Project Report Format

1. INTRODUCTION

1.1 Project Overview

Machine learning and deep learning plays an important role in computer technology and artificial intelligence. With the use of deep learning and machine learning, human effort can be reduced in recognizing, learning, predictions and many more areas. This article presents recognizing the handwritten digits (0 to 9) from the famous MNIST dataset, comparing classifiers like KNN, PSVM, NN and convolution neural network on basis of performance, accuracy, time, sensitivity, positive productivity, and specificity with using different parameters with the classifiers

1.2 Purpose

The handwritten digit recognition is the ability of computers to recognize human handwritten digits. It is a hard task for the machine because handwritten digits are not perfect and can be made with many different flavors. The handwritten digit recognition is the solution to this problem which uses the image of a digit and recognizes the digit present in the image.

2. LITERATURE SURVEY

2.1 Existing problem

“Handwritten digit recognition using deep learning algorithm.”

The reliance of humans over machines has never been so high such that from object classification in photographs to adding sound to silent movies everything can be performed with the help of deep learning and machine learning algorithms. Likewise, Handwritten text recognition is one of the significant areas of research and development with a streaming number of possibilities that could be attained.

Handwriting recognition (HWR), also known as Handwritten Text Recognition (HTR), is the ability to receive and interpret intelligible handwritten input from sources such as paper documents, photographs, touch-screens and other devices. Apparently, in this paper, we have performed handwritten digit recognition with the help of MNIST datasets using Multi-Layer Perceptron (MLP) and Convolution Neural Network (CNN) models. Our main objective is to compare the accuracy of the models stated above along with their execution time to get the best possible model for digit recognition.

“Recognition of Handwritten Digit using Convolutional Neural Network.”

Humans can see and visually sense the world around them by using their eyes and brains. Computer vision works on enabling computers to see and process images in the same way that human vision does. Several algorithms developed in the area of computer vision to recognize images.

The goal of our work will be to create a model that will be able to identify and determine the handwritten digit from its image with better accuracy. We aim to complete this using the concepts of Convolutional Neural Network and MNIST dataset. Though the goal is to create a model which can recognize the digits, we can extend it for letters and then a person's handwriting. Through this work, we aim to learn and practically apply the concepts of Convolutional Neural Networks.

"Handwritten Digit Recognition using Machine Learning Algorithms."

Handwritten character recognition is one of the practically important issues in pattern recognition applications. The applications of digit recognition include in postal mail sorting, bank check processing, form data entry, etc. The heart of the problem lies within the ability to develop an efficient algorithm that can recognize hand written digits and which is submitted by users by the way of a scanner, tablet, and other digital devices. This paper presents an approach to off-line handwritten digit recognition based on different machine learning technique.

The main objective of this paper is to ensure effective and reliable approaches for recognition of handwritten digits. Several machines learning algorithm namely, Multilayer Perceptron, Random Forest, J48 and Random Tree has been used for the recognition of digits using WEKA.

"Effective Handwritten digit recognition using deep neural network."

This paper proposed a simple neural network approach towards handwritten digit recognition using convolution. With machine learning algorithms like KNN, SVM/SOM, recognizing digits is considered as one of the unsolvable tasks due to its distinctiveness in the style of writing.

In this paper, Convolution Neural Networks are implemented with an MNIST dataset of 70000 digits with 250 distinct forms of writings. The proposed method achieved 98.51% accuracy for real-world handwritten digit prediction with less than 0.1 % loss on training with 60000 digits while 10000 under validation.

2.2 References

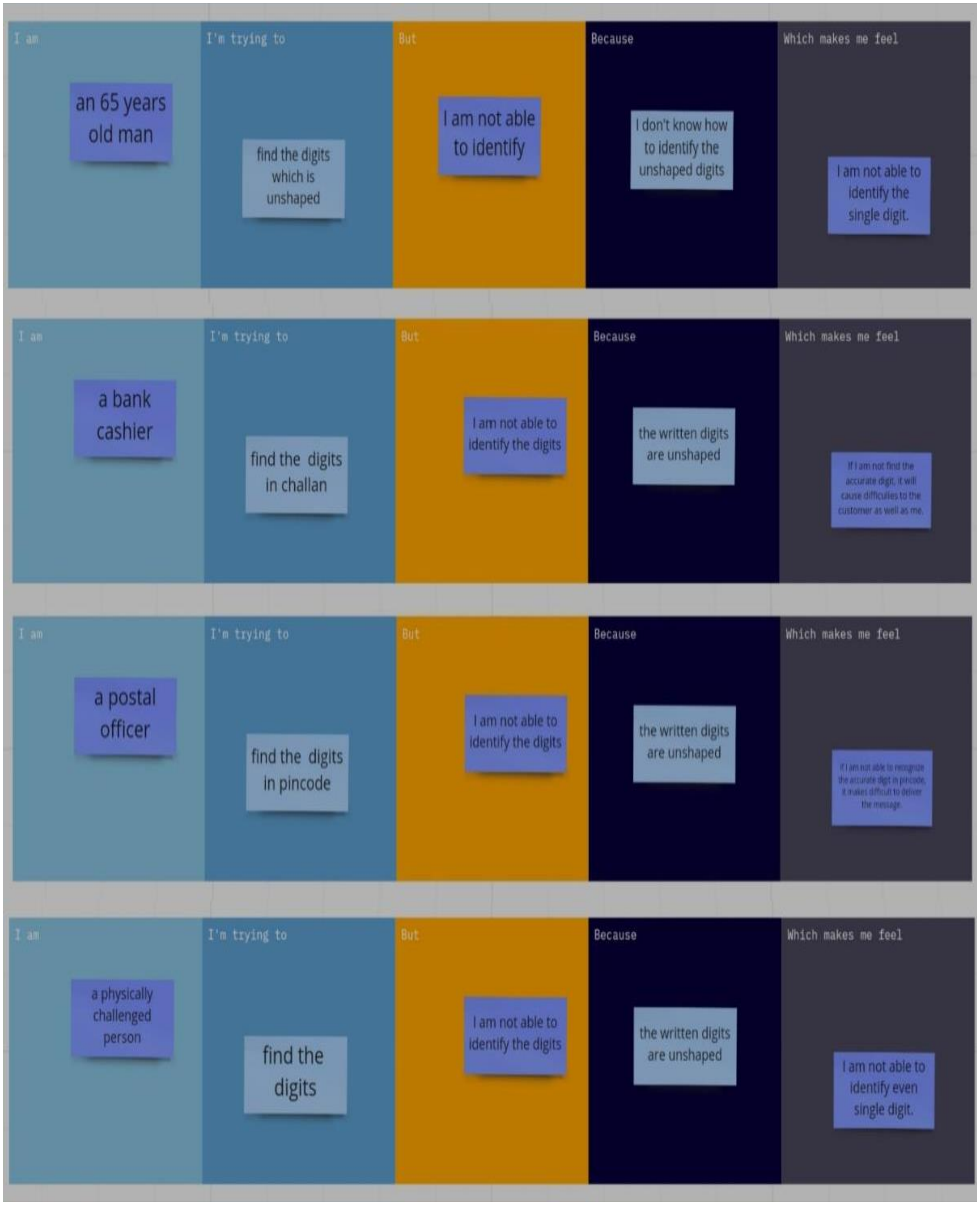
Reference 1: https://arxiv.org/pdf/2106.12614.pdf
Reference 2: https://core.ac.uk/download/pdf/231148505.pdf
Reference 3: https://globaljournals.org/GJCST_Volume18/3-Handwritten-Digit-Recognition.pdf
Reference 4: http://www.warse.org/IJATCSE/static/pdf/file/ijatcse66922020.pdf

2.3 Problem Statement Definition

Create a problem statement to understand your customer's point of view. The Customer Problem Statement template helps you focus on what matters to create experiences people will love.

A well-articulated customer problem statement allows you and your team to find the ideal solution for the challenges your customers face. Throughout the process,

you’ll also be able to empathize with your customers, which helps you better understand how they perceive your product or service.



I am	I'm trying to	But	Because	Which makes me feel
an 65 years old man	find the digits which is unshaped	I am not able to identify	I don't know how to identify the unshaped digits	I am not able to identify the single digit.

I am	I'm trying to	But	Because	Which makes me feel
a bank cashier	find the digits in challan	I am not able to identify the digits	the written digits are unshaped	If I am not find the accurate digit, it will cause difficulties to the customer as well as me.

I am	I'm trying to	But	Because	Which makes me feel
a postal officer	find the digits in pincode	I am not able to identify the digits	the written digits are unshaped	If I am not able to recognize the accurate digit in pincode, it makes difficult to deliver the message.

I am	I'm trying to	But	Because	Which makes me feel
a physically challenged person	find the digits	I am not able to identify the digits	the written digits are unshaped	I am not able to identify even single digit.

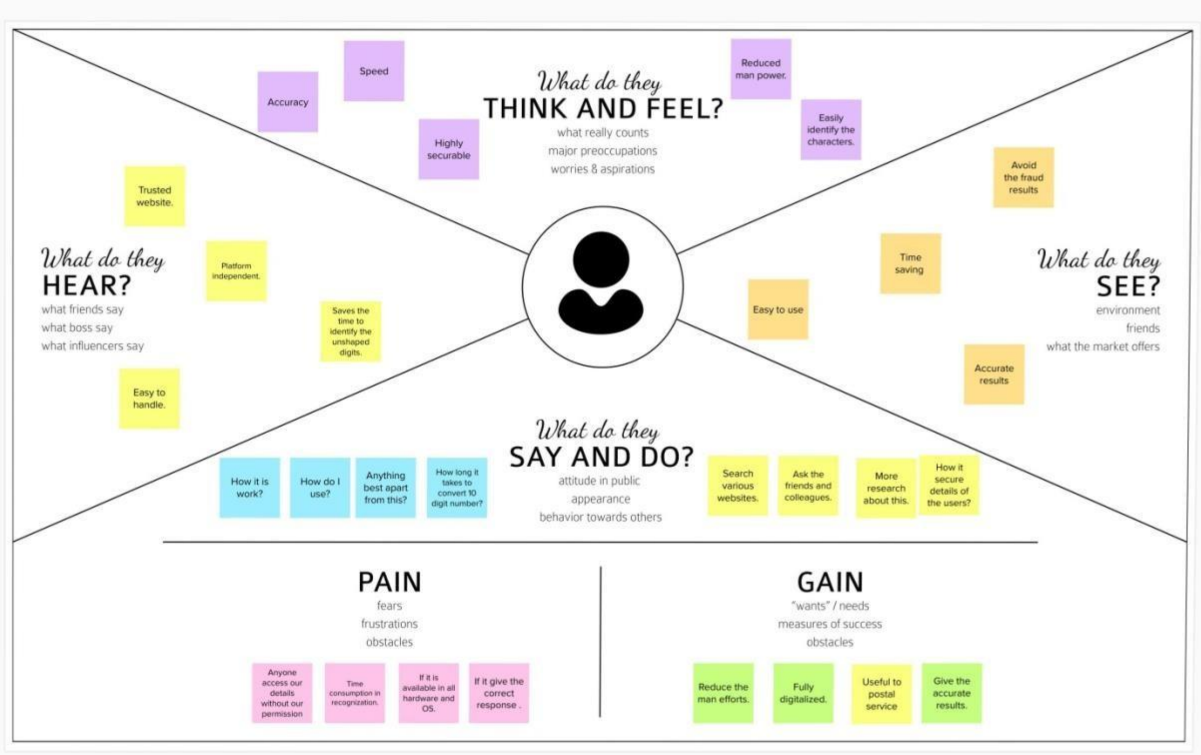
Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	A 65 yearsold man	find the digits	I am not able to identify	I don't know how to identify the unshaped digits	I am not able to identify even a single digit.
PS-2	A postalofficer	find the digits in pin code	I am not able to identify the digits	the written digits are unshaped	If I am not able to recognize the accurate digits in pin code, it makes difficult to deliver the post in accurate location.
PS-3	A bankcashier	find the digits in challan	I am not able to identify the digits	the written digits are unshaped	If I not find the accurate digit it will cause difficulties to the customer as well me.
PS-4	A physically challenged person	find the digits	I am not able to identify the digits	the written digits are unshaped	I am not able to identify the digits

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviors and attitudes.

It is a useful tool to helps teams better understand their users.
Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.



Reference:
https://app.mural.co/invitation/mural/virtualworld6297/1662195099256?sender=u9e2cb0f0137232a310757183&key=74a671cf-34c0-4561-b6fc-ced135f6799f

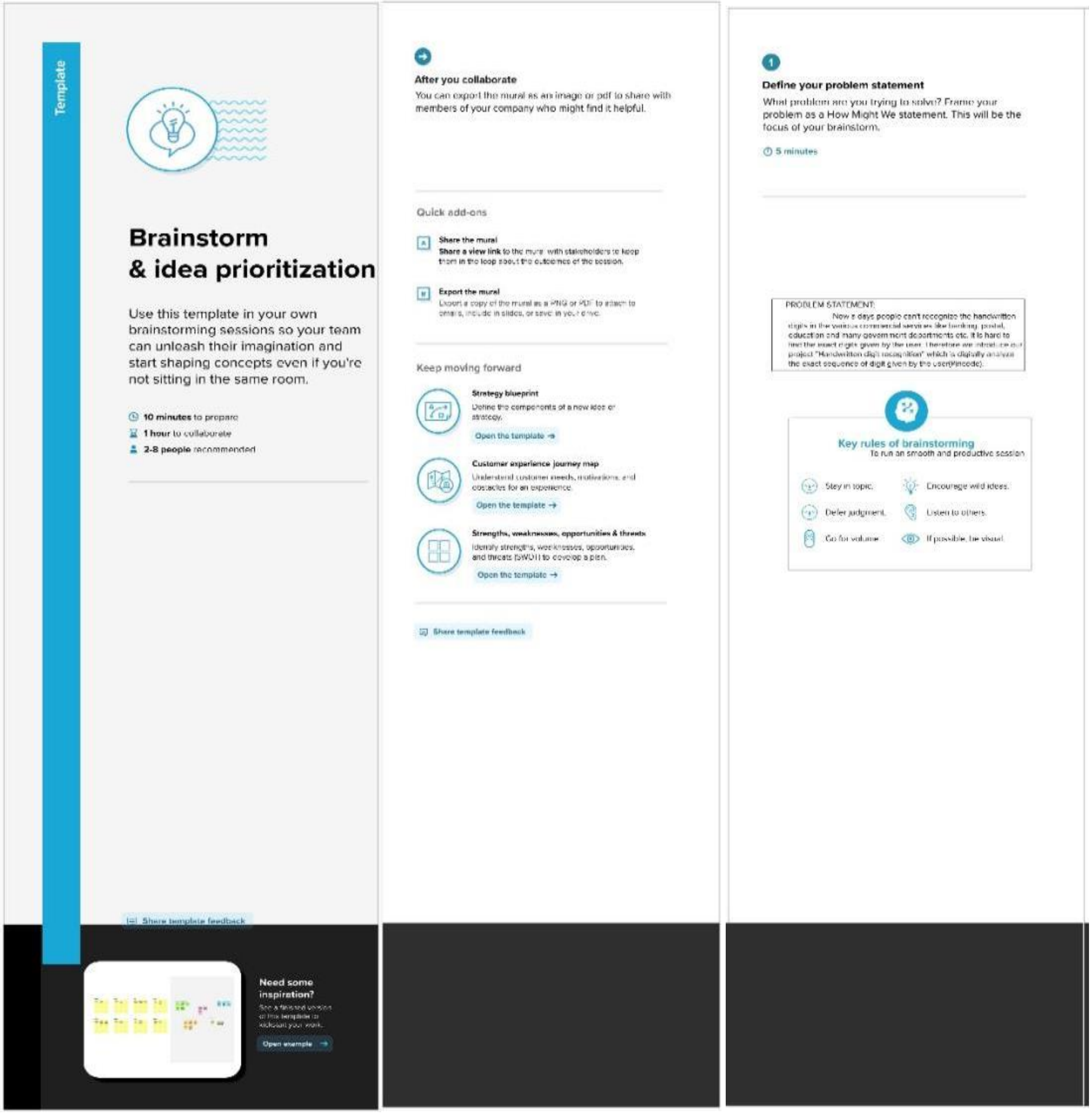
3.2 Ideation & Brainstorming

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich number of creative solutions.

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.

Reference: https://www.mural.co/templates/empathy-map-canvas

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



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

Share template feedback

Need some inspiration?

See a finished version of this template to kickstart your work.

Open example →

After you collaborate

You can export the mural as an image or pdf to share with members of your company who might find it helpful.

Quick add-ons

Share the mural
Share a view link to the mural with stakeholders to keep them in the loop about the outcomes of the session.

Export the mural
Export a copy of the mural as a PNG or PDF to attach to emails, include in slides, or save in your drive.

Keep moving forward

Strategy blueprint
Define the components of a new idea or strategy.
Open the template →

Customer experience journey map
Understand customer needs, motivations, and obstacles for an experience.
Open the template →

Strengths, weaknesses, opportunities & threats
Identify strengths, weaknesses, opportunities, and threats (SWOT) to develop a plan.
Open the template →

Share template feedback

1

Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

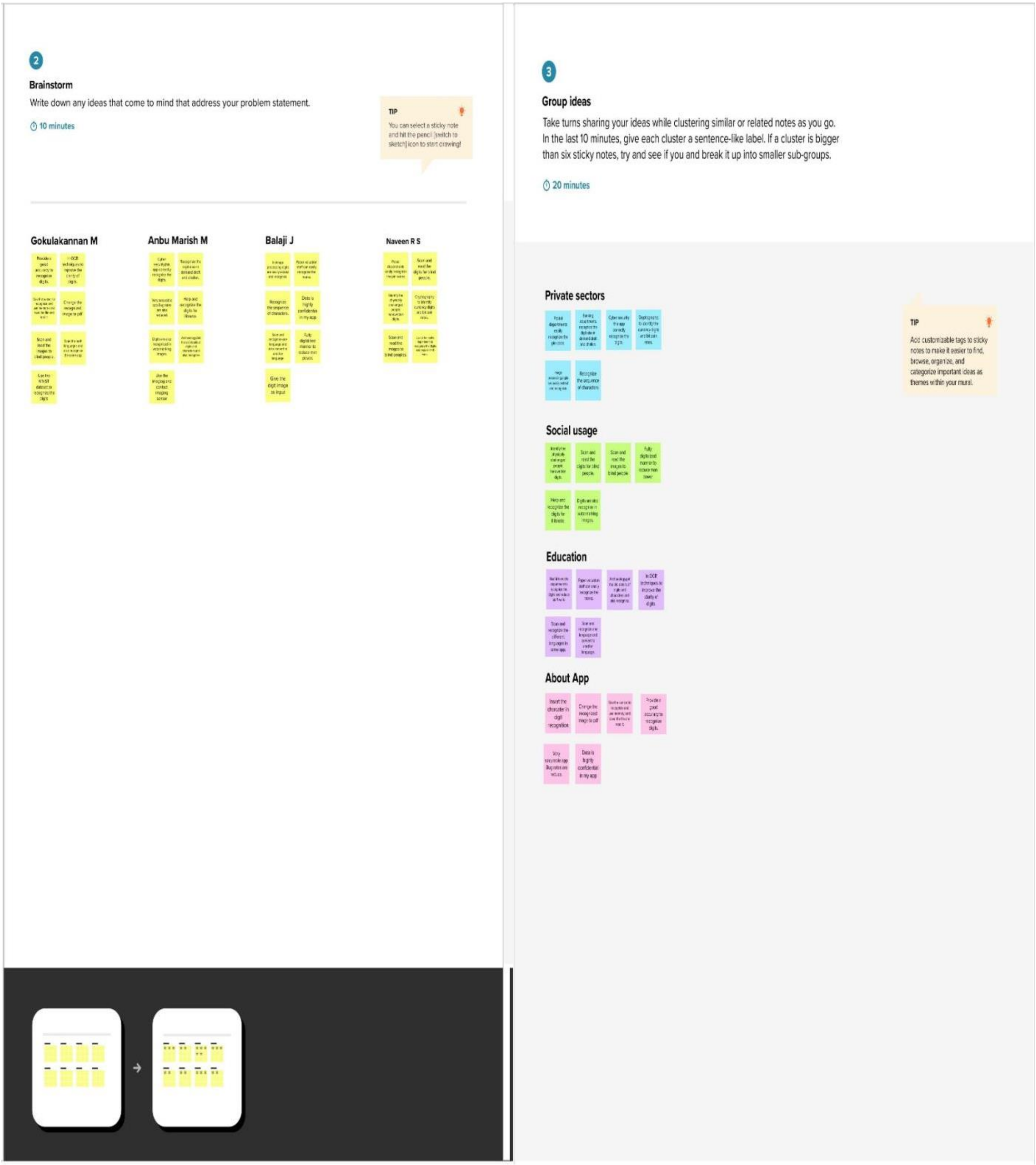
5 minutes

PROBLEM STATEMENT:
Now a days people can't recognize the handwritten digits in the various commercial services like banking, postal, education and many government departments etc. It is hard to find the exact digits given by the user. Therefore we introduce our project "Handwritten digit recognition" which is digitally analyse the exact sequence of digit given by the user(Pincode).

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.

Step-2: Brainstorm, Idea Listing and Grouping



2

Brainstorm

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

10 minutes

TIP

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

Gokulakannan M

Provides a good accuracy to recognize digits.

In OCR techniques to improve the clarity of digits.

Change the recognized image to pdf

Scan and read the images to blind peoples.

Use the MNIST dataset to recognize the digits

Anbu Marish M

Cyber security the app correctly recognize the digits.

Help and recognize the digits for illiterate.

Digits are also recognized in watermarking images.

Use the imaging and contact imaging sensor

Balaji J

Recognize the sequence of characters.

Data is highly confidential in my app.

Fully digital based manner to reduce man power.

Give the digit image as input

Naveen R S

Scan and read the digits for blind people.

Scan and read the images to blind peoples.

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.

Private sectors

Cyber security the app correctly recognize the digits.

Recognize the sequence of characters

Social usage

Scan and read the digits for blind people.

Scan and read the images to blind people.

Fully digits based manner to reduce man power

Help and recognize the digits for illiterate.

Digits are also recognize in watermarking images.

Education

In OCR techniques to improve the clarity of digits.

Scan and recognize the different languages in same app.

About App

Insert the character in digit recognition

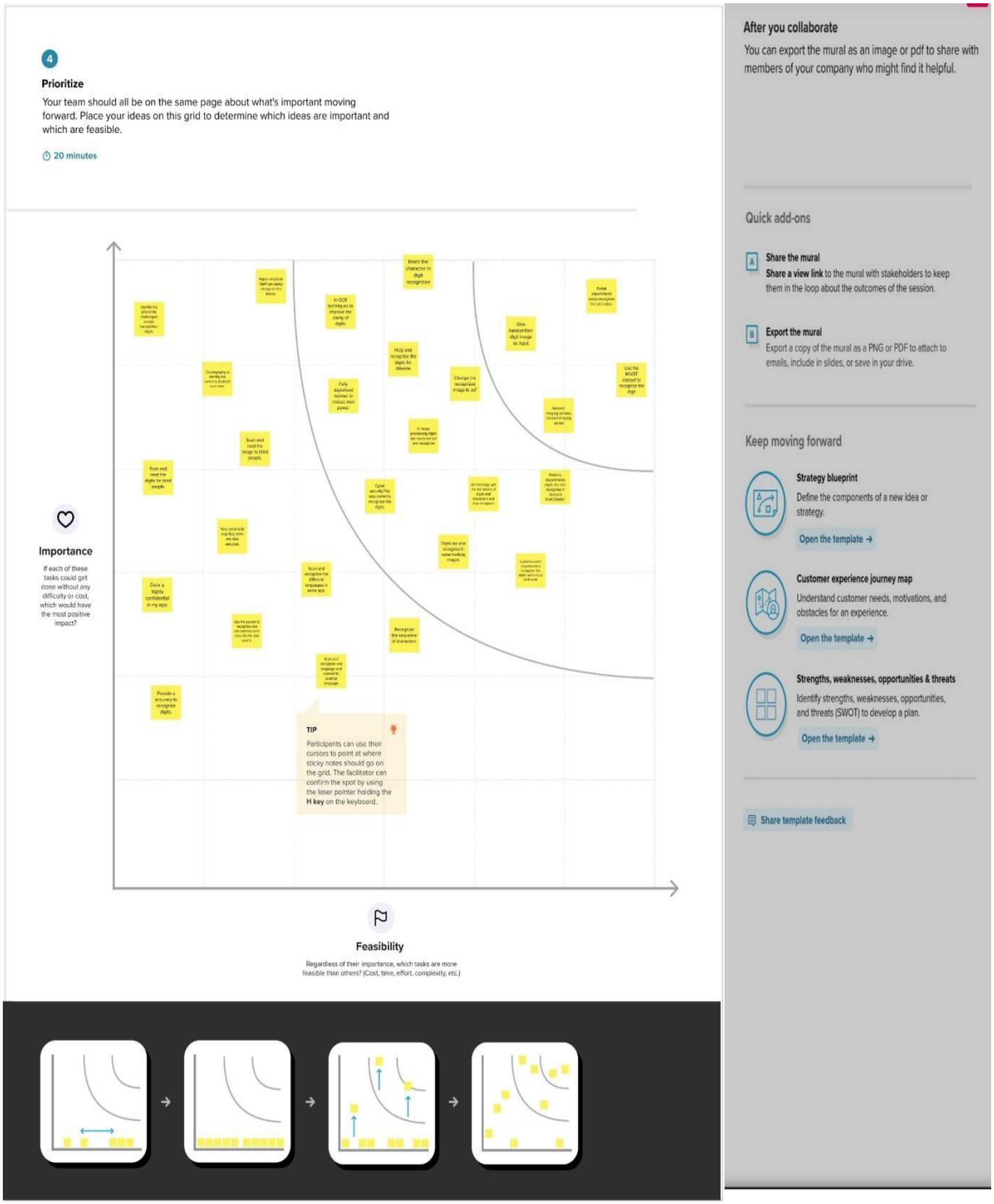
Change the recognized image to pdf

Provide a good accuracy to recognize digits.

Very accurate app Bug rates are reduce.

Data is highly confidential in my app

Step-3: Idea Prioritization



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?

Feasibility

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

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.

After you collaborate

You can export the mural as an image or pdf to share with members of your company who might find it helpful.

Quick add-ons

A **Share the mural**
Share a view link to the mural with stakeholders to keep them in the loop about the outcomes of the session.

B **Export the mural**
Export a copy of the mural as a PNG or PDF to attach to emails, include in slides, or save in your drive.

Keep moving forward

Strategy blueprint
Define the components of a new idea or strategy.
Open the template →

Customer experience journey map
Understand customer needs, motivations, and obstacles for an experience.
Open the template →

Strengths, weaknesses, opportunities & threats
Identify strengths, weaknesses, opportunities, and threats (SWOT) to develop a plan.
Open the template →

Share template feedback

Insert the character in digit recognition
In OCR techniques to improve the clarity of digits.
Help and recognize the digits for illiterate
Fully digitalized manner to reduce man power.
Scan and read the image to blind people.
Scan and read the digits for blind people.
Give handwritten digit image as input
Change the recognized image to pdf
Use the MNIST dataset to recognize the digit
Cyber security/This app correctly recognize the digits.
Digits are also recognize in watermarking images
Data is highly confidential in my app
Scan and recognize the different languages in same app.
Recognize the sequence of characters.
Provide a accuracy to recognize digits.

3.3 Proposed Solution

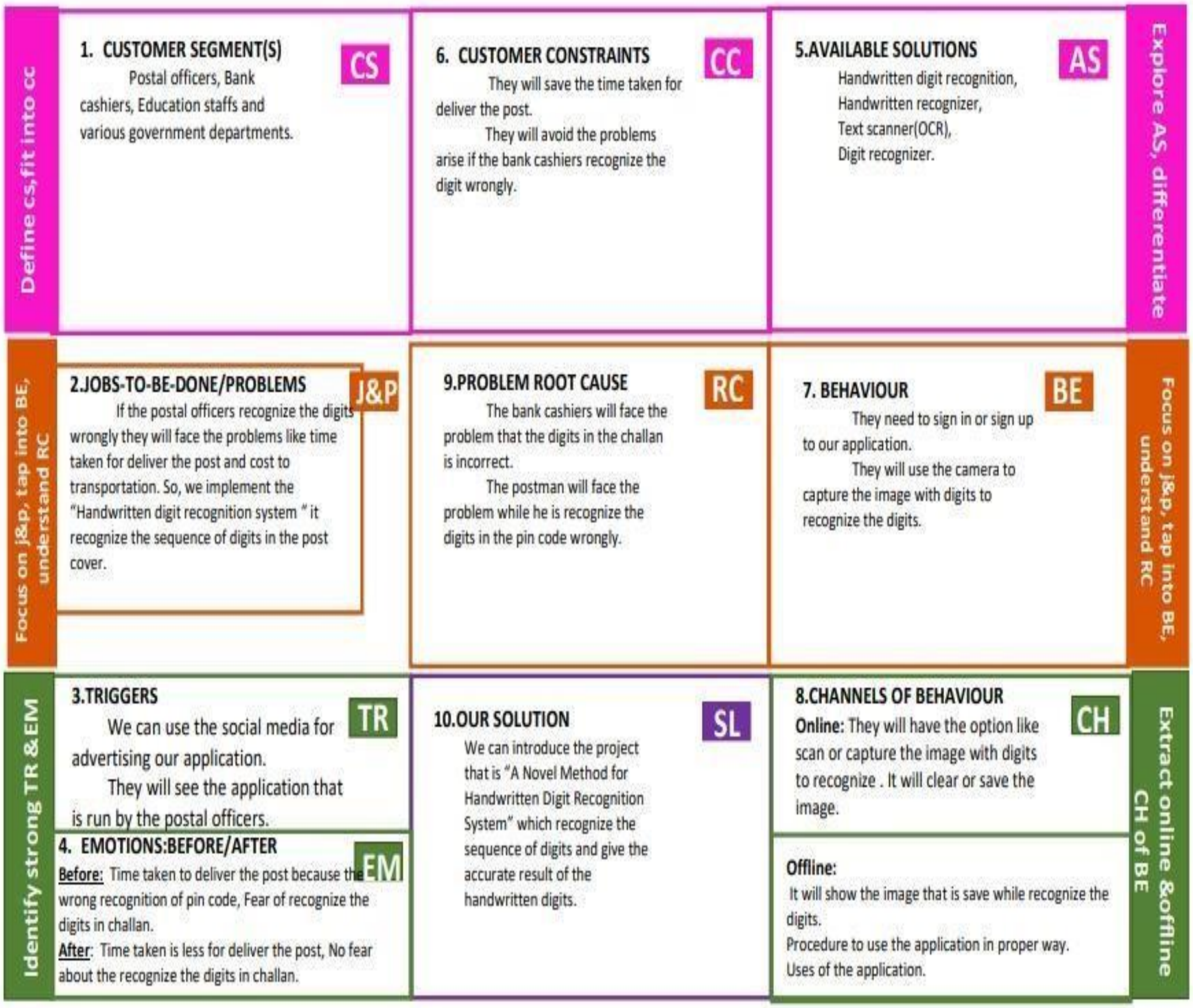
S.No.	Parameter	Description
1.	Problem Statement (Problem to besolved)	Now a days people can’t recognize the handwritten digits in the various commercial services like banking, postal, education, and many government departments etc. It is hardto find the exact digits given by the user. Therefore, we introduce our project “Handwritten digit recognition” which is digitally analyze the exact sequence of digit given by the user (Pin code).
2.	Idea / Solution description	Identify the exact digits which is in the sequence form.
3.	Novelty / Uniqueness	Identify the sequence of digits.
4.	Social Impact / Customer Satisfaction	1.) To identify the pin code inpostal Service. 2.) To identify the digits in challan, it avoids the fake results.
5.	Business Model (Revenue Model)	1.) Include ads in the application. 2.) We can use the upgrade versionto earn money. 3.) We can sell our project to the customers.
6.	Scalability of the Solution	We can add the features like 1.) recognize the characters with thedigits. 2.) We can add the multiple languages to recognize the characters.

3.4 Problem Solution fit

The Problem-Solution Fit simply means that you have found a problem with your customer and thatthe solution you have realized for it actually solves the customer’s problem. It helps entrepreneurs, marketers and corporate innovators identify behavioral patterns and recognize what would work and why

PURPOSE:

- ❑ Solve complex problems in a way that fits the state of your customers.
- ❑ Succeed faster and increase your solution adoption by tapping into existing mediums andchannels of behavior.
- ❑ Sharpen your communication and marketing strategy with the right triggers and messaging.
- ❑ Increase touch-points with your company by finding the right problem-behavior fit andbuilding trust by solving frequent annoyances, or urgent or costly problems.
- ❑ Understand the existing situation in order to improve it for your target group.



Define cs,fit into cc

1. CUSTOMER SEGMENT(S) CS

Postal officers, Bank cashiers, Education staffs and various government departments.

6. CUSTOMER CONSTRAINTS CC

They will save the time taken for deliver the post.

They will avoid the problems arise if the bank cashiers recognize the digit wrongly.

5.AVAILABLE SOLUTIONS AS

Handwritten digit recognition,
Handwritten recognizer,
Text scanner(OCR),
Digit recognizer.

Explore AS, differentiate

Focus on j&p, tap into BE, understand RC

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

If the postal officers recognize the digits wrongly they will face the problems like time taken for deliver the post and cost to transportation. So, we implement the “Handwritten digit recognition system ” it recognize the sequence of digits in the post cover.

9.PROBLEM ROOT CAUSE RC

The bank cashiers will face the problem that the digits in the challan is incorrect.

The postman will face the problem while he is recognize the digits in the pin code wrongly.

7. BEHAVIOUR BE

They need to sign in or sign up to our application.

They will use the camera to capture the image with digits to recognize the digits.

Focus on j&p, tap into BE, understand RC

Identify strong TR &EM

3.TRIGGERS TR

We can use the social media for advertising our application.

They will see the application that is run by the postal officers.

4. EMOTIONS:BEFORE/AFTER EM

Before: Time taken to deliver the post because the wrong recognition of pin code, Fear of recognize the digits in challan.

After: Time taken is less for deliver the post, No fear about the recognize the digits in challan.

10.OUR SOLUTION SL

We can introduce the project that is “A Novel Method for Handwritten Digit Recognition System” which recognize the sequence of digits and give the accurate result of the handwritten digits.

8.CHANNELS OF BEHAVIOUR CH

Online: They will have the option like scan or capture the image with digits to recognize . It will clear or save the image.

Offline:

It will show the image that is save while recognize the digits.

Procedure to use the application in proper way.

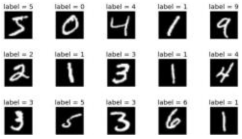
Uses of the application.

Extract online &offline CH of BE

4. REQUIREMENT ANALYSIS

4.1 Functional requirements

FR No.	Functional Requirement and description:
FR-1	**Image data**: Handwritten digit recognition is the ability of a computer to recognizethe human handwritten digits from different sources like images, papers, touch screens, etc and classify them into 10 predefined classes (0-9). This has been atopic of boundless-research in the field of deep learning.
FR-2	**Digit _Classifier_ Model**: Use the MNIST database of handwritten digits to train a convolutional network to predict the digit given an image. First obtain the training and validation model.

FR-3	**MNIST dataset:** The MNIST dataset is an acronym that stands for the modified National Institute of Standards and Technology dataset. It is a dataset of 60,000 small square 28 *28-pixel greyscale images of handwritten single digits between 0and to 9.
	
	 label = 5
	label = 0
	label = 4
	label = 1
	label = 9
	label = 2
	label = 1
	label = 3
	label = 1
	label = 4
	label = 3
	label = 5
	label = 3
	label = 6
	label = 1

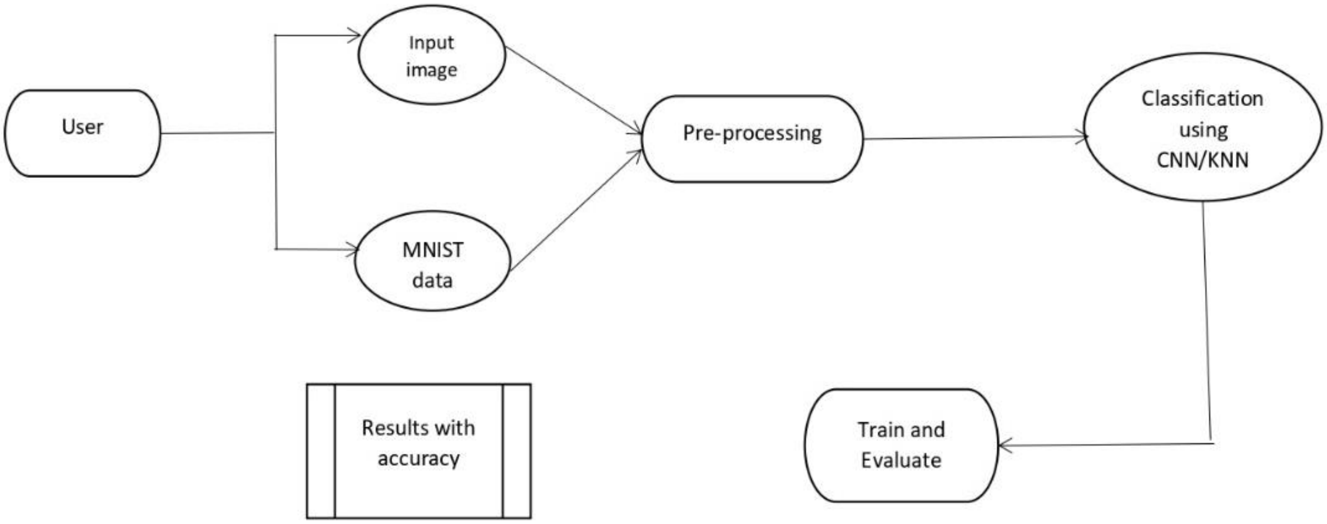
4.2 Non-Functional requirements

FR No.	Non-Functional Requirement
NFR-1	**Usability:** Handwritten digit recognition is one of the practically important issues inpattern recognition applications. The applications of digit recognition include in postal mail sorting, bank check processing, form data entry and etc.
NFR-2	**Reliability:** 1) The system not only produces a classification of the digit but also a richdescription of the instantiation parameters. 2) The generative models can perform recognition driven segmentation 3) The method invokes a relatively.
NFR-3	**Performance:** The neural network uses the examples to automatically infer rules for recognizing handwritten digits. Furthermore, by increasing the number of training examples, the network can learn more about handwriting, and so improve its accuracy. There are a number of ways and algorithms to recognize handwrittendigits, including Deep Learning/CNN, SVM, Gaussian Naive Bayes, KNN, Decision Trees, Random Forests etc.
NFR-4	**Accuracy:** Optical Character Recognition (OCR) technology provides higher than 99% accuracy with typed characters in high-quality images. However, the diversity in human writing types, spacing differences and irregularities of handwriting causes less accurate character recognition.

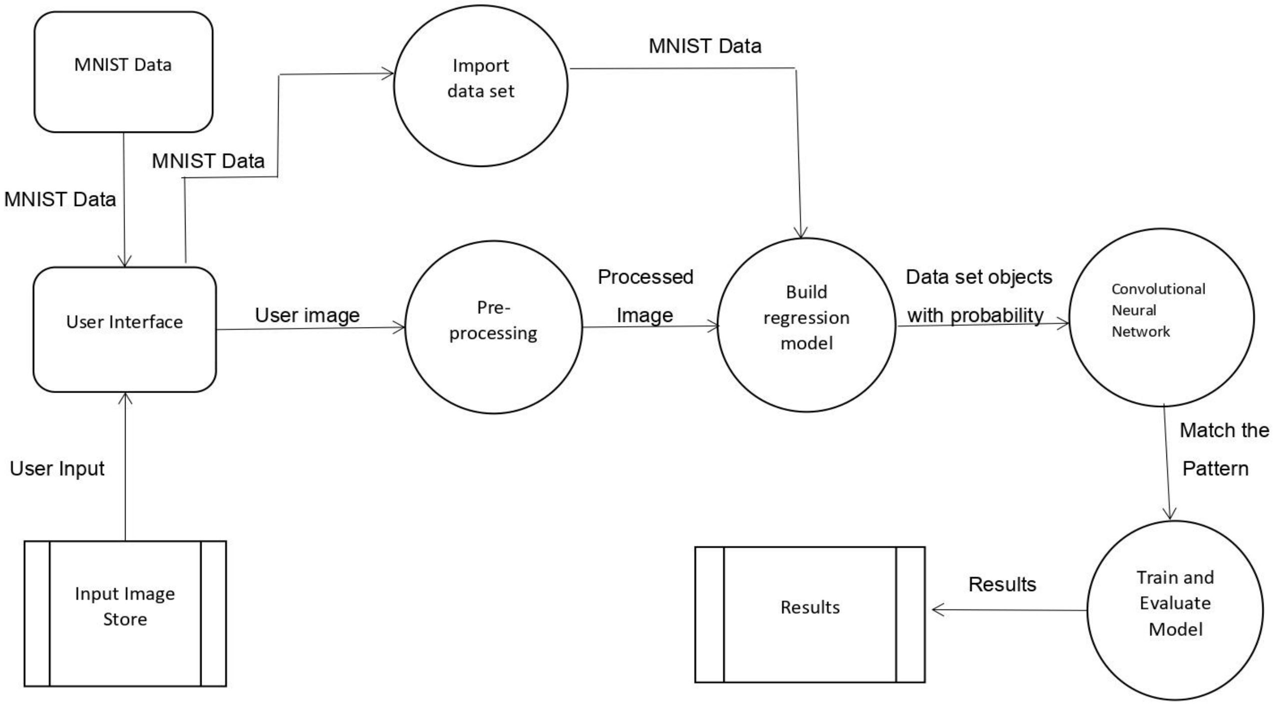
5. PROJECT DESIGN

5.1 Data Flow Diagrams

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



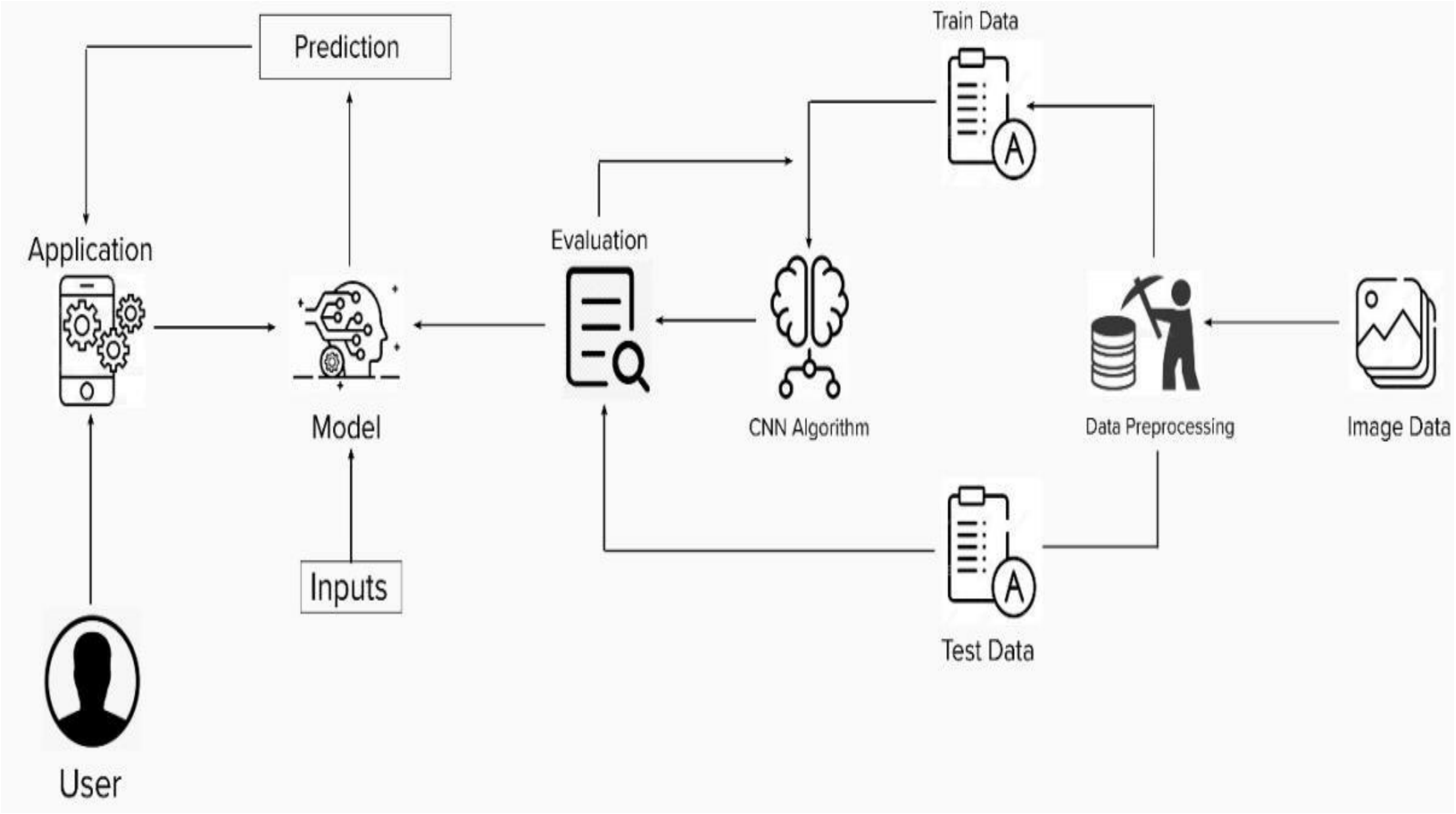
Example: DFD Level 0 (Industry Standard



5.2 Solution & Technical Architecture

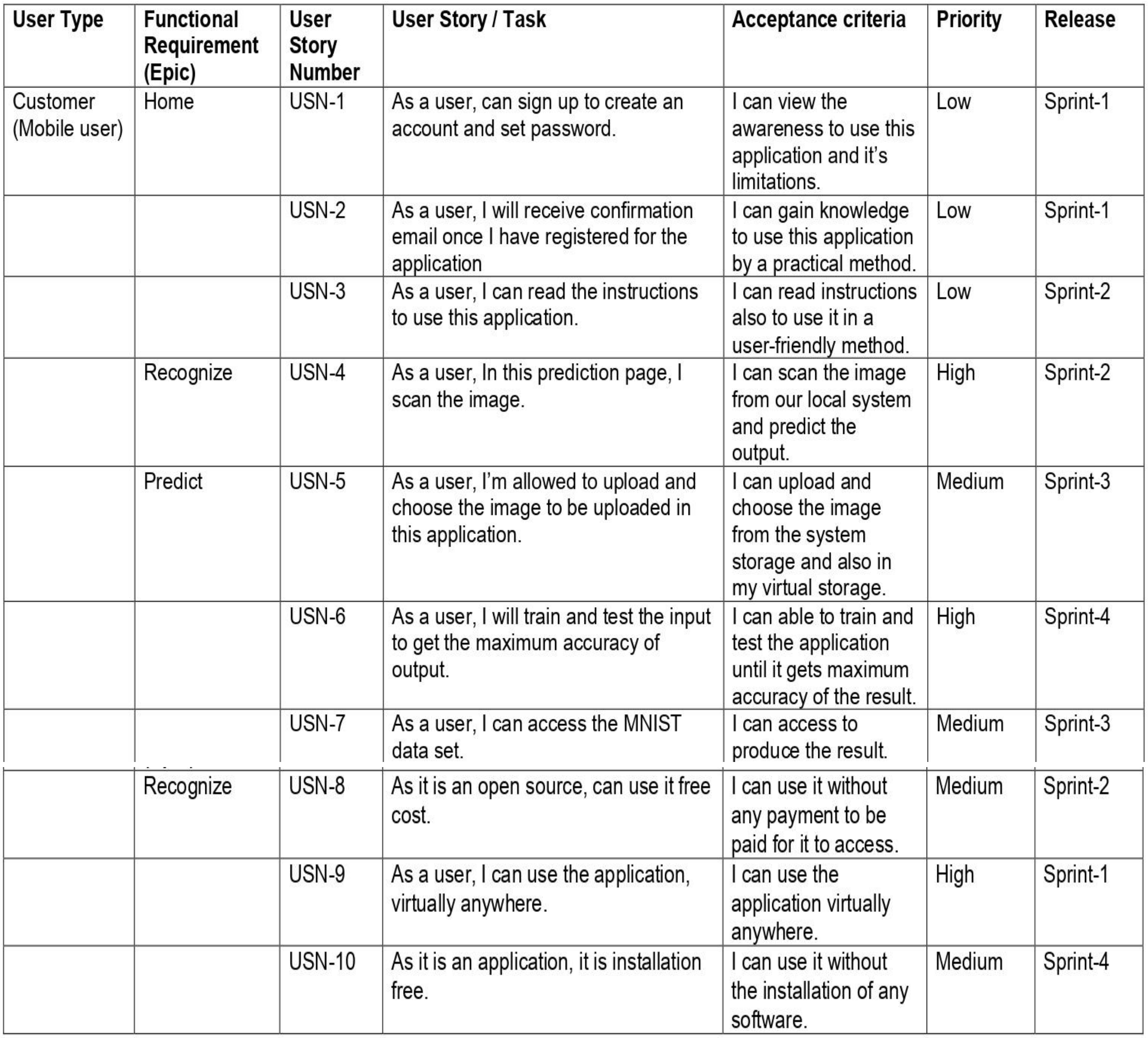
Solution architecture is a complex process – with many sub-processes – that bridgesthe gap between business problems and technology solutions. Its goals are to:

- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.
- Define features, development phases, and solution requirements.
- Provide specifications according to which the solution is defined, managed,and delivered.



Reference:
https://app.mural.co/invitation/mural/ibm80676/1665134485750?sender=uae023cbd0fd8b1d992506520&key=041057db-8752-4aad-b22a-47f057bd6a99

5.3 User Stories



User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Home	USN-1	As a user, can sign up to create an account and set password.	I can view the awareness to use this application and it's limitations.	Low	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can gain knowledge to use this application by a practical method.	Low	Sprint-1
		USN-3	As a user, I can read the instructions to use this application.	I can read instructions also to use it in a user-friendly method.	Low	Sprint-2
	Recognize	USN-4	As a user, In this prediction page, I scan the image.	I can scan the image from our local system and predict the output.	High	Sprint-2
	Predict	USN-5	As a user, I'm allowed to upload and choose the image to be uploaded in this application.	I can upload and choose the image from the system storage and also in my virtual storage.	Medium	Sprint-3
		USN-6	As a user, I will train and test the input to get the maximum accuracy of output.	I can able to train and test the application until it gets maximum accuracy of the result.	High	Sprint-4
		USN-7	As a user, I can access the MNIST data set.	I can access to produce the result.	Medium	Sprint-3
	Recognize	USN-8	As it is an open source, can use it free cost.	I can use it without any payment to be paid for it to access.	Medium	Sprint-2
		USN-9	As a user, I can use the application, virtually anywhere.	I can use the application virtually anywhere.	High	Sprint-1
		USN-10	As it is an application, it is installation free.	I can use it without the installation of any software.	Medium	Sprint-4

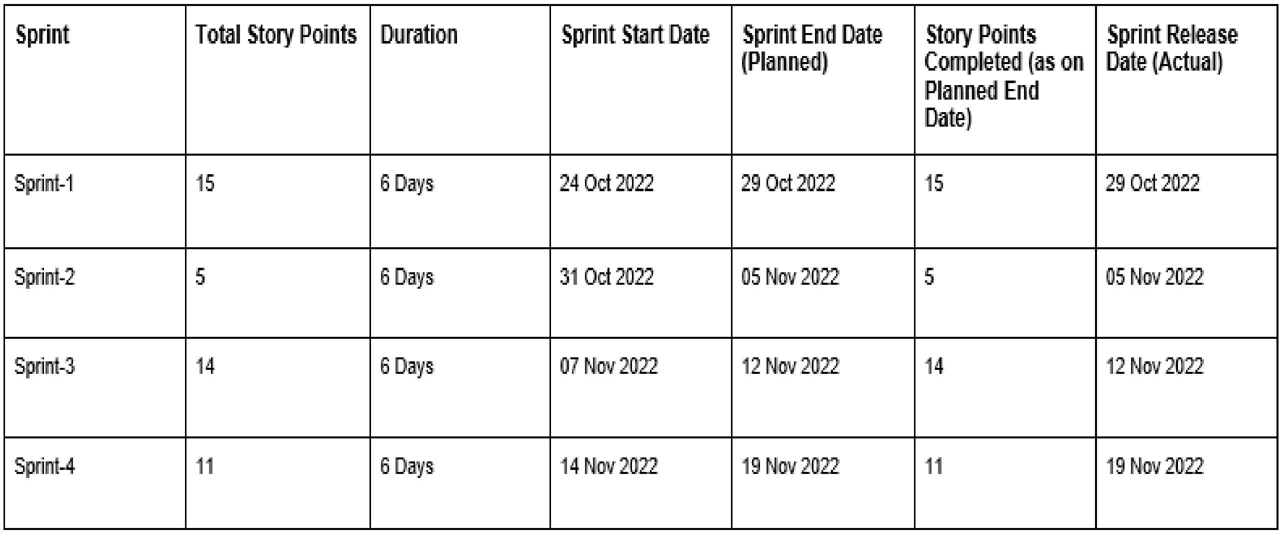
6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming mypassword.	2	High	Gokulakannan M
Sprint-1	confirmation	USN-2	As a user, I will receive confirmation email once Ihave registered for the application	1	High	Anbu Marish.M
Sprint-1	Login	USN-3	As a user, I can log into the application by enteringemail & password	1	High	NAVEEN.R.S
Sprint-2	Data Collection	USN-4	As a user, I can collect the dataset fromvarious resources with different handwritings.	1	Low	Balaji J

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-2	Data Preprocessing	USN-5	As a user, I can load the dataset, handle the missing data, scaling and split data into train and test.	3	Medium	Gokulakannan M
Sprint-2	Model Building	USN-6	As a user, I will get an application with CNN model which provides high accuracy of recognized handwritten digit.	1	High	Anbu Marish.M
Sprint-3	Add CNN layers	USN-7	Creating the model and adding the input, hidden, and output layers to it.	5	High	Naveen.R.S
Sprint-3	Compiling the model	USN-8	With both the training data defined and model defined, it's time to configure the learning process.	2	Medium	Balaji J
Sprint-3	Train & test the model	USN-9	As a user, let us train our model with our image dataset.	5	Medium	Gokulakannan M
Sprint-3	Save the model	USN-10	As a user, the model is saved & integrated with an android application in order to predict something.	2	Low	Anbu Marish.M
Sprint-4	Train the model on IBM	USN-11	As a user, I train the model on IBM and integrate flask/Django with scoring end points.	3	High	Naveen.R.S
Sprint-4	Cloud Deployment	USN-12	As a user, I can access the web application and make use of the product from anywhere.	8	High	Balaji J

6.2 Sprint Delivery Schedule

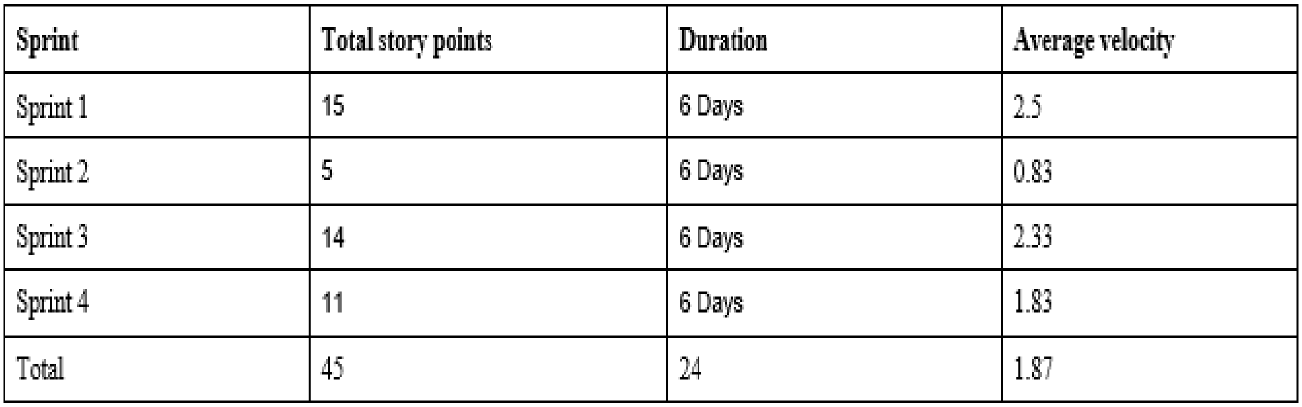


Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	15	6 Days	24 Oct 2022	29 Oct 2022	15	29 Oct 2022
Sprint-2	5	6 Days	31 Oct 2022	05 Nov 2022	5	05 Nov 2022
Sprint-3	14	6 Days	07 Nov 2022	12 Nov 2022	14	12 Nov 2022
Sprint-4	11	6 Days	14 Nov 2022	19 Nov 2022	11	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)

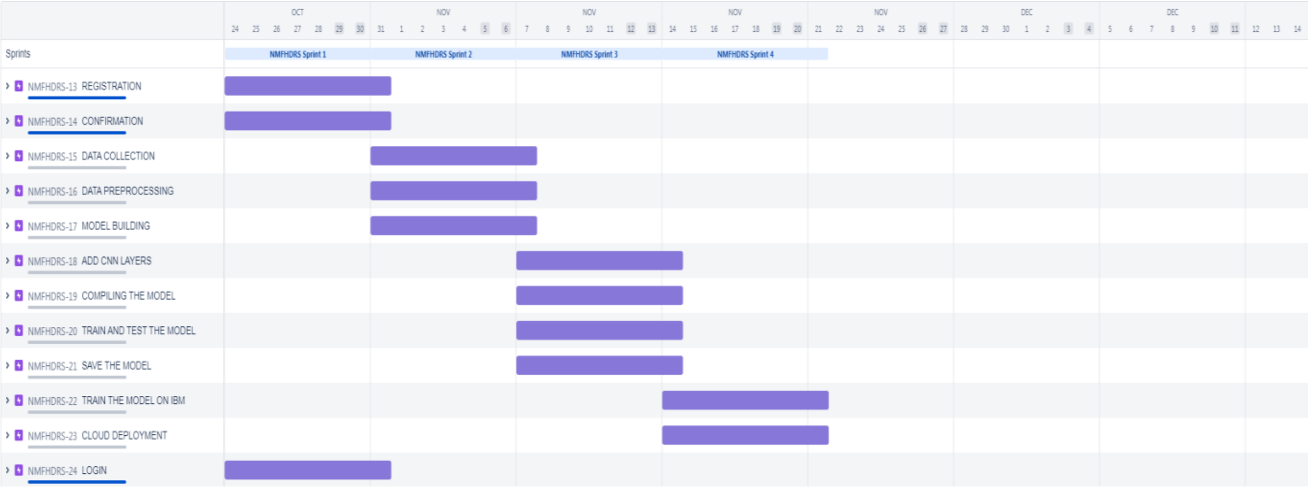
$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$



Sprint	Total story points	Duration	Average velocity
Sprint 1	15	6 Days	2.5
Sprint 2	5	6 Days	0.83
Sprint 3	14	6 Days	2.33
Sprint 4	11	6 Days	1.83
Total	45	24	1.87

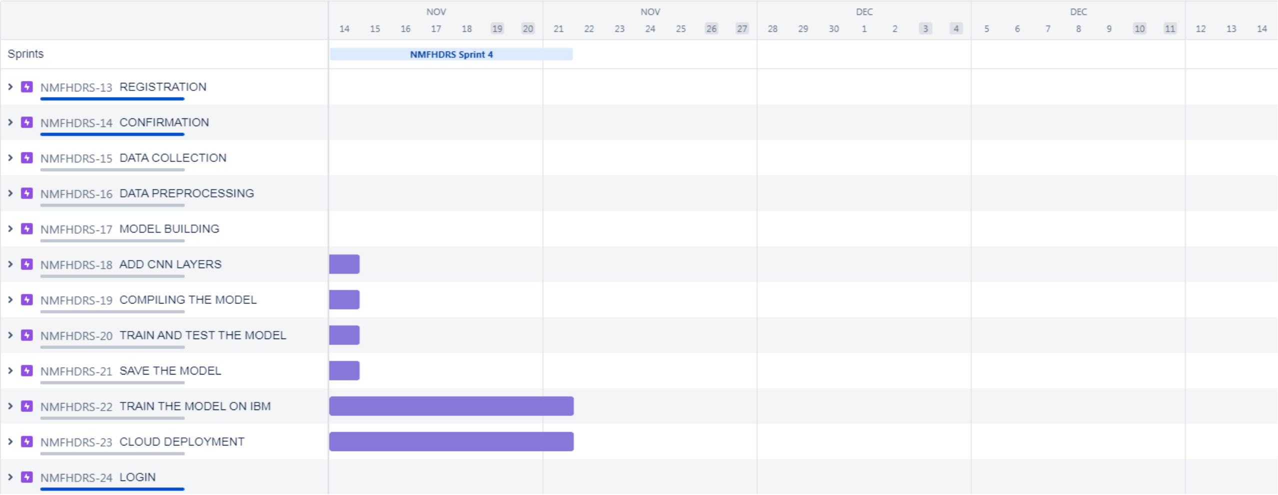
Burndown Chart:

A burndown 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, burndown charts can be applied to any project containing measurable progress over time.



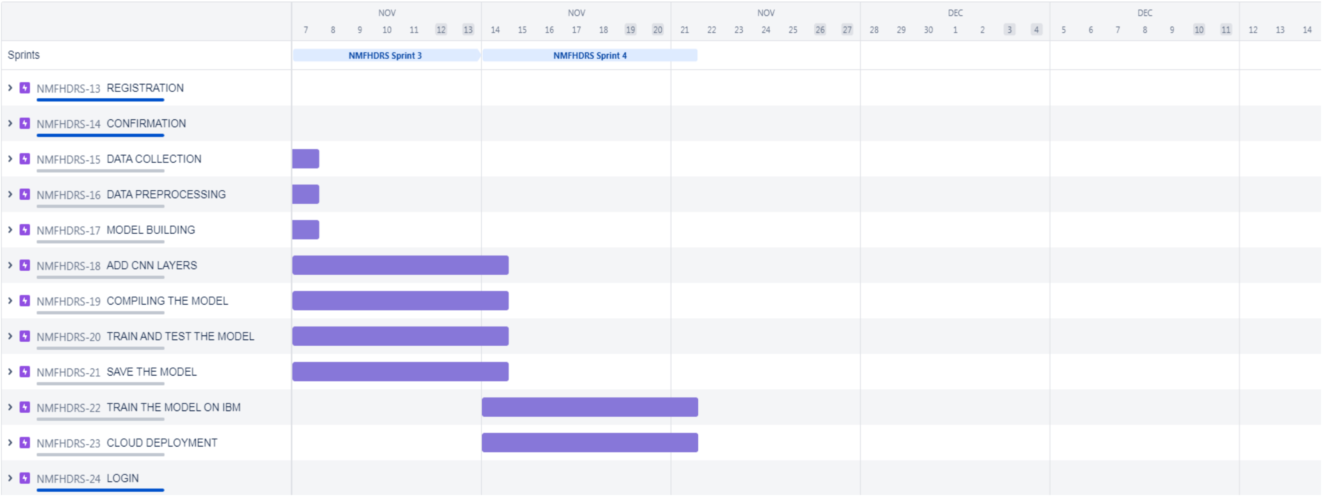
Sprints

- NMFHDRS-13 REGISTRATION
- NMFHDRS-14 CONFIRMATION
- NMFHDRS-15 DATA COLLECTION
- NMFHDRS-16 DATA PREPROCESSING
- NMFHDRS-17 MODEL BUILDING
- NMFHDRS-18 ADD CNN LAYERS
- NMFHDRS-19 COMPILING THE MODEL
- NMFHDRS-20 TRAIN AND TEST THE MODEL
- NMFHDRS-21 SAVE THE MODEL
- NMFHDRS-22 TRAIN THE MODEL ON IBM
- NMFHDRS-23 CLOUD DEPLOYMENT
- NMFHDRS-24 LOGIN



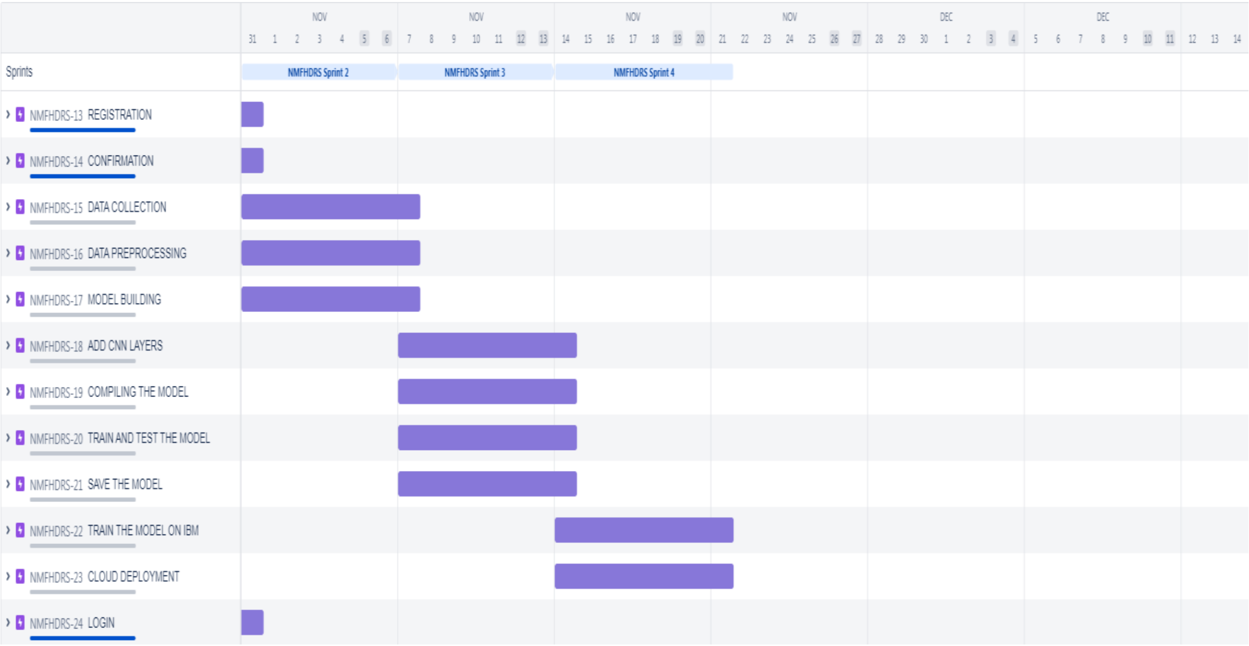
Sprints

- NMFHDRS-13 REGISTRATION
- NMFHDRS-14 CONFIRMATION
- NMFHDRS-15 DATA COLLECTION
- NMFHDRS-16 DATA PREPROCESSING
- NMFHDRS-17 MODEL BUILDING
- NMFHDRS-18 ADD CNN LAYERS
- NMFHDRS-19 COMPILING THE MODEL
- NMFHDRS-20 TRAIN AND TEST THE MODEL
- NMFHDRS-21 SAVE THE MODEL
- NMFHDRS-22 TRAIN THE MODEL ON IBM
- NMFHDRS-23 CLOUD DEPLOYMENT
- NMFHDRS-24 LOGIN



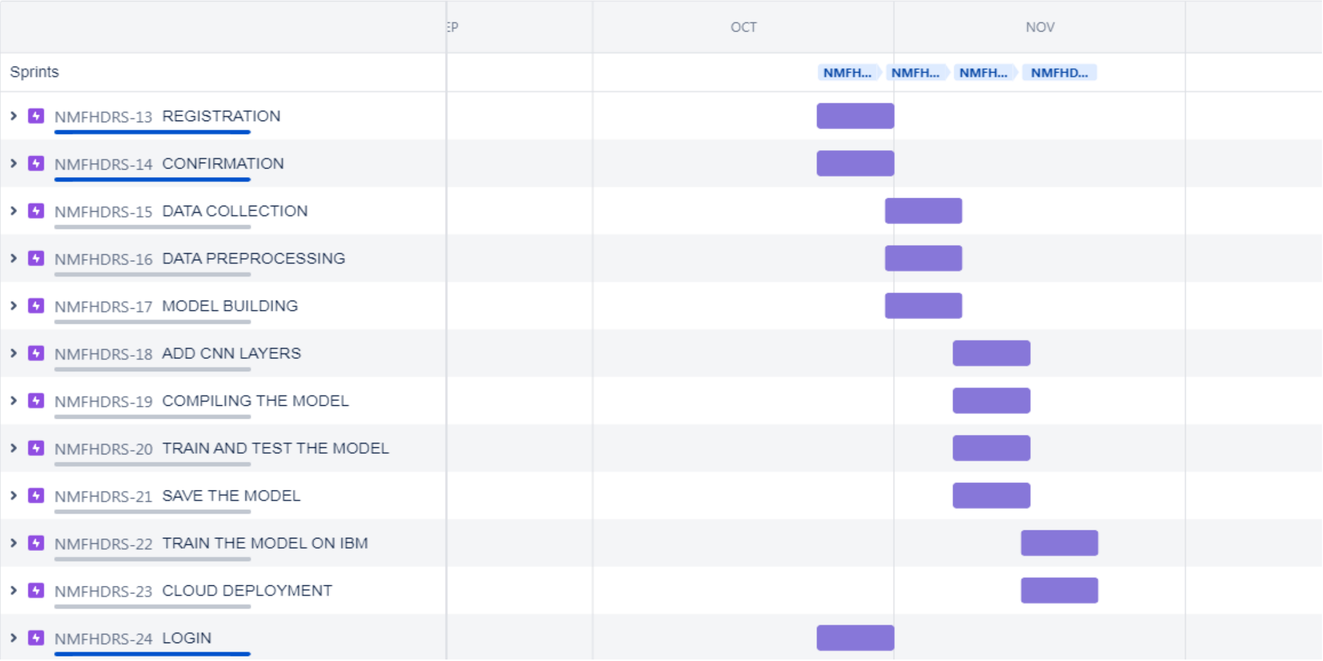
Sprints
NMFHDRS Sprint 3
NMFHDRS Sprint 4

NMFHDRS-13 REGISTRATION
NMFHDRS-14 CONFIRMATION
NMFHDRS-15 DATA COLLECTION
NMFHDRS-16 DATA PREPROCESSING
NMFHDRS-17 MODEL BUILDING
NMFHDRS-18 ADD CNN LAYERS
NMFHDRS-19 COMPILING THE MODEL
NMFHDRS-20 TRAIN AND TEST THE MODEL
NMFHDRS-21 SAVE THE MODEL
NMFHDRS-22 TRAIN THE MODEL ON IBM
NMFHDRS-23 CLOUD DEPLOYMENT
NMFHDRS-24 LOGIN



Sprints
NMFHDRS Sprint 2
NMFHDRS Sprint 3
NMFHDRS Sprint 4

NMFHDRS-13 REGISTRATION
NMFHDRS-14 CONFIRMATION
NMFHDRS-15 DATA COLLECTION
NMFHDRS-16 DATA PREPROCESSING
NMFHDRS-17 MODEL BUILDING
NMFHDRS-18 ADD CNN LAYERS
NMFHDRS-19 COMPILING THE MODEL
NMFHDRS-20 TRAIN AND TEST THE MODEL
NMFHDRS-21 SAVE THE MODEL
NMFHDRS-22 TRAIN THE MODEL ON IBM
NMFHDRS-23 CLOUD DEPLOYMENT
NMFHDRS-24 LOGIN



EP OCT NOV

Sprints NMFH... NMFH... NMFH... NMFHD...

NMFHDRS-13 REGISTRATION

NMFHDRS-14 CONFIRMATION

NMFHDRS-15 DATA COLLECTION

NMFHDRS-16 DATA PREPROCESSING

NMFHDRS-17 MODEL BUILDING

NMFHDRS-18 ADD CNN LAYERS

NMFHDRS-19 COMPILING THE MODEL

NMFHDRS-20 TRAIN AND TEST THE MODEL

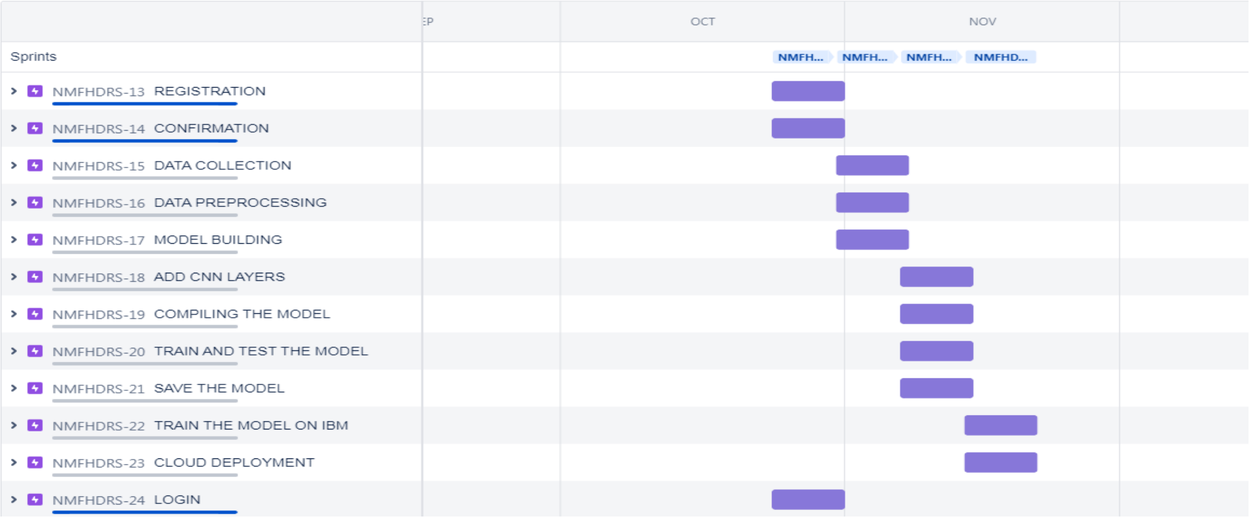
NMFHDRS-21 SAVE THE MODEL

NMFHDRS-22 TRAIN THE MODEL ON IBM

NMFHDRS-23 CLOUD DEPLOYMENT

NMFHDRS-24 LOGIN

6.3 Reports from JIRA



EP OCT NOV

Sprints NMFH... NMFH... NMFH... NMFHD...

NMFHDRS-13 REGISTRATION

NMFHDRS-14 CONFIRMATION

NMFHDRS-15 DATA COLLECTION

NMFHDRS-16 DATA PREPROCESSING

NMFHDRS-17 MODEL BUILDING

NMFHDRS-18 ADD CNN LAYERS

NMFHDRS-19 COMPILING THE MODEL

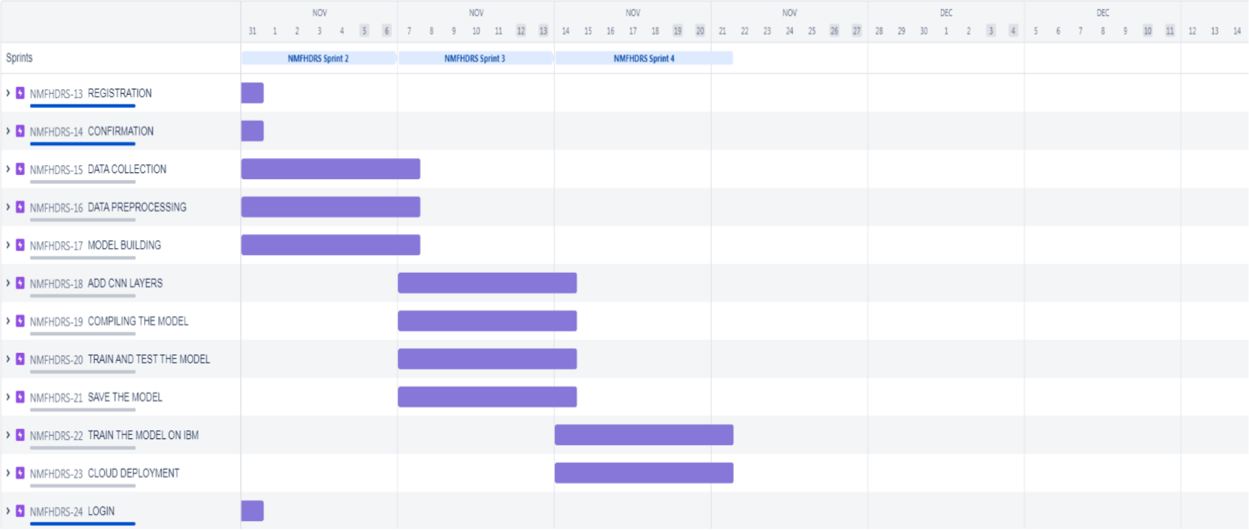
NMFHDRS-20 TRAIN AND TEST THE MODEL

NMFHDRS-21 SAVE THE MODEL

NMFHDRS-22 TRAIN THE MODEL ON IBM

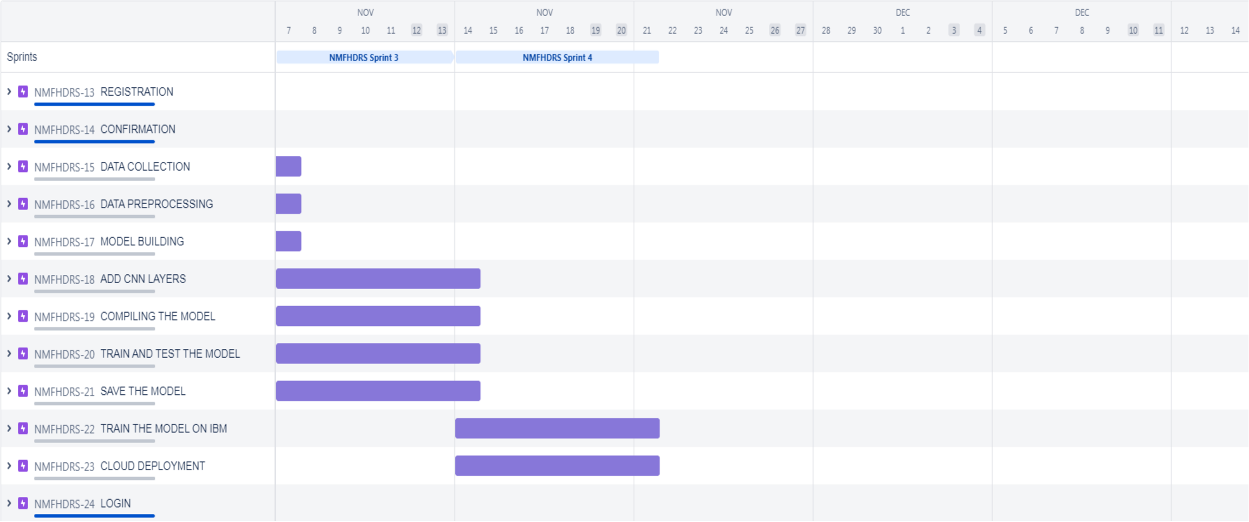
NMFHDRS-23 CLOUD DEPLOYMENT

NMFHDRS-24 LOGIN



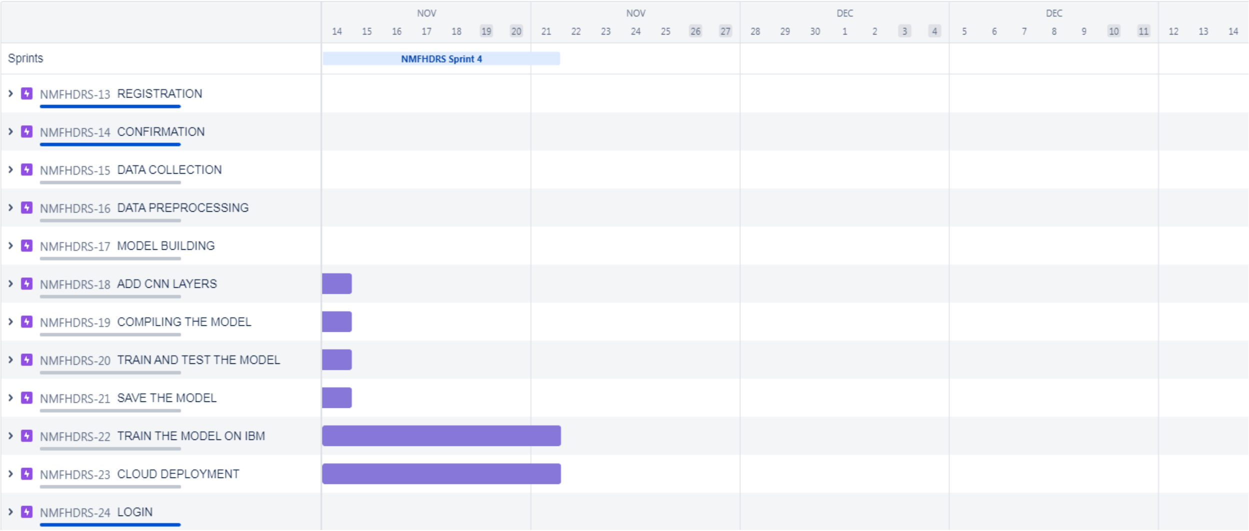
Sprints

- NMFHDRS-13 REGISTRATION
- NMFHDRS-14 CONFIRMATION
- NMFHDRS-15 DATA COLLECTION
- NMFHDRS-16 DATA PREPROCESSING
- NMFHDRS-17 MODEL BUILDING
- NMFHDRS-18 ADD CNN LAYERS
- NMFHDRS-19 COMPILING THE MODEL
- NMFHDRS-20 TRAIN AND TEST THE MODEL
- NMFHDRS-21 SAVE THE MODEL
- NMFHDRS-22 TRAIN THE MODEL ON IBM
- NMFHDRS-23 CLOUD DEPLOYMENT
- NMFHDRS-24 LOGIN



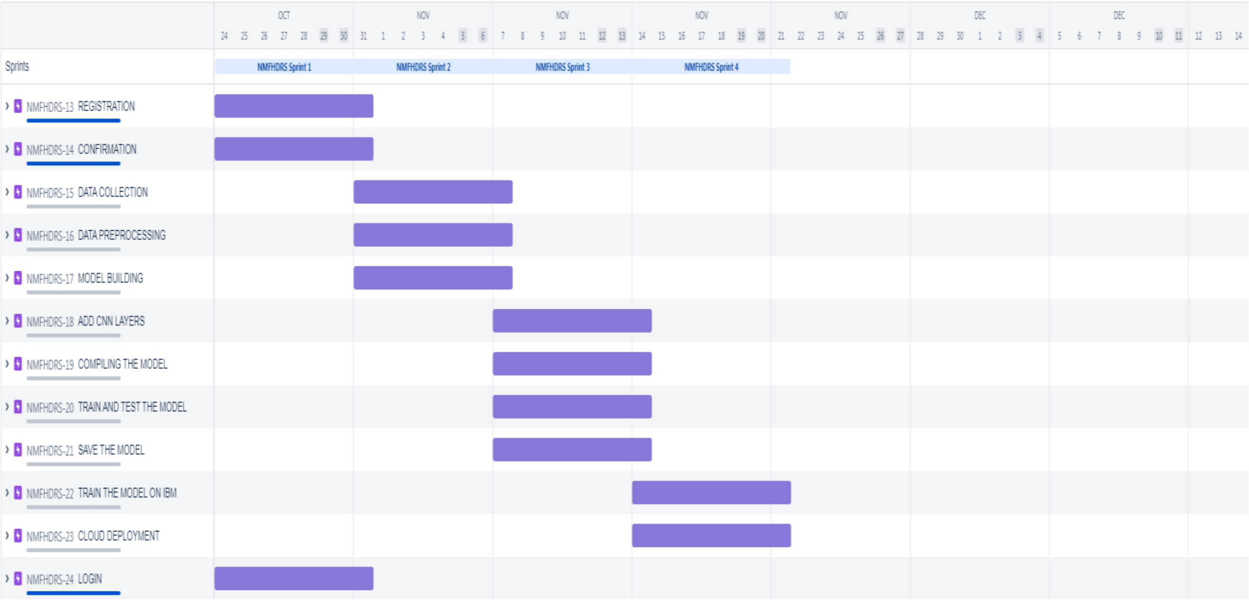
Sprints

- NMFHDRS-13 REGISTRATION
- NMFHDRS-14 CONFIRMATION
- NMFHDRS-15 DATA COLLECTION
- NMFHDRS-16 DATA PREPROCESSING
- NMFHDRS-17 MODEL BUILDING
- NMFHDRS-18 ADD CNN LAYERS
- NMFHDRS-19 COMPILING THE MODEL
- NMFHDRS-20 TRAIN AND TEST THE MODEL
- NMFHDRS-21 SAVE THE MODEL
- NMFHDRS-22 TRAIN THE MODEL ON IBM
- NMFHDRS-23 CLOUD DEPLOYMENT
- NMFHDRS-24 LOGIN



Sprints

- NMFHDRS-13 REGISTRATION
- NMFHDRS-14 CONFIRMATION
- NMFHDRS-15 DATA COLLECTION
- NMFHDRS-16 DATA PREPROCESSING
- NMFHDRS-17 MODEL BUILDING
- NMFHDRS-18 ADD CNN LAYERS
- NMFHDRS-19 COMPILING THE MODEL
- NMFHDRS-20 TRAIN AND TEST THE MODEL
- NMFHDRS-21 SAVE THE MODEL
- NMFHDRS-22 TRAIN THE MODEL ON IBM
- NMFHDRS-23 CLOUD DEPLOYMENT
- NMFHDRS-24 LOGIN

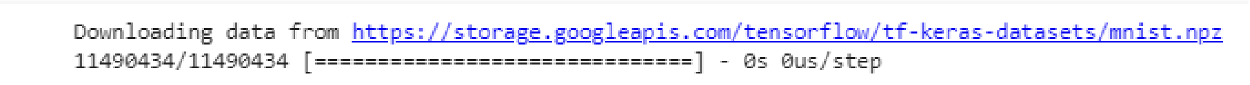


7. CODING & SOLUTIONING (Explain the features added in the project along with code)

7.1 Feature

Load the data

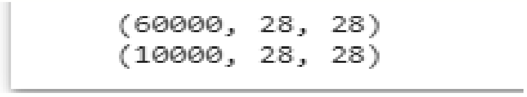
```
(X_train, y_train), (X_test, y_test) = mnist.load_data()
```



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

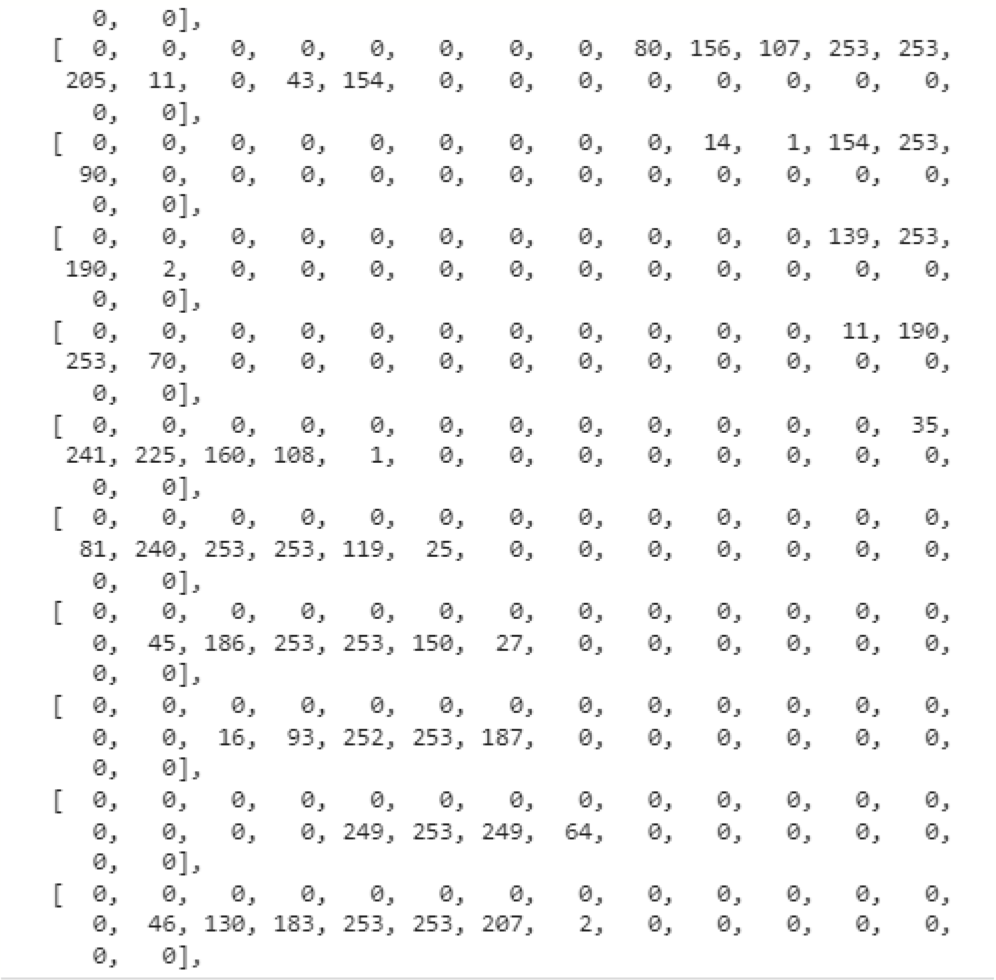
Analyzing the data

```
print(X_train.shape)

print(X_test.shape)
```



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

X_train[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,
         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],
```

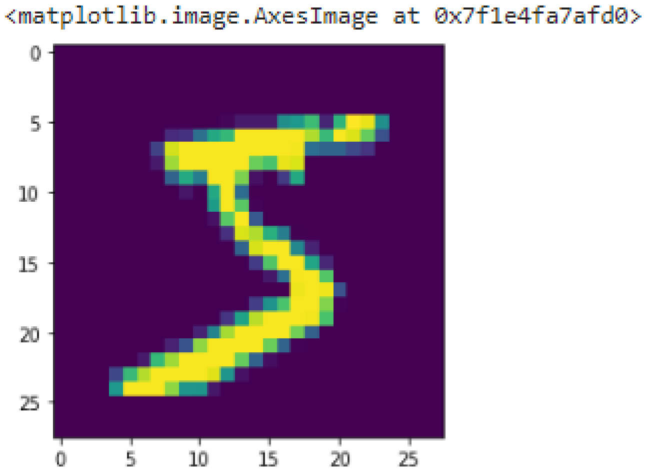
y_train[0]

<u>Output:</u>

5

```
plt.imshow(X_train[0])
```

Output:



```
<matplotlib.image.AxesImage at 0x7f1e4fa7afd0>
```

Process the data

```
    X_train = X_train.reshape(60000, 28, 28, 1).astype('float32')

X_test = X_test.reshape(10000, 28, 28, 1).astype('float32')

   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)
```

Output:

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

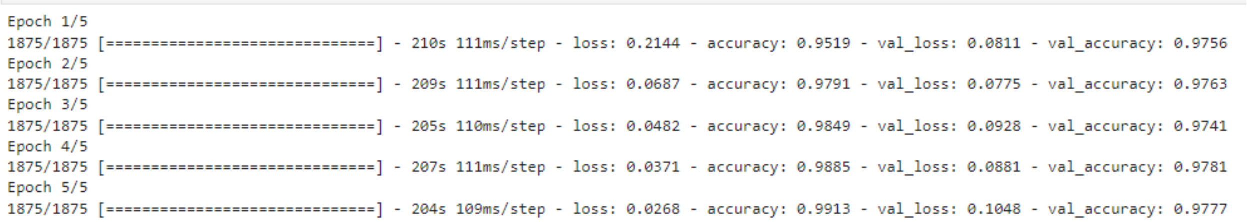
Creating the model

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

Training the model

```
model.fit(X_train, Y_train, batch_size=32, epochs=5,
validation_data=(X_test,Y_test))
```

output

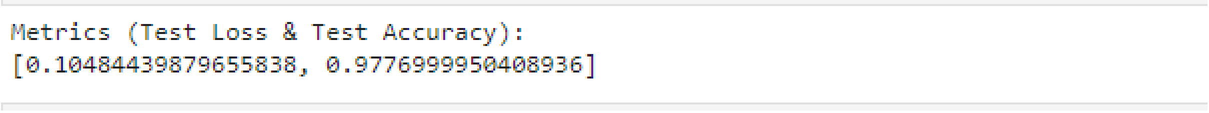


```
Epoch 1/5
1875/1875 [==============================] - 210s 111ms/step - loss: 0.2144 - accuracy: 0.9519 - val_loss: 0.0811 - val_accuracy: 0.9756
Epoch 2/5
1875/1875 [==============================] - 209s 111ms/step - loss: 0.0687 - accuracy: 0.9791 - val_loss: 0.0775 - val_accuracy: 0.9763
Epoch 3/5
1875/1875 [==============================] - 205s 110ms/step - loss: 0.0482 - accuracy: 0.9849 - val_loss: 0.0928 - val_accuracy: 0.9741
Epoch 4/5
1875/1875 [==============================] - 207s 111ms/step - loss: 0.0371 - accuracy: 0.9885 - val_loss: 0.0881 - val_accuracy: 0.9781
Epoch 5/5
1875/1875 [==============================] - 204s 109ms/step - loss: 0.0268 - accuracy: 0.9913 - val_loss: 0.1048 - val_accuracy: 0.9777
```

Testing the model

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

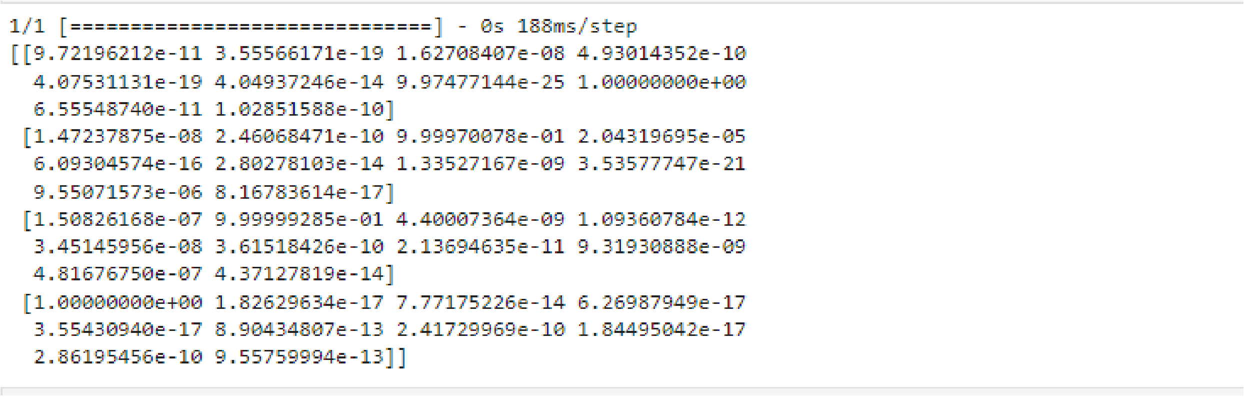
output



```
Metrics (Test Loss & Test Accuracy):
[0.10484439879655838, 0.9776999950408936]
```

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

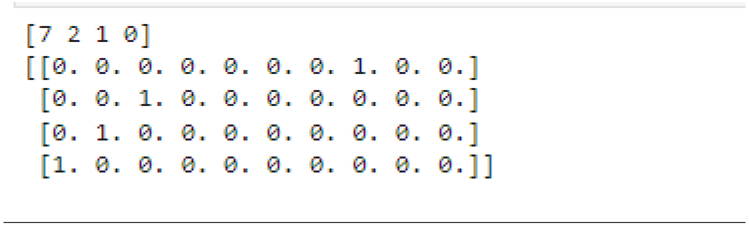
output



```
1/1 [==============================] - 0s 188ms/step
[[9.72196212e-11 3.55566171e-19 1.62708407e-08 4.93014352e-10
  4.07531131e-19 4.04937246e-14 9.97477144e-25 1.00000000e+00
  6.55548740e-11 1.02851588e-10]
 [1.47237875e-08 2.46068471e-10 9.99970078e-01 2.04319695e-05
  6.09304574e-16 2.80278103e-14 1.33527167e-09 3.53577747e-21
  9.55071573e-06 8.16783614e-17]
 [1.50826168e-07 9.99999285e-01 4.40007364e-09 1.09360784e-12
  3.45145956e-08 3.61518426e-10 2.13694635e-11 9.31930888e-09
  4.81676750e-07 4.37127819e-14]
 [1.00000000e+00 1.82629634e-17 7.77175226e-14 6.26987949e-17
  3.55430940e-17 8.90434807e-13 2.41729969e-10 1.84495042e-17
  2.86195456e-10 9.55759994e-13]]
```

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

output



```
[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.]]
```

Save the model

```
model.save("model.h5")
```

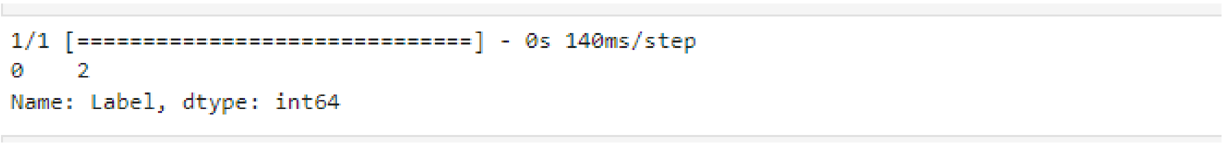
Test the saved model

```
model=load_model("model.h5")
```

In [20]:

```
img = Image.open("sample.png").convert("L")
img = img.resize((28, 28))
img2arr = np.array(img)
img2arr = img2arr.reshape(1, 28, 28, 1)
results  = model.predict(img2arr)
results = np.argmax(results,axis = 1)
results = pd.Series(results,name="Label")
print(results)
```

output



```
1/1 [==============================] - 0s 140ms/step
0    2
Name: Label, dtype: int64
```

Import Libraries

```
import requests
import numpy as np
from PIL import Image, ImageOps
import matplotlib.pyplot as plt
```

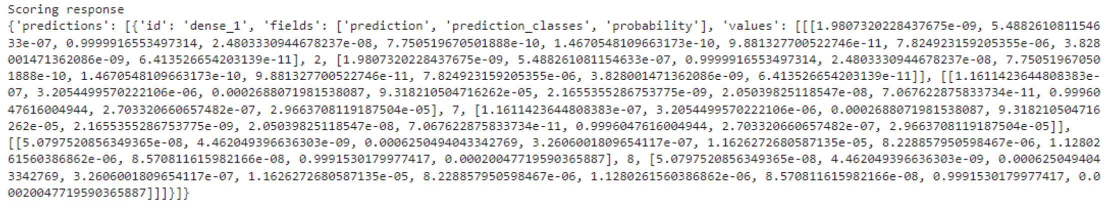
Input Pre-processing

```
img = Image.open(f"sample/sample 1.png").convert("L")
img = ImageOps.invert(img)
img = img.resize((28, 28))
img_arr = np.array(img)
img_arr = img_arr / 255.0
img_arr = img_arr.reshape(28, 28, 1)

img2 = Image.open(f"sample/sample 2.png").convert("L")
img2 = ImageOps.invert(img2)
img2 = img2.resize((28, 28))
img2_arr = np.array(img2)
img2_arr = img2_arr / 255.0
img2_arr = img2_arr.reshape(28, 28, 1)

img3 = Image.open(f"sample/sample 3.png").convert("L")
img3 = ImageOps.invert(img3)
img3 = img3.resize((28, 28))
img3_arr = np.array(img3)
img3_arr = img3_arr / 255.0
img3_arr = img3_arr.reshape(28, 28, 1)
```

Get results from deployed model

```
import requests

API_KEY = "hXFJaBL1u3AzIC6m6xVnnXWh3Bx_9is80s39JaByOKfY"
token_response = requests.post('https://iam.cloud.ibm.com/identity/token',
data={"apikey":
 API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})
mltoken = token_response.json()["access_token"]

header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' +
mltoken}

# NOTE: manually define and pass the array(s) of values to be scored in the
next line
payload_scoring = {"input_data": [{"fields": [], "values": [img_arr.tolist(),
img2_arr.tolist(), img3_arr.tolist()]}]}

response_scoring = requests.post('https://us-
south.ml.cloud.ibm.com/ml/v4/deployments/8c54e659-5b13-406f-8e9f-
2e240c2d5012/predictions?version=2022-11-16', json=payload_scoring,
 headers={'Authorization': 'Bearer ' + mltoken})
print("Scoring response")
print(response_scoring.json())
```

output

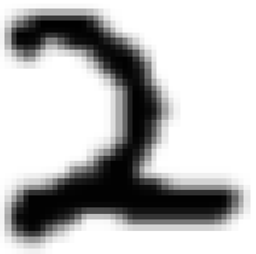


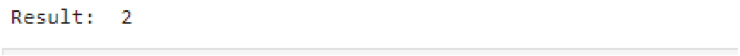
```
Scoring response
{'predictions': [{'id': 'dense_1', 'fields': ['prediction', 'prediction_classes', 'probability'], 'values': [[[1.9807320228437675e-09, 5.4882610811546
33e-07, 0.9999916553497314, 2.4803330944678237e-08, 7.750519670501888e-10, 1.4670548109663173e-10, 9.881327700522746e-11, 7.824923159205355e-06, 3.828
001471362086e-09, 6.413526654203139e-11], 2, [1.9807320228437675e-09, 5.488261081154633e-07, 0.9999916553497314, 2.4803330944678237e-08, 7.75051967050
1888e-10, 1.4670548109663173e-10, 9.881327700522746e-11, 7.824923159205355e-06, 3.828001471362086e-09, 6.413526654203139e-11]], [[1.1611423644808383e-
07, 3.2054499570222106e-06, 0.0002688071981538087, 9.318210504716262e-05, 2.1655355286753775e-09, 2.05039825118547e-08, 7.067622875833734e-11, 0.99960
47616004944, 2.703320660657482e-07, 2.9663708119187504e-05], 7, [1.1611423644808383e-07, 3.2054499570222106e-06, 0.0002688071981538087, 9.318210504716
262e-05, 2.1655355286753775e-09, 2.05039825118547e-08, 7.067622875833734e-11, 0.9996047616004944, 2.703320660657482e-07, 2.9663708119187504e-05]],
[[5.0797520856349365e-08, 4.462049396636303e-09, 0.0006250494043342769, 3.2606001809654117e-07, 1.1626272680587135e-05, 8.228857950598467e-06, 1.12802
61560386862e-06, 8.570811615982166e-08, 0.9991530179977417, 0.00020047719590365887], 8, [5.0797520856349365e-08, 4.462049396636303e-09, 0.000625049404
3342769, 3.2606001809654117e-07, 1.1626272680587135e-05, 8.228857950598467e-06, 1.1280261560386862e-06, 8.570811615982166e-08, 0.9991530179977417, 0.0
0020047719590365887]]]}]}
```

Display results

```
plt.imshow(plt.imread("sample/sample 1.png"))
plt.axis('off')
plt.show()
print("Result: ", response_scoring.json()['predictions'][0]['values'][0][1])
```

<u>output</u>

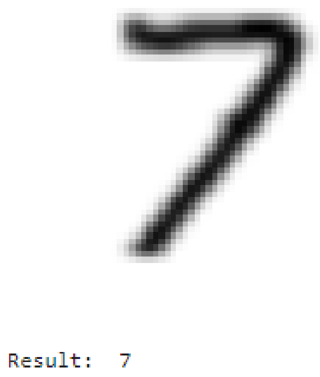




```
Result:  2
```

```
plt.imshow(plt.imread("sample/sample 2.png"))
plt.axis('off')
plt.show()
print("Result: ", response_scoring.json()['predictions'][0]['values'][1][1])
```

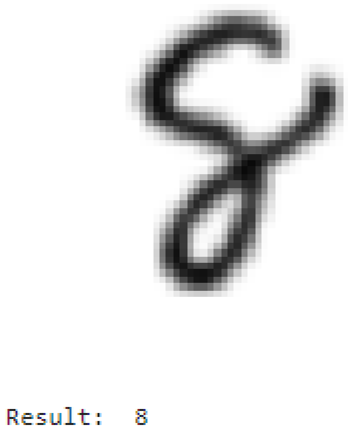
<u>output</u>



```
Result:  7
```

```
plt.imshow(plt.imread("sample/sample 3.png"))
plt.axis('off')
plt.show()
print("Result: ", response_scoring.json()['predictions'][0]['values'][2][1])
```

<u>output</u>



```
Result:  8
```

8. TESTING

8.1 Test Cases

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

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	8	0	0	8
Client Application	50	0	0	50
Security	3	0	0	3
Outsource Shipping	2	0	0	2
Exception Reporting	8	0	0	8
Final Report Output	5	0	0	5
Version Control	3	0	0	3

8.2 User Acceptance Testing

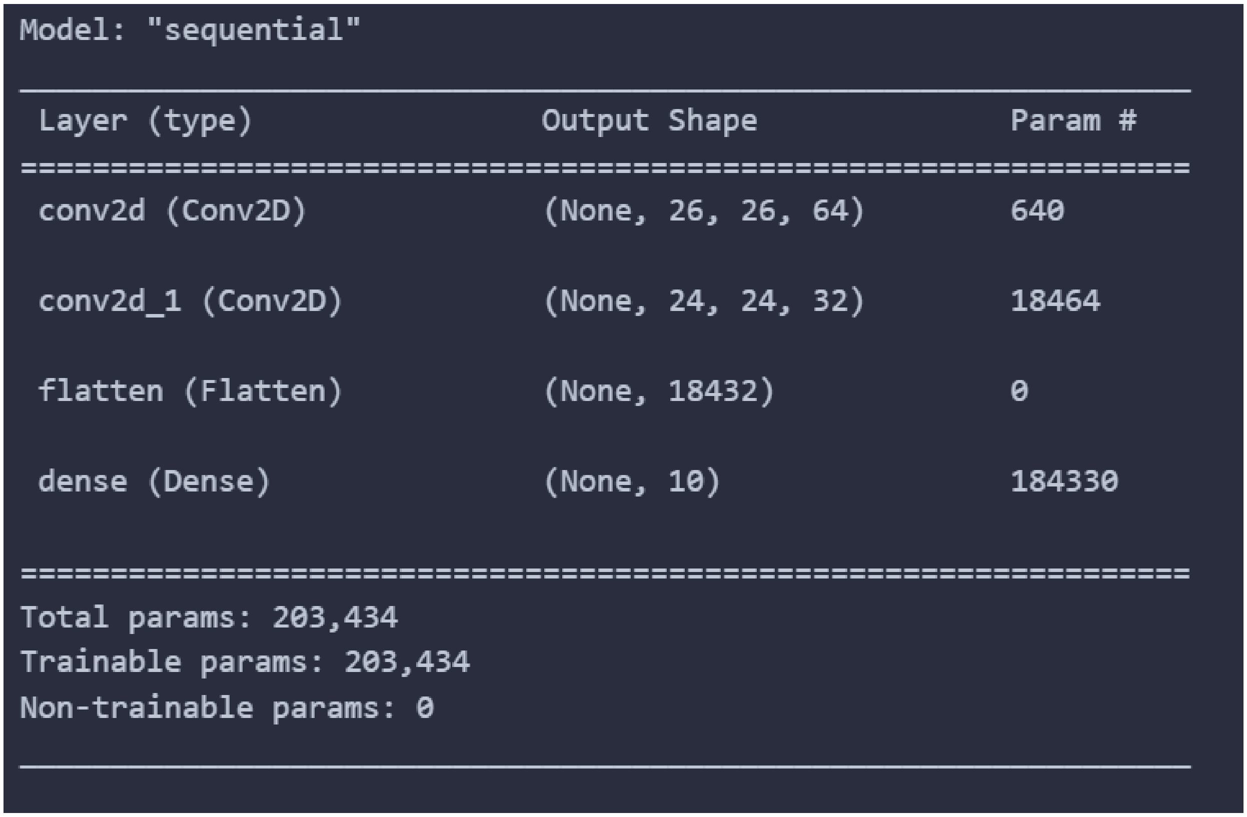
This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	8	3	3	2	16
Duplicate	2	1	2	0	5
External	2	3	0	1	6
Fixed	9	3	3	18	33
Not Reproduced	1	2	0	0	3
Skipped	0	1	1	1	3
Won't Fix	0	5	2	1	8
Totals	22	18	11	23	74

9. RESULTS

9.1 Performance Metrics

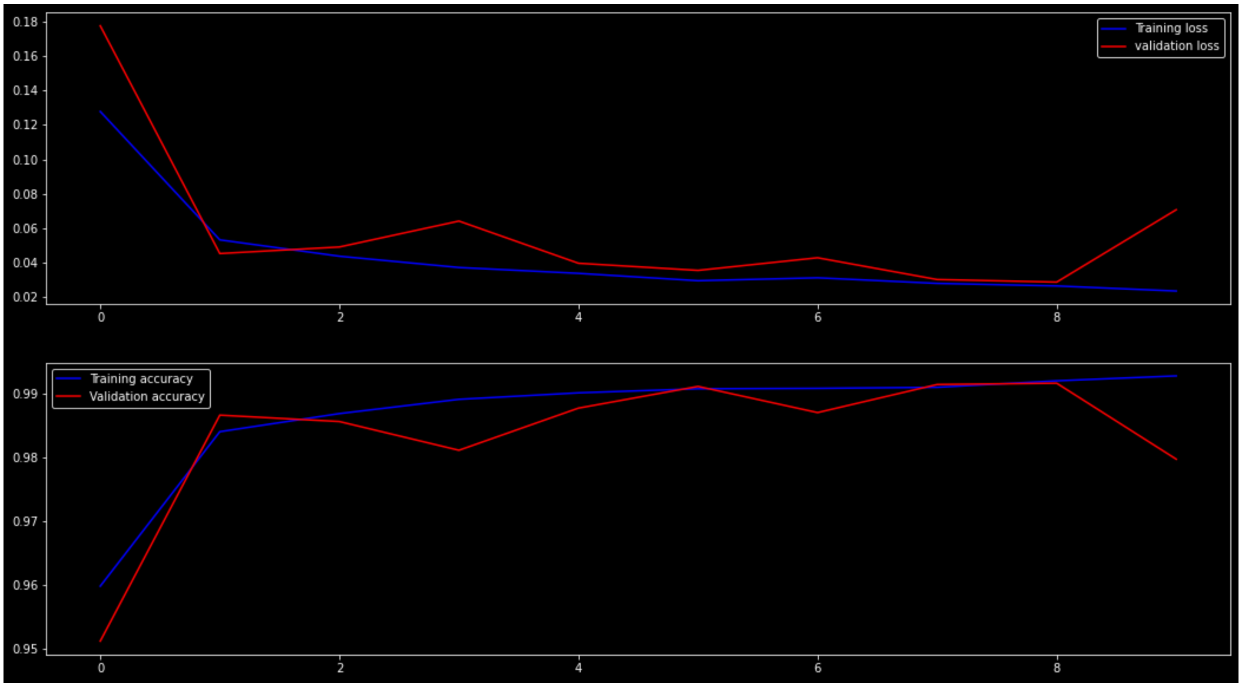
MODEL SUMMARY



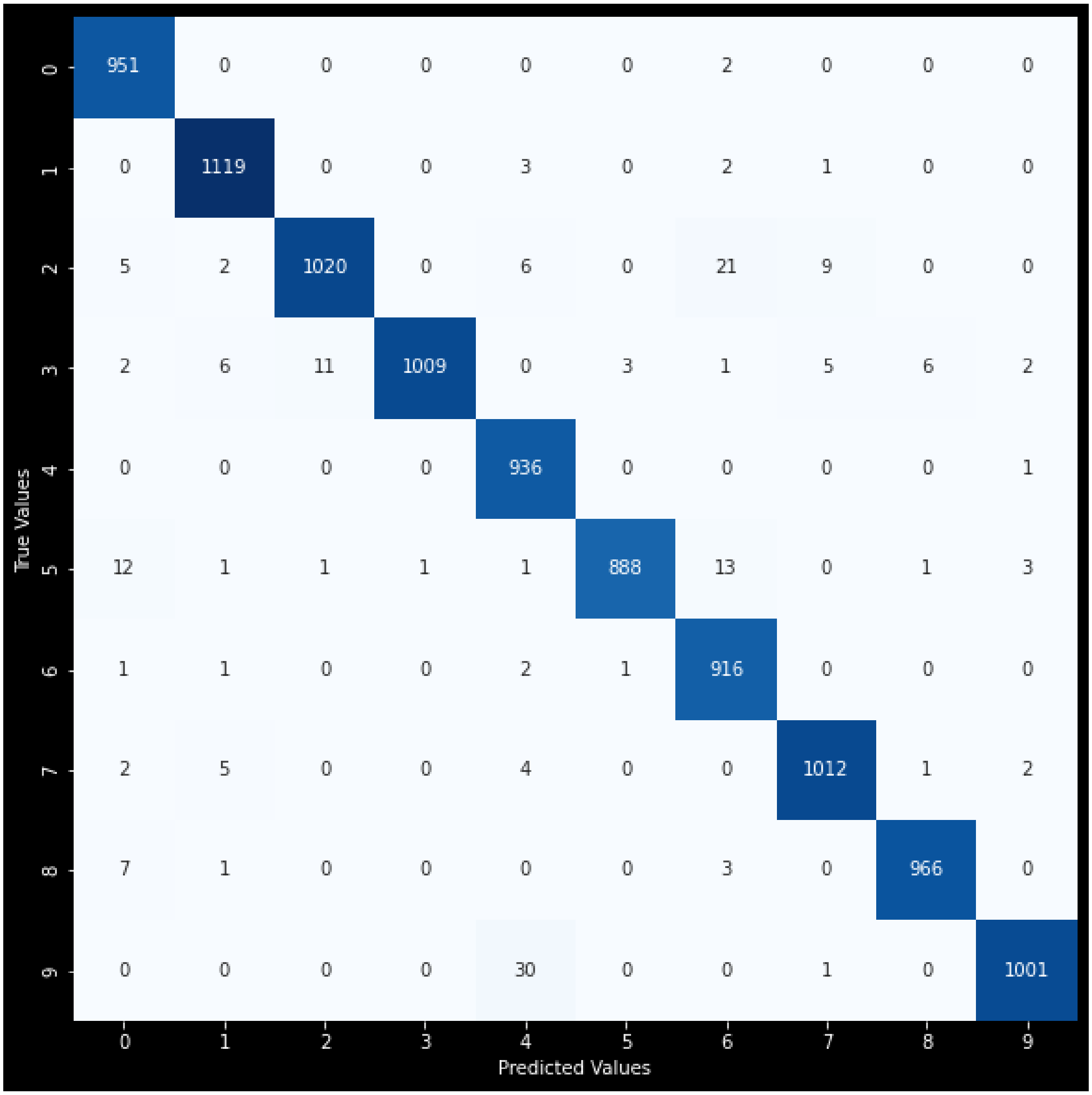
```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 26, 26, 64)        640

 conv2d_1 (Conv2D)           (None, 24, 24, 32)        18464

 flatten (Flatten)           (None, 18432)             0

 dense (Dense)               (None, 10)                184330

=================================================================
Total params: 203,434
Trainable params: 203,434
Non-trainable params: 0
_________________________________________________________________
```

ACCURACY

CONTENT	VALUE
Training Accuracy	99.14%
Training Loss	2.70%
Validation Accuracy	97.76%
Validation Loss	10.36%

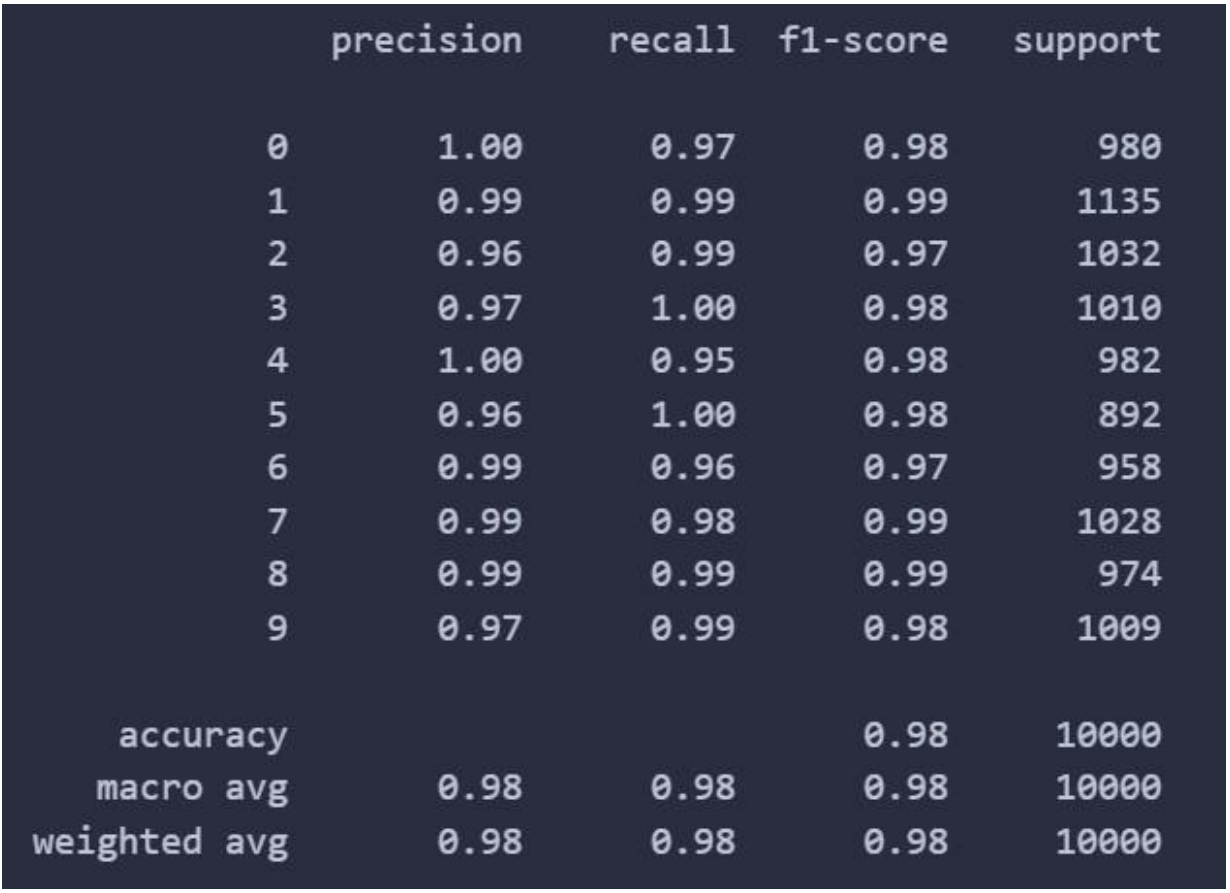


CONFUSION MATRIX



True Values \ Predicted Values	0	1	2	3	4	5	6	7	8	9
0	951	0	0	0	0	0	2	0	0	0
1	0	1119	0	0	3	0	2	1	0	0
2	5	2	1020	0	6	0	21	9	0	0
3	2	6	11	1009	0	3	1	5	6	2
4	0	0	0	0	936	0	0	0	0	1
5	12	1	1	1	1	888	13	0	1	3
6	1	1	0	0	2	1	916	0	0	0
7	2	5	0	0	4	0	0	1012	1	2
8	7	1	0	0	0	0	3	0	966	0
9	0	0	0	0	30	0	0	1	0	1001

C



```
              precision    recall  f1-score   support

           0       1.00      0.97      0.98       980
           1       0.99      0.99      0.99      1135
           2       0.96      0.99      0.97      1032
           3       0.97      1.00      0.98      1010
           4       1.00      0.95      0.98       982
           5       0.96      1.00      0.98       892
           6       0.99      0.96      0.97       958
           7       0.99      0.98      0.99      1028
           8       0.99      0.99      0.99       974
           9       0.97      0.99      0.98      1009

    accuracy                           0.98     10000
   macro avg       0.98      0.98      0.98     10000
weighted avg       0.98      0.98      0.98     10000
```

10.ADVANTAGES & DISADVANTAGES

ADVANTAGE:

It is used in the postal service to recognize the pin code by the postman to identify the accurate place using this system.

It is used in the bank to identify the handwritten digits in the challans to avoid the money loss because of wrong identification of digits.

DISADVANTAGE:

It is not useful for illiterate peoples because they don’t have enough knowledge to use this application.

It mainly focuses on only digit recognition, not for character recognitions.

CONFUSION MATRIX

11.CONCLUSION

By using this system, we can easily identify the handwritten digits. It will reduce the time To identify the accurate result by using the CNN model.

The proposed method obtained 98% accuracy and is able to identify real-world images as well; the loss percentage in both training and evaluation is less than 0.1, which is negligible.

12.FUTURE SCOPE

In future we can add features like,

- character recognition
- character recognition for multiple languages
- character with digit recognition
- make useful for blind peoples

13.APPENDIX

Source Code

```
{
"cells": [
{
"cell_type": "markdown",
"id": "5c5236a8",
"metadata": {},
"source": [
"Import packages"
]
},
{
"cell_type": "code",
"execution_count": 1,
"id": "e869d9d2",
"metadata": {},
"outputs": [],
"source": [
"import numpy as np\n",
"import pandas as pd\n",
"import matplotlib.pyplot as plt\n",
"from keras.utils import np_utils\n",
"from tensorflow.keras.datasets import mnist\n",
"from tensorflow.keras.models import Sequential\n",
"from tensorflow.keras.layers import Conv2D, Dense, Flatten\n",
"from tensorflow.keras.optimizers import Adam\n",
"from tensorflow.keras.models import load_model\n",
"from PIL import Image, ImageOps"
]
},
{
"cell_type": "markdown",
"id": "fcc79e8e",
"metadata": {},
"source": [
"Load the data"
]
},
{
"cell_type": "code",
"execution_count": 2,
"id": "e977ee08",
"metadata": {},
"outputs": [],
"source": [
"(X_train, y_train), (X_test, y_test) = mnist.load_data()"
]
},
{
"cell_type": "markdown",
"id": "d8aa2993",
"metadata": {},
"source": [
"Data analysis"
]
},
{
"cell_type": "code",
"execution_count": 3,
"id": "03ab1283",
"metadata": {},
"outputs": [
{
"name": "stdout",
"output_type": "stream",
"text": [
"(60000, 28, 28)\n",
"(10000, 28, 28)\n"
]
}
],
"source": [
"print(X_train.shape)\n",
"print(X_test.shape)"
]
},
{
"cell_type": "code",
"execution_count": 4,
"id": "5b4e57f4",
"metadata": {},
"outputs": [
{
"data": {
"text/plain": [
"array([[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,\n",
" 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,\n",
" 0, 0],\n",
" [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3,\n",
" 18, 18, 18, 126, 136, 175, 26, 166, 255, 247, 127, 0, 0,\n",
" 0, 0],\n",
" [ 0, 0, 0, 0, 0, 0, 0, 0, 30, 36, 94, 154, 170,\n",
" [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,\n",
" 0, 0, 16, 93, 252, 253, 187, 0, 0, 0, 0, 0, 0,\n",
" 0, 0],\n",
" [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,\n",
" 0, 0, 0, 0, 249, 253, 249, 64, 0, 0, 0, 0, 0,\n",
" 0, 0],\n",
" [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,\n",
" 0, 46, 130, 183, 253, 253, 207, 2, 0, 0, 0, 0, 0,\n",
```

CONFUSION MATRIX

```
" 0, 0],\n",
CONFUSION MATRIX
" [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 39,\n",
" 148, 229, 253, 253, 253, 250, 182, 0, 0, 0, 0, 0, 0,\n",
" 0, 0],\n",
" [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 24, 114, 221,\n",
" 253, 253, 253, 253, 201, 78, 0, 0, 0, 0, 0, 0, 0,\n",
" 0, 0],\n",
" [ 0, 0, 0, 0, 0, 0, 0, 0, 23, 66, 213, 253, 253,\n",
" 253, 253, 198, 81, 2, 0, 0, 0, 0, 0, 0, 0, 0,\n",
" 0, 0],\n",
" [ 0, 0, 0, 0, 0, 0, 18, 171, 219, 253, 253, 253, 253,\n",
" 195, 80, 9, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,\n",
" 0, 0],\n",
" [ 0, 0, 0, 0, 55, 172, 226, 253, 253, 253, 253, 244, 133,\n",
" 11, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,\n",
" 0, 0],\n",
" [ 0, 0, 0, 0, 136, 253, 253, 253, 212, 135, 132, 16, 0,\n",
" 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,\n",
" 0, 0],\n",
" [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,\n",
" 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,\n",
},
"execution_count": 4,
"metadata": {},
"output_type": "execute_result"
}
],
"source": [
```

CONFUSION MATRIX

```
"X_train[0]"
]
},
{
"cell_type": "code",
"execution_count": 5,
"id": "4db1edb5",
"metadata": {},
"outputs": [
{
"data": {
"text/plain": [
"5"
]
},
"execution_count": 5,
"metadata": {},
"output_type": "execute_result"
}
],
"source": [
"y_train[0]"
]
},
{
"cell_type": "code",
"execution_count": 6,
"id": "64da4472",
```

CONFUSION MATRIX

"metadata": {},
"outputs": [
{
"data": {
"text/plain": [
"<matplotlib.image.AxesImage at 0x250a85a2a30>"
]
},
"execution_count": 6,
"metadata": {},
"output_type": "execute_result"
},
{
"data": {
"image/png":
"iVBORw0KGgoAAAANSUhEUgAAAPsAAAD4CAYAAAAq5pAIAAAAO
XRFW
HRTb2Z0d2FyZQBNYXRwbG90bGliIHZlcnNpb24zLjMuNCwgaHR0cHM6
Ly9tY
XRwbG90bGliLm9yZy8QVMy6AAAACXBIWXMAAAsTAAALEwEAmpw
YAAA
OX0lEQVR4nO3dbYxc5XnG8euKbUwxJvHGseMQFxzjFAg0Jl0ZkBFQoV
CCIgGK
CLGiiFBapwlOQutKUFoVWtHKrRIiSimSKS6m4iWQgPAHmsSyECRqcF
moARO
HN+MS4+0aswIDIfZ6fffDjqsFdp5dZs68eO//T1rNzLnnzLk1cPmcmeeceRwR
AjD5fa

DTDQBoD8IOJEHYgSQIO5AEYQeSmNrOjR3i6XGoZrRzk0Aqv9Fb2ht7P
FatqbDb
PkfS9ZKmSPrXiFhVev6hmqGTfVYzmwRQsDE21K01fBhve4qkGyV9TtLxk
pbZPr7
R1wPQWs18Zl8i6fmI2BoReyXdJem8atoCULVmwn6kpF+Nery9tuwdbC+33
We7b0h
7mtgcgGY0E/axvgR4z7m3EbE6InojoneapjexOQDNaCbs2yXNH/X445J2NNc
OgFZ
pJuyPSlpke4HtQyR9SdK6atoCULWGh94iYp/tFZJ+rJGhtzUR8XRlnQGoV
FPj7BH
": {
"needs_background": "light"
},
"output_type": "display_data"
}
],
"source": [
"plt.imshow(X_train[0])"
]
},
{
"cell_type": "markdown",
"id": "42ec1c47",
"metadata": {},
"source": [
"Data Pre-processing"
]
},

CONFUSION MATRIX

{
"cell_type": "code",
"execution_count": 7,
"id": "8453610d",
"metadata": {},
"outputs": [],
"source": [
"X_train = X_train.reshape(60000, 28, 28, 1).astype('float32')\n",
"X_test = X_test.reshape(10000, 28, 28, 1).astype('float32')"
]
},
{
"cell_type": "code",
"execution_count": 8,
"id": "7abb41b0",
"metadata": {},
"outputs": [],
"source": [
"number_of_classes = 10\n",
"Y_train = np_utils.to_categorical(y_train, number_of_classes)\n",
"Y_test = np_utils.to_categorical(y_test, number_of_classes)"
]
},
{
"cell_type": "code",
"execution_count": 9,
"id": "fd70ac31",
"metadata": {},

```
"outputs": [
{
CONFUSION MATRIX
"data": {
"text/plain": [
"array([0., 0., 0., 0., 0., 1., 0., 0., 0., 0.], dtype=float32)"
]
},
"execution_count": 9,
"metadata": {},
"output_type": "execute_result"
}
],
"source": [
"Y_train[0]"
]
},
{
"cell_type": "markdown",
"id": "161265a6",
"metadata": {},
"source": [
"Create model"
]
},
{
"cell_type": "code",
"execution_count": 10,
"id": "90bf6dd3",
```

"metadata": {},
"outputs": [],
"source": [
"model = Sequential()\n",
"model.add(Conv2D(64, (3, 3), input_shape=(28, 28, 1), activation=\"relu\"))\n",
"model.add(Conv2D(32, (3, 3), activation=\"relu\"))\n",
"model.add(Flatten())\n",
"model.add(Dense(number_of_classes, activation=\"softmax\"))"
]
},
{
"cell_type": "code",
"execution_count": 11,
"id": "1cf72280",
"metadata": {},
"outputs": [],
"source": [
"model.compile(loss='categorical_crossentropy', optimizer=\"Adam\", metrics=[\"accuracy\"])"
]
},
{
"cell_type": "markdown",
"id": "273dd04a",
"metadata": {},
"source": [
"Train the model"

```
]
},
```

CONFUSION MATRIX

```
{
"cell_type": "code",
"execution_count": 12,
"id": "294095fe",
"metadata": {},
"outputs": [
{
"name": "stdout",
"output_type": "stream",
"text": [
"Epoch 1/5\n",
"1875/1875 [==============================] - 92s 49ms/step - loss:
0.2377 - accuracy: 0.9523 - val_loss: 0.0928 - val_accuracy: 0.9719\n",
"Epoch 2/5\n",
"1875/1875 [==============================] - 100s 53ms/step - loss:
0.0656 - accuracy: 0.9803 - val_loss: 0.0910 - val_accuracy: 0.9729\n",
"Epoch 3/5\n",
"1875/1875 [==============================] - 105s 56ms/step - loss:
0.0458 - accuracy: 0.9857 - val_loss: 0.1193 - val_accuracy: 0.9684\n",
"Epoch 4/5\n",
"1875/1875 [==============================] - 109s 58ms/step - loss:
0.0368 - accuracy: 0.9883 - val_loss: 0.1116 - val_accuracy: 0.9766\n",
"Epoch 5/5\n",
"1875/1875 [==============================] - 106s 56ms/step - loss:
0.0274 - accuracy: 0.9914 - val_loss: 0.1173 - val_accuracy: 0.9748\n"
]
```

CONFUSION MATRIX

```
},
{
"data": {
"text/plain": [
"<keras.callbacks.History at 0x250a7b7ddf0>"
]
},
"execution_count": 12,
"metadata": {},
"output_type": "execute_result"
}
],
"source": [
"model.fit(X_train, Y_train, batch_size=32, epochs=5, validation_data=(X_test,Y_test))"
]
},
{
"cell_type": "markdown",
"id": "9b14c399",
"metadata": {},
"source": [
"Test the model"
]
},
{
"cell_type": "code",
"execution_count": 13,
"id": "452c47bc",
```

CONFUSION MATRIX

```
"metadata": {},
"outputs": [
{
"name": "stdout",
"output_type": "stream",
"text": [
"Metrics (Test Loss & Test Accuracy): \n",
"[0.11725354194641113, 0.9747999906539917]\n"
]
}
],
"source": [
"metrics = model.evaluate(X_test, Y_test, verbose=0)\n",
"print(\"Metrics (Test Loss & Test Accuracy): \")\n",
"print(metrics)"
]
},
{
"cell_type": "code",
"execution_count": 14,
"id": "ba63597f",
"metadata": {},
"outputs": [
{
"name": "stdout",
"output_type": "stream",
"text": [
```

CONFUSION MATRIX

```
"1/1 [==============================] - 0s 82ms/step\n",
" [[2.0987019e-13 2.4276264e-14 1.3246451e-09 1.8783747e-05 1.2520702e-19\n",
" 1.8974715e-15 1.2059106e-19 9.9998116e-01 5.5190859e-11 3.0066661e-12]\n",
" [3.6490474e-11 8.6607568e-11 9.9999988e-01 1.4327577e-09 6.2738566e-15\n",
" 1.4905276e-18 3.5467995e-09 3.2217497e-17 7.2324944e-08 2.6815476e-20]\n",
" [5.2345695e-07 9.8446769e-01 5.0688081e-04 2.0625328e-09 7.7538867e-03\n",
" 8.2934766e-06 2.1169055e-06 2.6437547e-07 7.2603868e-03 2.1643283e-10]\n",
" [1.0000000e+00 2.3432516e-15 1.6870000e-10 1.4166539e-14 4.9352419e-14\n",
" 8.6952261e-13 4.7428284e-10 3.0202582e-16 1.2158017e-12 3.0176420e-10]]\n"
]
}
],
"source": [
"prediction = model.predict(X_test[:4])\n",
"print(prediction)"
]
},
{
"cell_type": "code",
"execution_count": 16,
```

"id": "4dc08f2c",

CONFUSION MATRIX

"metadata": {},

"outputs": [

{

"name": "stdout",

"output_type": "stream",

"text": [

"[7 2 1 0]\n",

"[[0. 0. 0. 0. 0. 0. 0. 1. 0. 0.]\n",

" [0. 0. 1. 0. 0. 0. 0. 0. 0. 0.]\n",

" [0. 1. 0. 0. 0. 0. 0. 0. 0. 0.]\n",

" [1. 0. 0. 0. 0. 0. 0. 0. 0. 0.]]\n"

]

}

],

"source": [

"print(np.argmax(prediction, axis=1))\n",

"print(Y_test[:4])"

]

},

{

"cell_type": "markdown",

"id": "97233b2d",

"metadata": {},

"source": [

"Save the model"

]

},

CONFUSION MATRIX

{
"cell_type": "code",
"execution_count": 17,
"id": "2eb83236",
"metadata": {},
"outputs": [],
"source": [
"model.save(\"model.h5\")"
]
},
{
"cell_type": "markdown",
"id": "6a940125",
"metadata": {},
"source": [
"Test the saved model"
]
},
{
"cell_type": "code",
"execution_count": 18,
"id": "267154d5",
"metadata": {},
"outputs": [],
"source": [
"model=load_model(\"model.h5\")"
]
},

CONFUSION MATRIX

```
{
"cell_type": "code",
"execution_count": 20,
"id": "8308baec",
"metadata": {},
"outputs": [
{
"name": "stdout",
"output_type": "stream",
"text": [
"1/1 [==============================] - 0s 140ms/step\n",
"0 2\n",
"Name: Label, dtype: int64\n"
]
}
],
"source": [
"img = Image.open(\"sample.png\").convert(\"L\")\n",
"img = img.resize((28, 28))\n",
"img2arr = np.array(img)\n",
"img2arr = img2arr.reshape(1, 28, 28, 1)\n",
"results = model.predict(img2arr)\n",
"results = np.argmax(results,axis = 1)\n",
"results = pd.Series(results,name=\"Label\")\n",
"print(results)"
]
},
{
```

```
"cell_type": "code",
"execution_count": null,
"id": "0d65d64e",
"metadata": {},
"outputs": [],
"source": []
}
],
"metadata": {
"kernelspec": {
"display_name": "Python 3",
"language": "python",
"name": "python3"
},
"language_info": {
"codemirror_mode": {
"name": "ipython",
"version": 3
},
"file_extension": ".py",
"mimetype": "text/x-python",
"name": "python",
"nbconvert_exporter": "python",
"pygments_lexer": "ipython3",
"version": "3.8.8"
}
},
"nbformat": 4,
"nbformat_minor": 5
}
```

CONFUSION MATRIX

app.py

```
from flask import Flask,render_template,request
from recognizer import recognize
app=Flask(__name__)
@app.route('/')
def main():
return render_template("home.html")
@app.route('/predict',methods=['POST'])
def predict():
if request.method=='POST':
image = request.files.get('photo', '')
best, others, img_name = recognize(image)
return render_template("predict.html", best=best, others=others,
img_name=img_name)
if __name__=="__main__":
app.run()
```

requirements

```
absl-py==1.2.0
asttokens==2.0.5
astunparse==1.6.3
backcall==0.2.0
cachetools==5.2.0
certifi==2022.6.15
charset-normalizer==2.1.0
click==8.1.3
colorama==0.4.5
```

```
cycler==0.11.0
CONFUSION MATRIX
debugpy==1.6.2
decorator==5.1.1
entrypoints==0.4
executing==0.9.1
Flask==2.2.1
flatbuffers==1.12
fonttools==4.34.4
gast==0.4.0
google-auth==2.9.1
google-auth-oauthlib==0.4.6
google-pasta==0.2.0
grpcio==1.47.0
h5py==3.7.0
idna==3.3
ipykernel==6.15.1
ipython==8.4.0
itsdangerous==2.1.2
jedi==0.18.1
Jinja2==3.1.2
jupyter-client==7.3.4
jupyter-core==4.11.1
keras==2.9.0
Keras-Preprocessing==1.1.2
kiwisolver==1.4.4
libclang==14.0.6
Markdown==3.4.1
MarkupSafe==2.1.1
```

```
matplotlib==3.5.2
```

CONFUSION MATRIX

```
matplotlib-inline==0.1.3
nest-asyncio==1.5.5
numpy==1.23.1
oauthlib==3.2.0
opt-einsum==3.3.0
packaging==21.3
pandas==1.4.3
parso==0.8.3
pickleshare==0.7.5
Pillow==9.2.0
prompt-toolkit==3.0.30
protobuf==3.19.4
psutil==5.9.1
pure-eval==0.2.2
pyasn1==0.4.8
pyasn1-modules==0.2.8
Pygments==2.12.0
pyparsing==3.0.9
python-dateutil==2.8.2
pytz==2022.1
pywin32==304
pyzmq==23.2.0
requests==2.28.1
requests-oauthlib==1.3.1
rsa==4.9
six==1.16.0
stack-data==0.3.0
```

tensorboard==2.9.1

CONFUSION MATRIX

tensorboard-data-server==0.6.1

tensorboard-plugin-wit==1.8.1

tensorflow==2.9.1

tensorflow-estimator==2.9.0

tensorflow-io-gcs-filesystem==0.26.0

termcolor==1.1.0

tornado==6.2

traitlets==5.3.0

typing_extensions==4.3.0

urllib3==1.26.11

wcwidth==0.2.5

Werkzeug==2.2.1

wrapt==1.14.1

GitHub & Project Demo Link

GitHub link:

https://github.com/IBM-EPBL/IBM-Project-43208-1660714211

Project Demo Link:

https://www.mediafire.com/file/tcv05ds9kgrc0ab/Untitled_44_360p[1].mp4/file