTS



Handwritten Digit Recognizer

Easily analyze and detect handwritten digits



Prediction

2
100.0%

Other Predictions

```
0     1     3     4     5     6     7     8     9
0.0%  0.0%  0.0%  0.0%  0.0%  0.0%  0.0%  0.0%  0.0%
```

10.ADVANTAGES & 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

11.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 fifilled up by hand (tax forms) and so on. There is so much room for improvement, which can be implemented in subsequent versions.

Though the goal of our research is to create a model for digit recognition and classification, it can also be extended to letters and an individual’s handwriting. The existing methods in current image recognition use as inputs all the pixels of the image. The purpose of this work is to minimize the number of pixels by using as input the data extracted and calculated fromthe initial image. This will be used in many fields.

12.FUTURE SCOPE

The advance of handwriting processing results from a combination of various elements, for example: improvements in there cognition rates, the use of complex systems to integrate various kinds of information, and new technologies such as high quality high speed scanners and cheaper and more powerful CPUs. Some handwriting recognition system allows us to input our handwriting into the system.

Handwritten digit recognition is the ability of a computer system to recognize the handwritten inputs like digits, characters etc. from a wide variety of sources like emails, papers, images, letters etc. This has been a topic of research for decades. Some of the research areas include signature verification, bank check processing, postal address interpretation from envelopes etc.

13.APPENDIX

Source Code

recognizer.py

```
import os
import random
import string
from pathlib import Path
import numpy as np
from tensorflow.keras.models import load_model
from PIL import Image, ImageOps

def random_name_generator(n):
    return ".join(random.choices(string.ascii_uppercase + string.digits, k=n))
def recognize(image):
    model=load_model(Path("./model/model.h5"))
   img = Image.open(image).convert("L")
    img_name = random_name_generator(10) + '.jpg'

    if not os.path.exists(f"./static/data/"):
          os.mkdir(os.path.join('./static/', 'data'))
    img.save(Path(f"./static/data/{img_name}"))

    img = ImageOps.grayscale(img)
    img = ImageOps.invert(img)
    img = img.resize((28, 28))
   img2arr = np.array(img)
    img2arr = img2arr / 255.0
    img2arr = img2arr.reshape(1, 28, 28, 1)

    results  = model.predict(img2arr)
    best = np.argmax(results,axis = 1)[0]
    pred = list(map(lambda x: round(x*100, 2), results[0]))
   values = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
    others = list(zip(values, pred))
   best = others.pop(best)
   return best, others, img_name
```

app.py

```
from flask import Flask, render_template

app=Flask(__name__)

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

   app.run(debug=True)
```

GitHub

Github Link: https://github.com/IBM-EPBL/IBM-Project-4789-1658740332