Project Report

Team ID: PNT2022TMID24646

Project: A Novel Method for Handwritten Digit Recognition System

Team Members:

Kousalya B : 210419104090

Jeya Yoga Lakshmi K : 210419104072

Kavyashree N V : 210419104083

Madhusree M P : 210419104095

TABLE OF CONTENTS

1.Introduction

1.1 Project Overview

Handwritten digit recognition is the ability of a computer to recognize the human handwritten digits from different sources like images, papers, touch screens, etc, and classify them into 10 predefined classes (0-9). Digit recognition has many applications like number plate recognition, postal mail sorting, bank check processing, etc. The MNIST data collection, which contains 70000 handwritten digits, is frequently utilized for this recognition method. In order to train these photos and create a deep learning model, we use artificial neural networks. A web application is developed that allows users to upload pictures of handwritten numbers. The model examines this image and the detected result is returned to the UI.

1.2 Purpose

This project is useful for the customers those who are facing difficultieswith understanding the handwritten digits. These customers are present inthe places like schools, colleges, hospitals, post offices , etc.

2. Literature Survey

2.1 Existing Problem

An early notable attempt in the area of character recognition research is by Grimsdale in 1959. The origin of a great deal of research work in the early sixties was based on an approach known as analysis by-synthesis method suggested by Eden in 1968. The great importance of Eden's work was that he formally proved that all handwritten characters are formed by a finite number of schematic features, a point that was implicitly included in previous works.

2.2 References

1. 12 June 2020 Improved Handwritten Digit Recognition Using Convolutional Neural Networks (CNN), Savita Ahlawat,Amit Choudhary, Anand Nayyar,Saurabh Singh,and Byungun Yoon.

2. 4 July 2020 Handwritten Digit Recognition Using Various Machine Learning Algorithms and Models,Pranit Patil and Bhupinder Kaur.

3. 2020 Handwritten Digit Recognition Using Computer Vision,Ashish Shekhar and Ajay Kaushik

4. 6 June 2019 Handwritten Digit Recognition using CNN, Vijayalaxmi R Rudraswamimath and Bhavanishankar K.

5. 2019 Recognition of Handwritten Digit using Convolutional Neural Network (CNN), Md. Anwar Hossain & Md. Mohon Ali.

6. 31 August 2019 An efficient and improved scheme for handwritten digit recognition based on convolutional neural network,Saqib Ali, Zeeshan Shaukat, Muhammad Azeem, Zareen Sakhawat, Tariq Mahmood & Khalil ur Rehman.

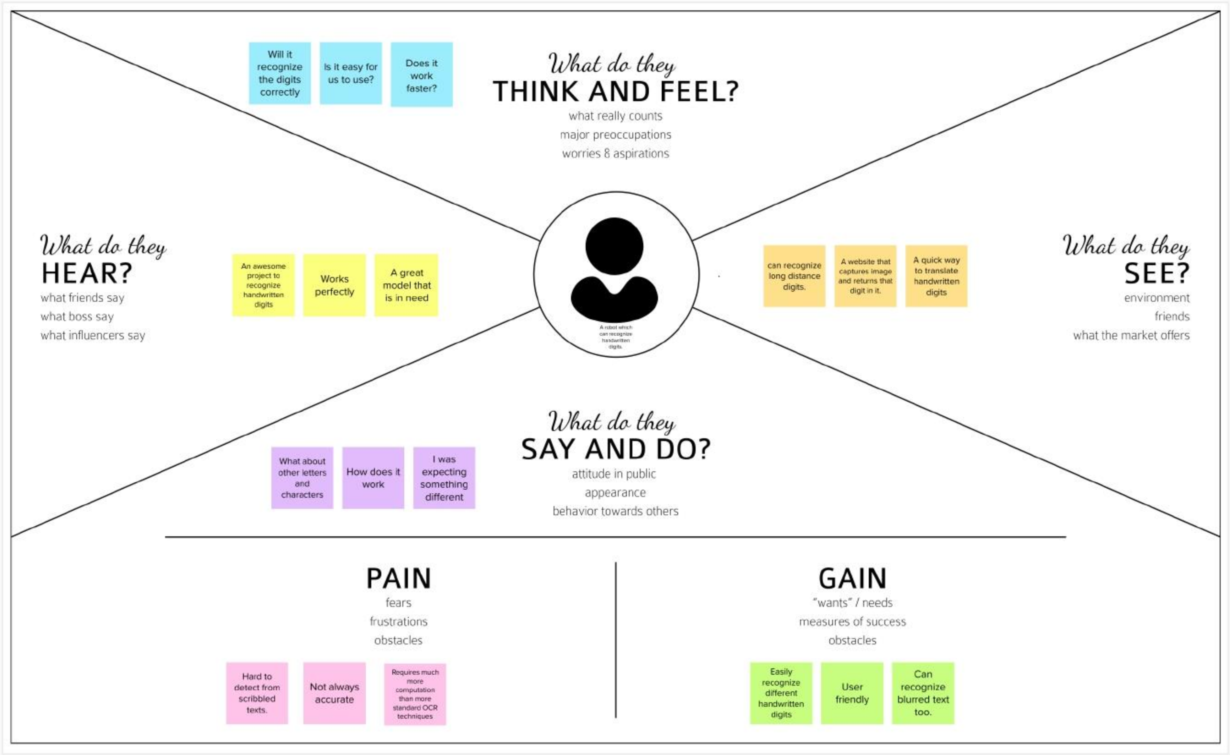
2.3 Problem Statement Definitions

It is easy for the human to perform a task accurately by practicing it repeatedly and memorizing it for the next time. Human brain can process and analyse images easily. Also, recognize the different elements present in the images. the goal is to correctly identify digits from a dataset of tens of thousands of handwritten images and experiment with different algorithms to learn first-hand what works well and how techniques compare.

3. Ideation and Proposed Solution

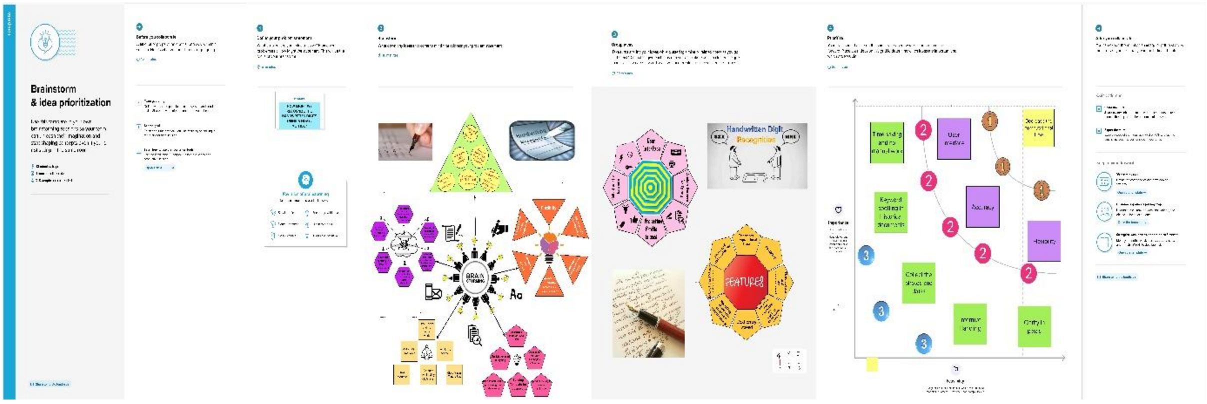
3.1 Empathy Map Canvas

An empathy map is a collaborative tool teams can use to gain a deeper insight into their customers. Much like a user persona, an empathy map can represent a group of users, such as a customer segment. The empathy map was originally created by Dave Gray and has gained much popularity within the agile community.


Will it recognize the digits correctly
Is it easy for us to use?
Does it work faster?
What do they THINK AND FEEL?
what really counts
major preoccupations
worries 8 aspirations
What do they HEAR?
what friends say
what boss say
what influencers say
An awesome project to recognize handwritten digits
Works perfectly
A great model that is in need
A robot which can recognize handwritten digits.
can recognize long distance digits.
A website that captures image and returns that digit in it.
A quick way to translate handwritten digits
What do they SEE?
environment
friends
what the market offers
What about other letters and characters
How does it work
I was expecting something different
What do they SAY AND DO?
attitude in public
appearance
behavior towards others
PAIN
fears
frustrations
obstacles
Hard to detect from scribbled texts.
Not always accurate
Requires much more computation than more standard OCR techniques
GAIN
"wants" / needs
measures of success
obstacles
Easily recognize different handwritten digits
User friendly
Can recognize blurred text too.

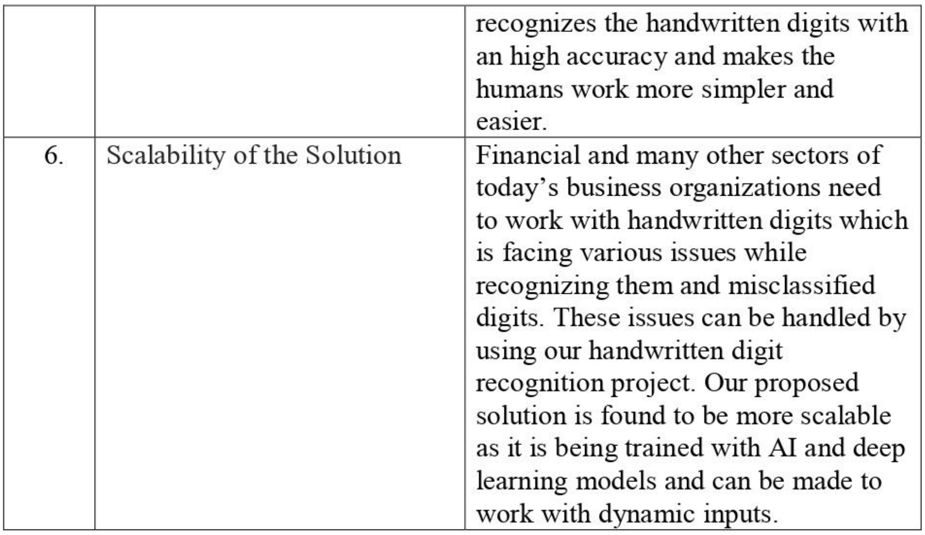
3.2 Ideation & Brainstorming

Ideation refers to the whole creative process of coming up with and communicating new ideas. Brainstorming is a group problem-solving method that involves the spontaneous contribution of creative ideas and solutions. This technique requires intensive, freewheeling discussion in which every member of the group is encouraged to think aloud and suggest as many ideas as possible based on their diverse knowledge.



3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	The digits that are wrote manually can be found of various sizes, shapes, thickness and directions which may lead to various difficulties that can be sorted out by using handwritten digit recognition
2.	Idea / Solution description	In order to overcome the above problem we will be implementing a classification algorithm that will be helpful to recognize the handwritten digits. This would be an efficient and an easy way to classify and recognize digits which has different appearances.
3.	Novelty / Uniqueness	✓ Result is found to be accurate by providing more number of dataset ✓ Can be used offline ✓ Digit can be recognized irrespective to their colour or background or text
4.	Social Impact / Customer Satisfaction	The main social impact of this project is to ensure the accuracy of recognizing the handwritten digits and implementing them may help customers find an easy way to recognize the handwritten digits in banking operations or in any other financial related works.
5.	Business Model (Revenue Model)	This method of handwritten digit recognition has been successfully achieved by many industries such as financial sectors, bank check processing, postal mail sorting, form data entry etc. Humans can find difficult to sort postal related mails or to enter the form data where our solution come into action which



		recognizes the handwritten digits with an high accuracy and makes the humans work more simpler and easier.
6.	Scalability of the Solution	Financial and many other sectors of today’s business organizations need to work with handwritten digits which is facing various issues while recognizing them and misclassified digits. These issues can be handled by using our handwritten digit recognition project. Our proposed solution is found to be more scalable as it is being trained with AI and deep learning models and can be made to work with dynamic inputs.

3.4 Problem Solution Fit

1. CUSTOMER SEGMENT(S): Our project is useful for the customers those who are facing difficulties with understanding the handwritten digits. These customers are present in the places like schools, colleges, hospitals, post offices, etc.

2. JOBS-TO-BE-DONE / PROBLEMS: Handwritten digits are not perfect and can be made with many different styles. So it is difficult to understand every handwriting and it may lead to errors.

3. TRIGGERS: Effectively extracting accurate handwritten digits from the image.

4. EMOTIONS - BEFORE and AFTER: Before – Worried and confused about the handwritten digits. After – Easily classified those handwritten digits.

5. AVAILABLE SOLUTIONS: There is no particular solution for recognizing handwritten digits but we some applications like Google lens which will predict everything.

6. CUSTOMER CONSTRAINTS: They might think that it won’t recognize the digits correctly.

7. BEHAVIOUR: The customers try to predict the handwritten digits by giving the image to the software and check the accuracy by checking whether the predicted digits are correct.

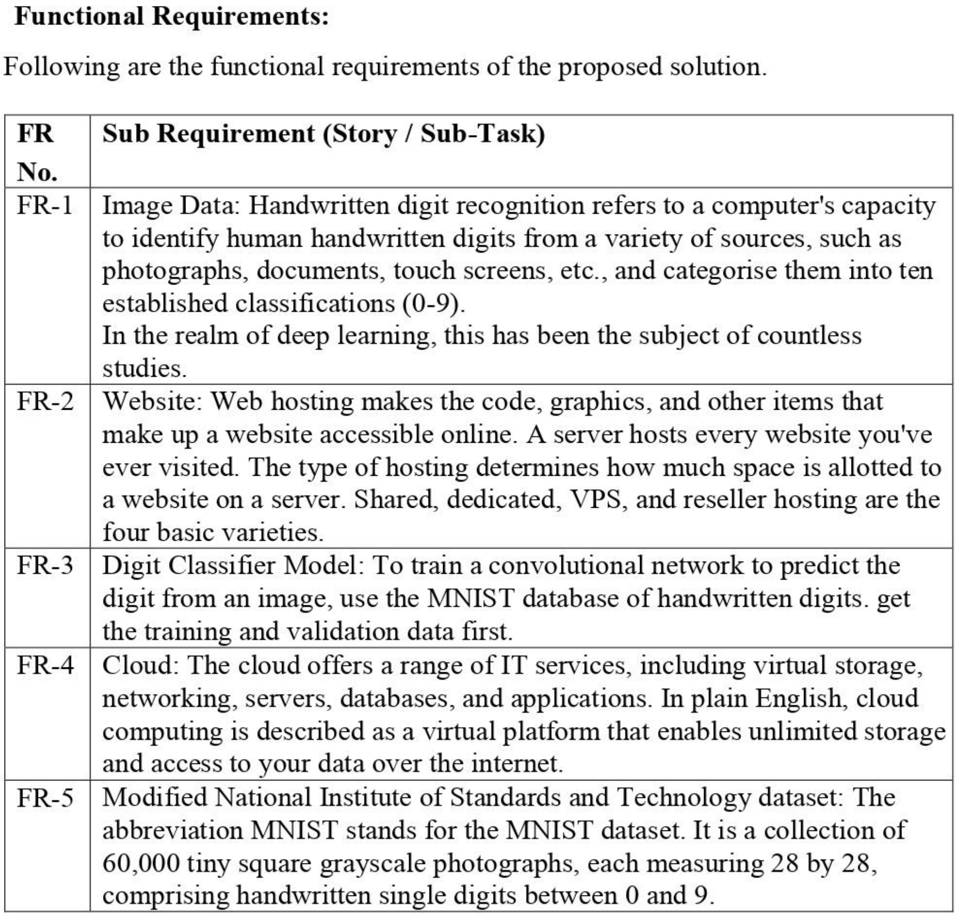
8. CHANNELS OF BEHAVIOUR: Utilizing the software that is offered in the online market. Enlisting the assistance of nearby people in order to identify the numbers that their clients have scribbled.

9. PROBLEM ROOT CAUSE: Because handwritten number recognition is not an optical character recognition, there are numerous difficulties due to the wide variety of writing styles used by different people. Customers find it difficult to read the handwritten digits as different people use different writing styles and different languages. This investigation offers a thorough comparison of various deep literacy and machine literacy algorithms for handwritten number recognition.

10.SOLUTION: A novel method for handwritten digit recognition system helps in recognizing the handwritten digits that uses MNIST dataset for training the model. The model gets the image of the handwritten digit and recognizes the handwritten digit. Convolution neural networks algorithm is used over the MNIST dataset to recognize the handwritten digits.

4. Requirement Analysis

4.1 Functional requirements

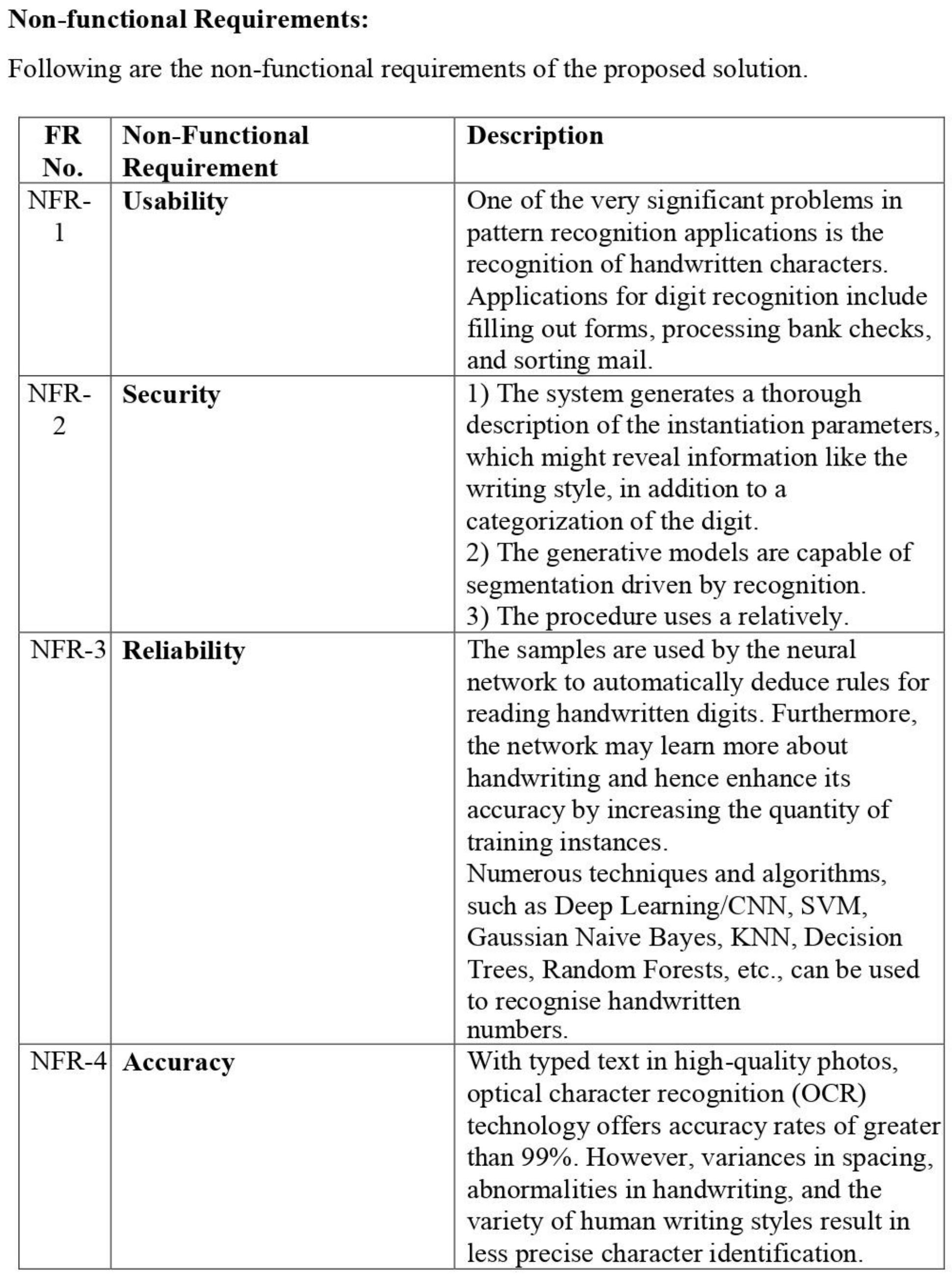


Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Sub Requirement (Story / Sub-Task)
FR-1	Image Data: Handwritten digit recognition refers to a computer's capacity to identify human handwritten digits from a variety of sources, such as photographs, documents, touch screens, etc., and categorise them into ten established classifications (0-9). In the realm of deep learning, this has been the subject of countless studies.
FR-2	Website: Web hosting makes the code, graphics, and other items that make up a website accessible online. A server hosts every website you've ever visited. The type of hosting determines how much space is allotted to a website on a server. Shared, dedicated, VPS, and reseller hosting are the four basic varieties.
FR-3	Digit Classifier Model: To train a convolutional network to predict the digit from an image, use the MNIST database of handwritten digits. get the training and validation data first.
FR-4	Cloud: The cloud offers a range of IT services, including virtual storage, networking, servers, databases, and applications. In plain English, cloud computing is described as a virtual platform that enables unlimited storage and access to your data over the internet.
FR-5	Modified National Institute of Standards and Technology dataset: The abbreviation MNIST stands for the MNIST dataset. It is a collection of 60,000 tiny square grayscale photographs, each measuring 28 by 28, comprising handwritten single digits between 0 and 9.

4.2 Non-Functional Requirements



Non-functional Requirements:

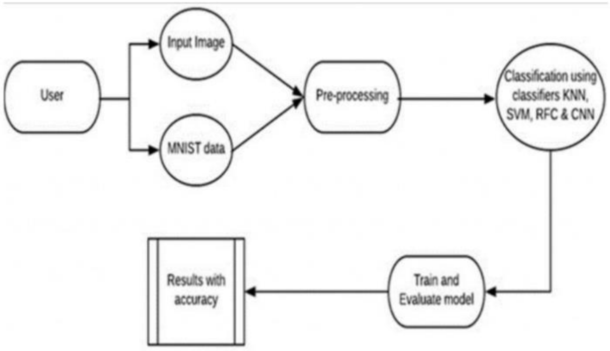
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	One of the very significant problems in pattern recognition applications is the recognition of handwritten characters. Applications for digit recognition include filling out forms, processing bank checks, and sorting mail.
NFR-2	**Security**	1) The system generates a thorough description of the instantiation parameters, which might reveal information like the writing style, in addition to a categorization of the digit. 2) The generative models are capable of segmentation driven by recognition. 3) The procedure uses a relatively.
NFR-3	**Reliability**	The samples are used by the neural network to automatically deduce rules for reading handwritten digits. Furthermore, the network may learn more about handwriting and hence enhance its accuracy by increasing the quantity of training instances. Numerous techniques and algorithms, such as Deep Learning/CNN, SVM, Gaussian Naive Bayes, KNN, Decision Trees, Random Forests, etc., can be used to recognise handwritten numbers.
NFR-4	**Accuracy**	With typed text in high-quality photos, optical character recognition (OCR) technology offers accuracy rates of greater than 99%. However, variances in spacing, abnormalities in handwriting, and the variety of human writing styles result in less precise character identification.

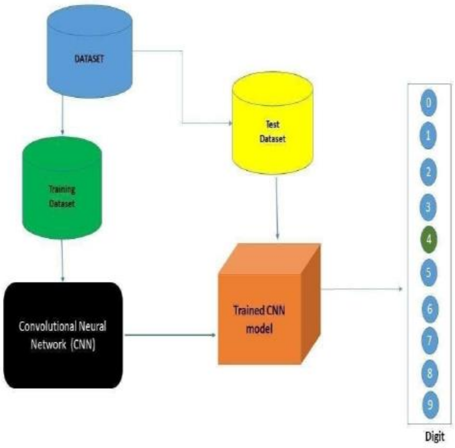
5. Project Design

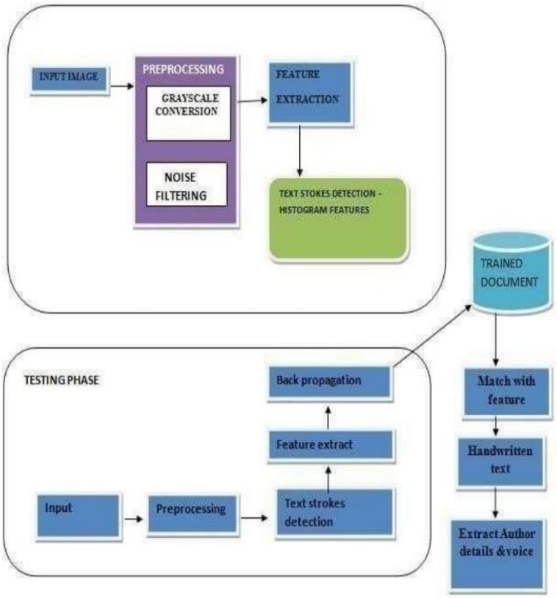
5.1 Data Flow Diagrams

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.

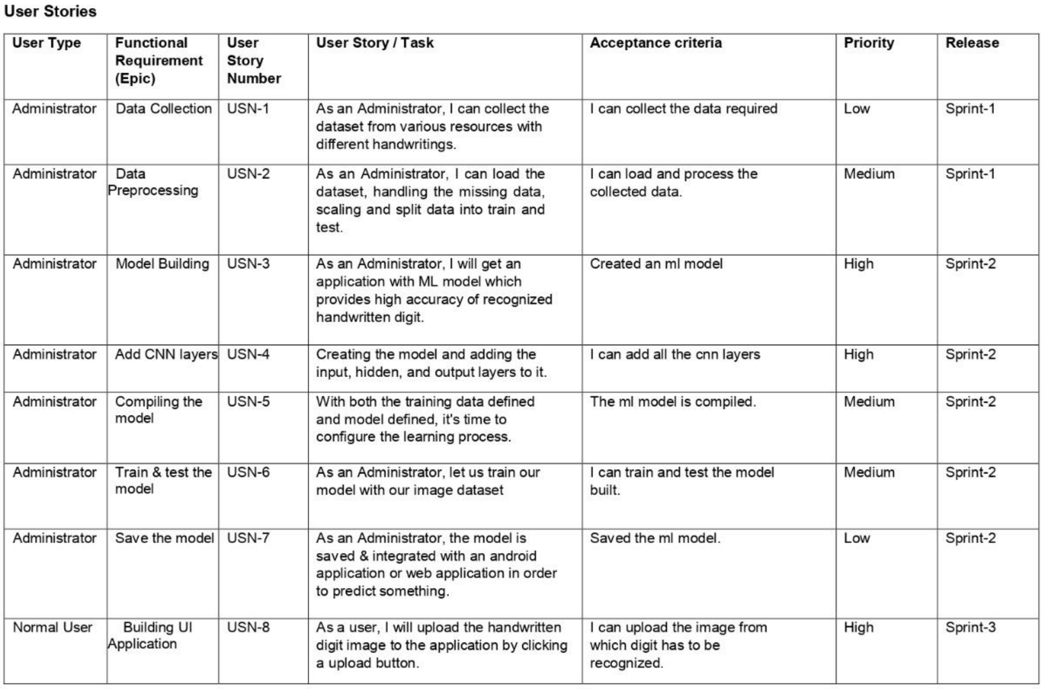


5.2 Solution and Technical Architecture



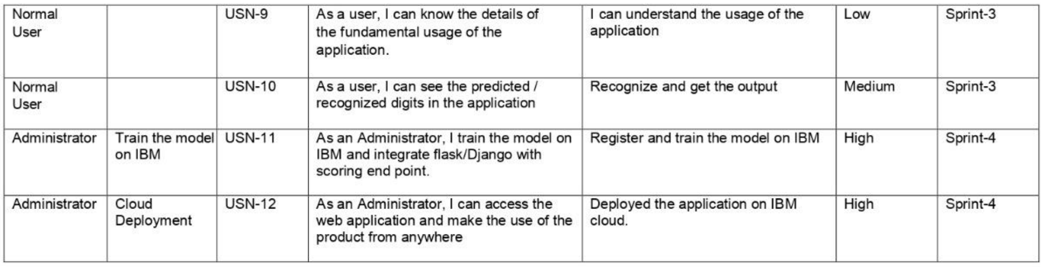


5.3 User Stories



User Stories

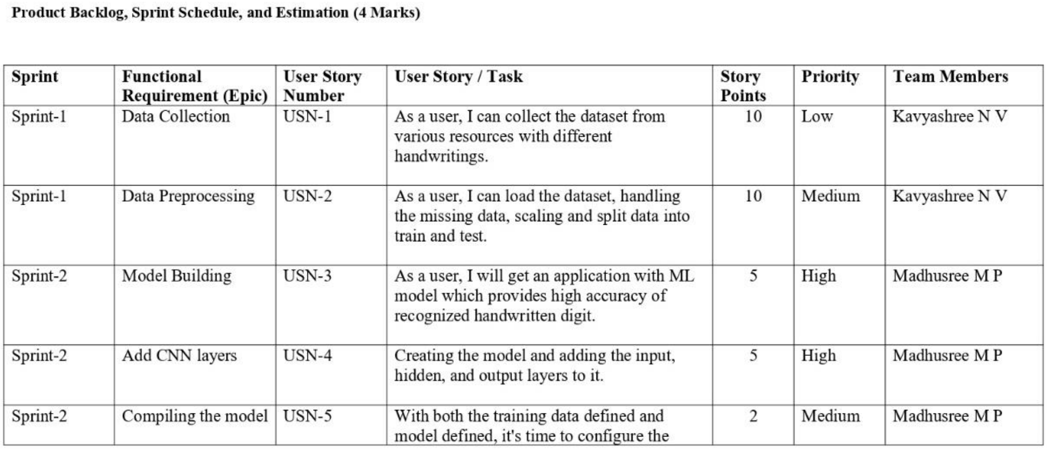
User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Administrator	Data Collection	USN-1	As an Administrator, I can collect the dataset from various resources with different handwritings.	I can collect the data required	Low	Sprint-1
Administrator	Data Preprocessing	USN-2	As an Administrator, I can load the dataset, handling the missing data, scaling and split data into train and test.	I can load and process the collected data.	Medium	Sprint-1
Administrator	Model Building	USN-3	As an Administrator, I will get an application with ML model which provides high accuracy of recognized handwritten digit.	Created an ml model	High	Sprint-2
Administrator	Add CNN layers	USN-4	Creating the model and adding the input, hidden, and output layers to it.	I can add all the cnn layers	High	Sprint-2
Administrator	Compiling the model	USN-5	With both the training data defined and model defined, it's time to configure the learning process.	The ml model is compiled.	Medium	Sprint-2
Administrator	Train & test the model	USN-6	As an Administrator, let us train our model with our image dataset	I can train and test the model built.	Medium	Sprint-2
Administrator	Save the model	USN-7	As an Administrator, the model is saved & integrated with an android application or web application in order to predict something.	Saved the ml model.	Low	Sprint-2
Normal User	Building UI Application	USN-8	As a user, I will upload the handwritten digit image to the application by clicking a upload button.	I can upload the image from which digit has to be recognized.	High	Sprint-3



Normal User		USN-9	As a user, I can know the details of the fundamental usage of the application.	I can understand the usage of the application	Low	Sprint-3
Normal User		USN-10	As a user, I can see the predicted / recognized digits in the application	Recognize and get the output	Medium	Sprint-3
Administrator	Train the model on IBM	USN-11	As an Administrator, I train the model on IBM and integrate flask/Django with scoring end point.	Register and train the model on IBM	High	Sprint-4
Administrator	Cloud Deployment	USN-12	As an Administrator, I can access the web application and make the use of the product from anywhere	Deployed the application on IBM cloud.	High	Sprint-4

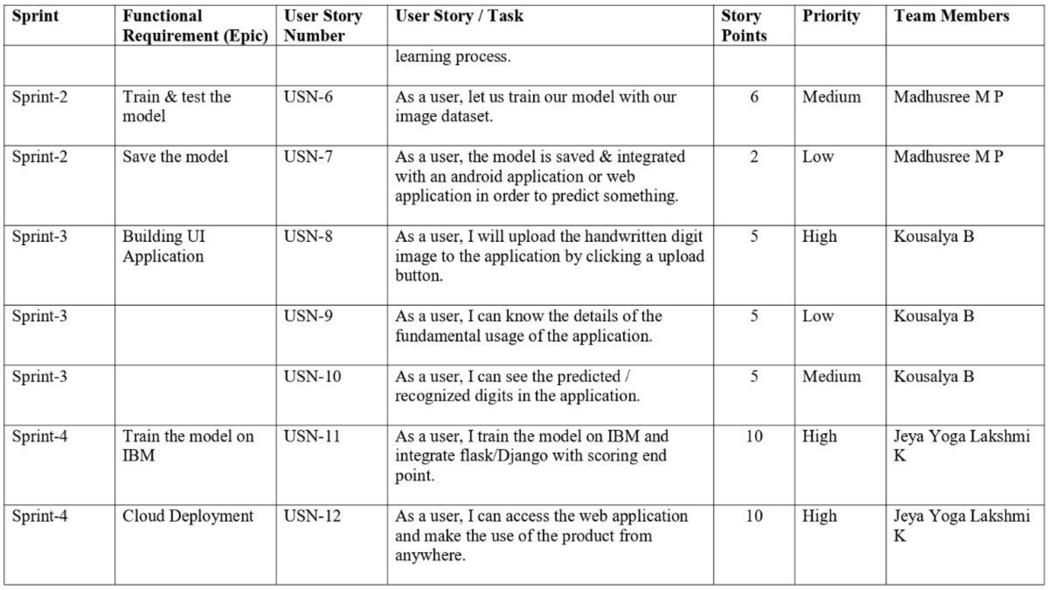
6. Project Planning and Scheduling

6.1 Sprint Planning & Estimation



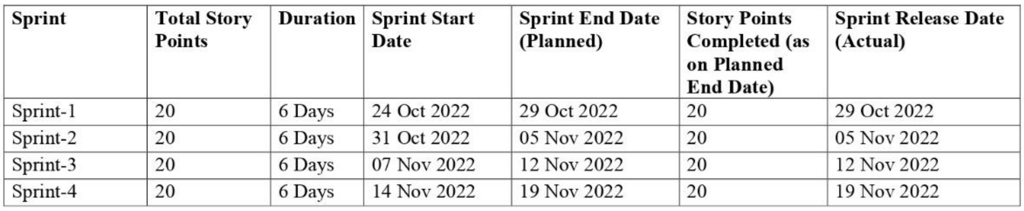
Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data Collection	USN-1	As a user, I can collect the dataset from various resources with different handwritings.	10	Low	Kavyashree N V
Sprint-1	Data Preprocessing	USN-2	As a user, I can load the dataset, handling the missing data, scaling and split data into train and test.	10	Medium	Kavyashree N V
Sprint-2	Model Building	USN-3	As a user, I will get an application with ML model which provides high accuracy of recognized handwritten digit.	5	High	Madhusree M P
Sprint-2	Add CNN layers	USN-4	Creating the model and adding the input, hidden, and output layers to it.	5	High	Madhusree M P
Sprint-2	Compiling the model	USN-5	With both the training data defined and model defined, it's time to configure the	2	Medium	Madhusree M P



Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
			learning process.			
Sprint-2	Train & test the model	USN-6	As a user, let us train our model with our image dataset.	6	Medium	Madhusree M P
Sprint-2	Save the model	USN-7	As a user, the model is saved & integrated with an android application or web application in order to predict something.	2	Low	Madhusree M P
Sprint-3	Building UI Application	USN-8	As a user, I will upload the handwritten digit image to the application by clicking a upload button.	5	High	Kousalya B
Sprint-3		USN-9	As a user, I can know the details of the fundamental usage of the application.	5	Low	Kousalya B
Sprint-3		USN-10	As a user, I can see the predicted / recognized digits in the application.	5	Medium	Kousalya B
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 point.	10	High	Jeya Yoga Lakshmi K
Sprint-4	Cloud Deployment	USN-12	As a user, I can access the web application and make the use of the product from anywhere.	10	High	Jeya Yoga Lakshmi K

6.2 Sprint Delivery Schedule

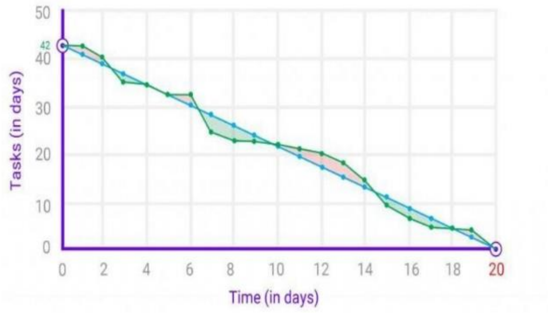


Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022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

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 and Solutioning

7.1 Feature-1 Model Building

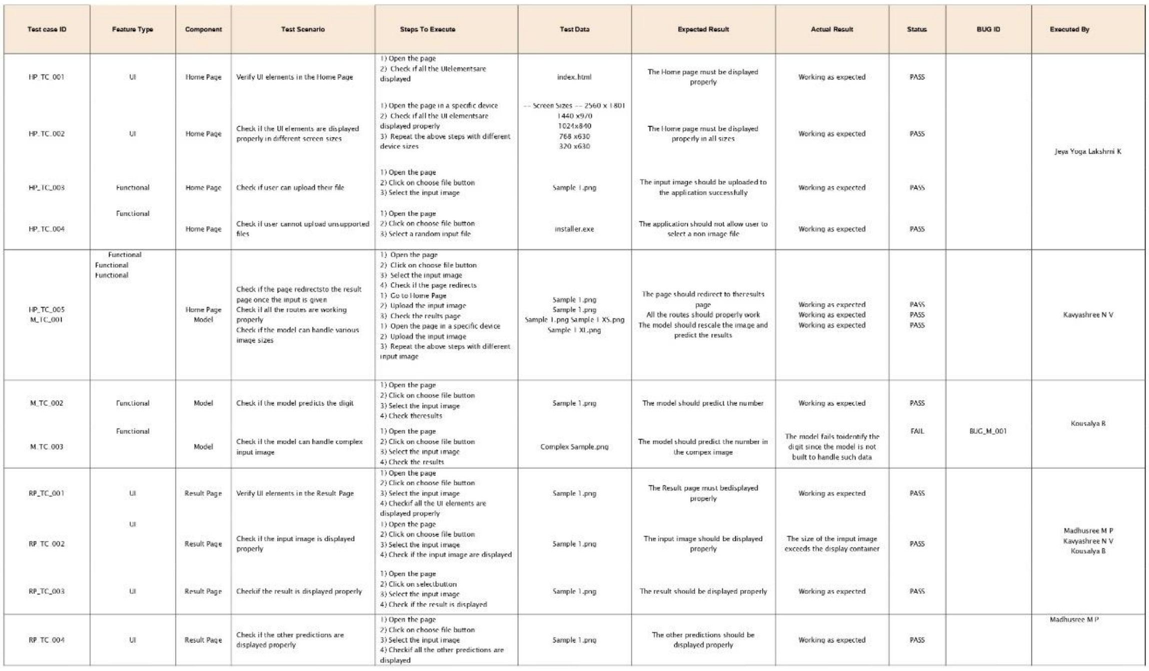
ML depends heavily on data, without data, it is impossible for a machine to learn. It is the most crucial aspect that makes algorithm training possible. In Machine Learning projects, we need a training data set. It is the actual data set used to train the model for performing various actions. TensorFlow already has a MNIST Data set so there is no need to explicitly download or create Dataset. The MNIST dataset contains ten classes: Digits from 0-9. Each digit is taken as a class. The required libraries are imported which are required for the model to run. The dataset for this model is imported from the Keras module. The data is split into train and test. Using the training dataset, the model is trained and the testing dataset is used to predict the results. Basically, the pixel values range from 0-255. The value of each image is stored is y_train. The model is built with convolutional, pooling and dense layers. The created model is then compiled and saved.

7.2 Feature-2 Web App

HTML, CSS and JavaScript are used to create the web pages for the front end. An html page that takes in image files as input using form and submits to the back end is created. A flask app is created using python flask, where it receives the image files from the templates, html pages and the prediction operation is done over this image. Later the predicted output is sent to the result page.

8.Testing

8.1. Test Cases

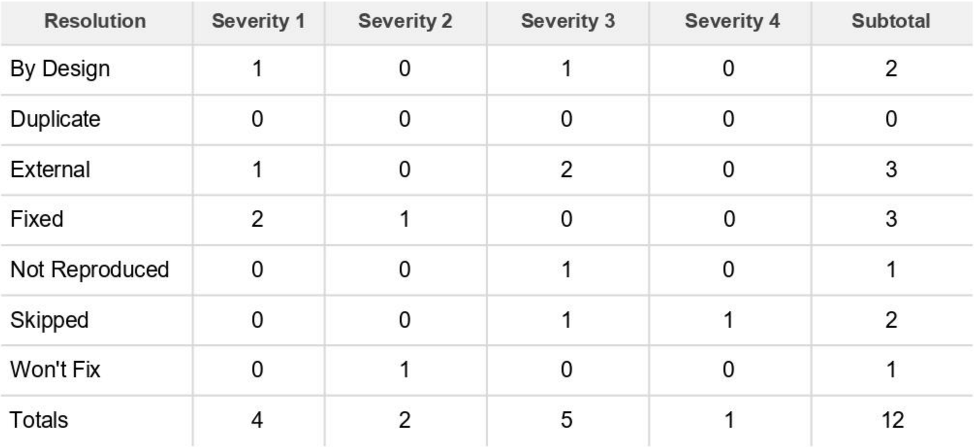


Test case ID	Feature Type	Component	Test Scenario	Steps To Execute	Test Data	Expected Result	Actual Result	Status	BUG ID	Executed By
HP_TC_001	UI	Home Page	Verify UI elements in the Home Page	1) Open the page 2) Check if all the Uielementsare displayed	index.html	The Home page must be displayed properly	Working as expected	PASS		Jeya Yoga Lakshmi K
HP_TC_002	UI	Home Page	Check if the UI elements are displayed properly in different screen sizes	1) Open the page in a specific device 2) Check if all the UI elementsare displayed properly 3) Repeat the above steps with different device sizes	-- Screen Sizes -- 2560 x 1801 1440 x970 1024x840 768 x630 320 x630	The Home page must be displayed properly in all sizes	Working as expected	PASS		
HP_TC_003	Functional	Home Page	Check if user can upload their file	1) Open the page 2) Click on choose file button 3) Select the input image	Sample 1.png	The input image should be uploaded to the application successfully	Working as expected	PASS		
HP_TC_004	Functional	Home Page	Check if user cannot upload unsupported files	1) Open the page 2) Click on choose file button 3) Select a random input file	installer.exe	The application should not allow user to select a non image file	Working as expected	PASS		
HP_TC_005 M_TC_001	Functional Functional Functional	Home Page Model	Check if the page redirectsto the result page once the input is given Check if all the routes are working properly Check if the model can handle various image sizes	1) Open the page 2) Click on choose file button 3) Select the input image 4) Check if the page redirects 1) Go to Home Page 2) Upload the input image 3) Check the reults page 1) Open the page in a specific device 2) Upload the input image 3) Repeat the above steps with different input image	Sample 1.png Sample 1.png Sample 1.png Sample 1 XS.png Sample 1 XL.png	The page should redirect to theresults page All the routes should properly work The model should rescale the image and predict the results	Working as expected Working as expected Working as expected	PASS PASS PASS		Kavyashree N V
M_TC_002	Functional	Model	Check if the model predicts the digit	1) Open the page 2) Click on choose file button 3) Select the input image 4) Check theresults	Sample 1.png	The model should predict the number	Working as expected	PASS		Kousalya B
M_TC_003	Functional	Model	Check if the model can handle complex input image	1) Open the page 2) Click on choose file button 3) Select the input image 4) Check the results	Complex Sample.png	The model should predict the number in the compex image	The model fails toidentify the digit since the model is not built to handle such data	FAIL	BUG_M_001	
RP_TC_001	UI	Result Page	Verify UI elements in the Result Page	1) Open the page 2) Click on choose file button 3) Select the input image 4) Checkif all the UI elements are displayed properly	Sample 1.png	The Result page must bedisplayed properly	Working as expected	PASS		Madhusree M P Kavyashree N V Kousalya B
RP_TC_002	UI	Result Page	Check if the input image is displayed properly	1) Open the page 2) Click on choose file button 3) Select the input image 4) Check if the input image are displayed	Sample 1.png	The input image should be displayed properly	The size of the imput image exceeds the display container	PASS		
RP_TC_003	UI	Result Page	Checkif the result is displayed properly	1) Open the page 2) Click on selectbutton 3) Select the input image 4) Check if the result is displayed	Sample 1.png	The result should be displayed properly	Working as expected	PASS		
RP_TC_004	UI	Result Page	Check if the other predictions are displayed properly	1) Open the page 2) Click on choose file button 3) Select the input image 4) Checkif all the other predictions are displayed	Sample 1.png	The other predictions should be displayed properly	Working as expected	PASS		Madhusree M P

8.2. User Acceptance Testing

Defect Analysis

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



Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	1	0	1	0	2
Duplicate	0	0	0	0	0
External	1	0	2	0	3
Fixed	2	1	0	0	3
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	1	0	0	1
Totals	4	2	5	1	12

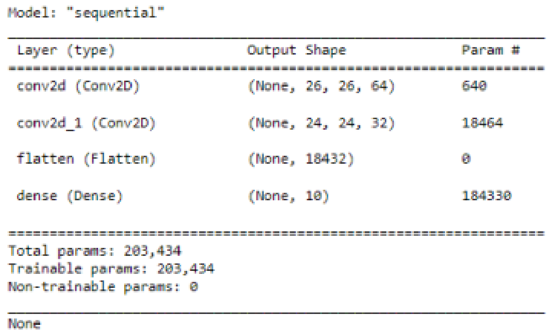
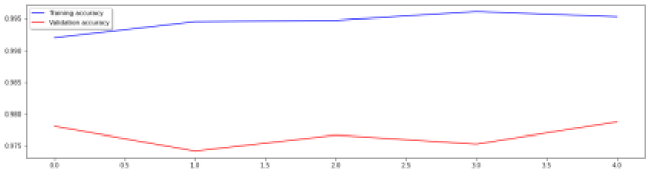
Test Case Analysis

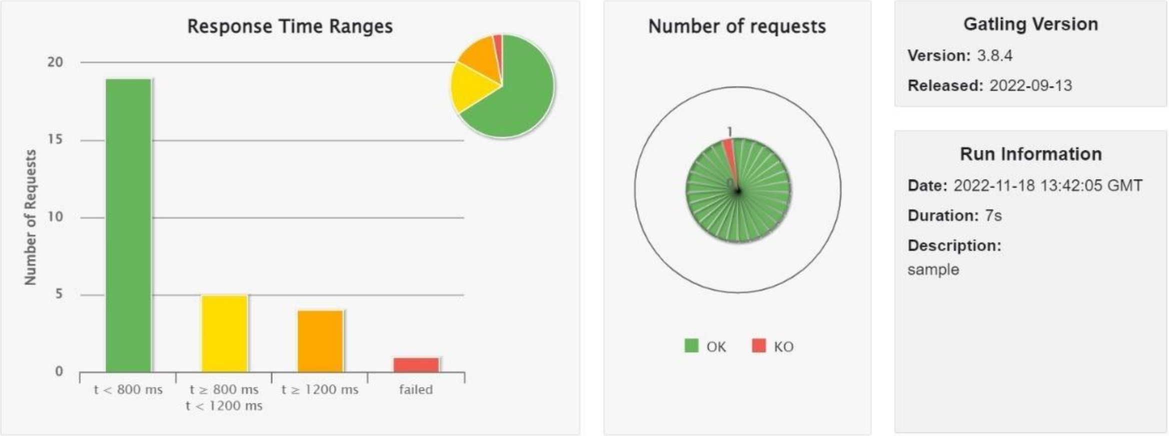
This report shows the number of test cases that have passed, failed, and untested.

Section	Total Cases	Not Tested	Fail	Pass
Client Application	7	0	1	6
Security	2	0	1	1
Performance	2	0	0	2
Exception Reporting	3	0	0	3

9.Results

9.1 Performance Metrics

S.No.	Parameter	Values	Screenshot
1.	Model Summary	- Model: "sequential_1"	 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 None
2.	Accuracy	Training Accuracy - 0.9805333614349365 Validation Accuracy – 0.9735000133514404	



Gatling Version

Version: 3.8.4

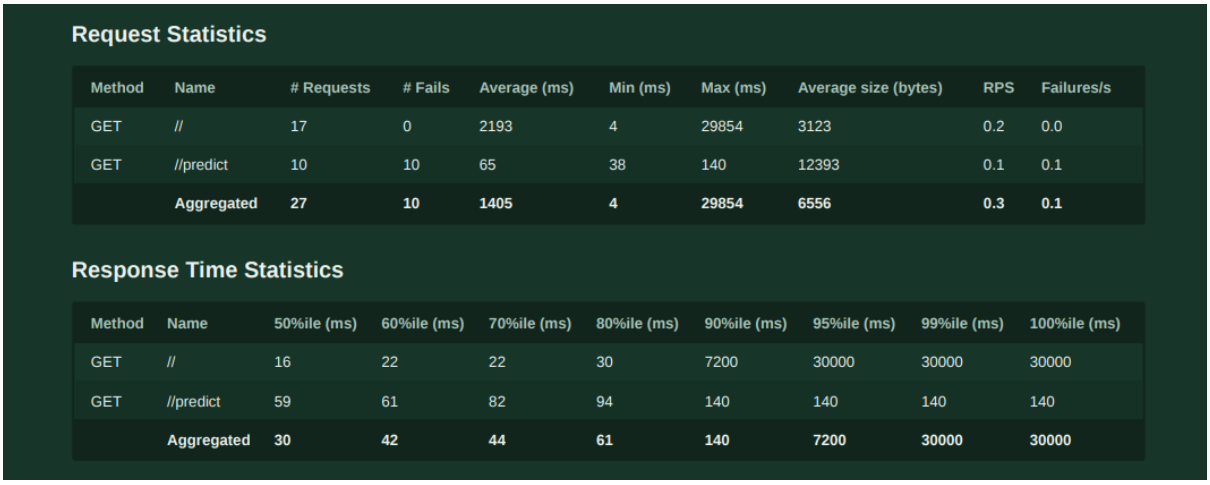
Released: 2022-09-13

Run Information

Date: 2022-11-18 13:42:05 GMT

Duration: 7s

Description:
sample



Request Statistics

Method	Name	# Requests	# Fails	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	RPS	Failures/s
GET	//	17	0	2193	4	29854	3123	0.2	0.0
GET	//predict	10	10	65	38	140	12393	0.1	0.1
	Aggregated	**27**	**10**	**1405**	**4**	**29854**	**6556**	**0.3**	**0.1**

Response Time Statistics

Method	Name	50%ile (ms)	60%ile (ms)	70%ile (ms)	80%ile (ms)	90%ile (ms)	95%ile (ms)	99%ile (ms)	100%ile (ms)
GET	//	16	22	22	30	7200	30000	30000	30000
GET	//predict	59	61	82	94	140	140	140	140
	Aggregated	**30**	**42**	**44**	**61**	**140**	**7200**	**30000**	**30000**

10.Advantages and Disadvantages

Advantages

- This can be used for sorting through mail by postal code.
- It reduces Manual work.
- Elimination of the need for feature engineering.
- Elimination of the cost required for data labelling.

Disadvantages

- It requires very large amount of data in order to perform better than other techniques.
- The handwritten digits are not always of the same size, width, orientation and justified to margins as they differ from writing of person to person.
- Problem is faced more when many people write a single digit with a variety of different handwritings.
- All the data must be in Digital format.

11. Conclusion

Handwritten digit recognition has immense applications in the field of medical, banking, student management, and taxation process etc. Many classifiers like KNN, SVM, and CNN are used to identify the digit from the handwritten image. Here we've used CNN for implementation. Convolutional Neural Network gets trained from the real-time data and makes the model very simple by reducing the number of variables and gives relevant accuracy.

MNIST dataset consist of handwritten numbers from 0-9 and it is a standard dataset used to find performance of classifiers. Results of HDR is improved a lot by using CNN classifier but it can be improved further in terms of complexity, duration of execution and accuracy of results by making combination of classifiers or using some additional algorithm with it. More accurate results can be established with more convolution layers and more number of hidden neurons. It can completely abolish the need for typing. Digit recognition is an excellent prototype problem for learning about neural networks and it gives a great way to develop more advanced techniques of deep learning.

12. Future Scope

In future, different architectures of CNN, namely, hybrid CNN, viz., CNN-RNN and CNN-HMM models, and domain-specific recognition systems, can be investigated. Evolutionary algorithms can be explored for optimizing CNN learning parameters, namely, the number of layers, learning rate and kernel sizes of convolutional filters.

The future development of the applications based on algorithms of deep and machine learning is practically boundless. In the future, we can work on a denser or hybrid algorithm than the current set of algorithms with more manifold data to achieve the solutions to many problems.

In future, the application of these algorithms lies from the public to high-level authorities, as from the differentiation of the algorithms above and with future development we can attain high-level functioning applications which can be used in the classified or government agencies as well as for the common people. Currently only the digits are recognized. In future the all the characters in all the language can be predicted with high accuracy rate.

13.Appendix

Source code

Front end – Html code
Index.html

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

```
 <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"
integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM"
crossorigin="anonymous"></script>
 <script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@latest"></script>
</head>
<script>
 function preview() {
  frame.src=URL.createObjectURL(event.target.files[0]);
}
  $(document).ready(function() {
     $('#clear_button').on('click', function() {
       $('#image').val('');
       $('#frame').attr('src',"");
      });
    });

</script>
<body>
 <h1 class="welcome">Handwritten Digit Recognition Application <br>
 <div id="team_id">TEAM ID : PNT2022TMID24646</div>
 </h1>
 <section id="title">
  <br><br>
   <p>
    A Novel Method for Handwritten Digit Recognition System.
   </p>
   <p>This is a very useful application as it allows you to quickly jot down numbers for
contacts,postal mail sorting, bank check processing, form data entry, etc.</p>
 </section>
 <section id="content">
    <div class="leftside">
    <form action="/predict" method="POST" enctype="multipart/form-data">
```

```
        <label style="color: #000;">Select a image:</label>
        <input id="image" type="file" name="image" accept="image/png, image/jpeg"
onchange="preview()"><br><br>
         <img id="frame" src="" width="100px" height="100px"/>
         <div class="buttons_div">
           <button type="submit" class="btn btn-dark" id="predict_button">Predict</button>
           <button type="button" class="btn btn-dark" id="clear_button">&nbsp Clear &nbsp</button>
         </div>
        </form>
        </div>
        <br>
  </section>
</body>
</html>
```

Style.css

```
#clear_button{
  margin-left: 15px;
  font-weight: bold;
  color: #153462;
}
#confidence{
  font-family: 'Josefin Sans', sans-serif;
  margin-top: 7.5%;
}
#content{
  margin: 0 auto;
  padding: 2% 15%;
  padding-bottom: 0;
  background-color: #4FA095;
}
.welcome{
   text-align: center;
   position: relative;
   color: #153462;
   background-color: #F6F6C9;
   padding-top: 1%;
   font-weight: bold;
   font-family: 'Prompt', sans-serif;
}
#team_id{
   text-align: right;
   font-size: 25px;
   padding-right: 3%;
}
#predict_button{
  margin-right: 15px;
  color: #153462;
  font-weight: bold;
}
#prediction_heading{
  font-family: 'Josefin Sans', sans-serif;
  margin-top: 7.5%;
}
#result{
  font-size: 5rem;
}
#title{
  padding: 1.5% 15%;
  margin: 0 auto;
  text-align: center;
  background-color: #BAD1C2;
}
```

```
.btn {
   font-size: 15px;
   padding: 10px;
   -webkit-appearance: none;
   background: #eee;
   border: 1px solid #888;
   margin-top: 20px;
   margin-bottom: 20px;
   border-radius: 25px;
   color: #4FA095;
}
.buttons_div{
  margin-bottom: 30px;
  margin-right: 80px;
}
.heading{
  font-family: 'Varela Round', sans-serif;
  font-weight: 700;
  font-size: 2rem;
  display: inline;
}
.leftside{
  text-align: center;
  margin: 0 auto;
  margin-top: 2%;
  /* padding-left: 10%; */
}
#frame{
  margin-right: 10%;
}
.predicted_answer{
  text-align: center;
  margin: 0 auto;
  padding: 3% 5%;
  padding-top: 0;
  /* padding-left: 10%; */
}
p{
  font-family: 'Source Code Pro', monospace,sans-serif;
  margin-top: 1%;
  color: #153462;
}
@media (min-width: 720px) {
  .leftside{
    padding-left: 10%;
  }
}
```

Predict.html

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

<style>
    body{
     background-color: #BAD1C2;
    }

    #rectangle{
     width:500px;
     height:120px;
     background-color: #4FA095;
     border-radius: 25px;
     position:absolute;
     top:25%;
     left:50%;
     transform:translate(-50%,50%);
    }

    #ans{
  text-align: center;
  font-size: 40px;
  margin: 0 auto;
  padding: 3% 5%;
  padding-top: 8%;
  color: #000;
    }

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

Backend – Python code

App.py

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

UPLOAD_FOLDER = 'C:/Users/jeyay/OneDrive/Desktop/Final Deliverables/Final code/app/uploads'

app = Flask(__name__)
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER
model = load_model("mnistCNN.h5")

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

@app.route('/predict', methods=['GET', 'POST'])
def upload():
    if request.method == "POST":
        f = request.files["image"]
        filepath = secure_filename(f.filename)
        f.save(os.path.join(app.config['UPLOAD_FOLDER'], filepath))
        upload_img = os.path.join(UPLOAD_FOLDER, filepath)
        img = Image.open(upload_img).convert("L")  # convert image to monochrome
        img = img.resize((28, 28))  # resizing of input image
        im2arr = np.array(img)  # converting to image
        im2arr = im2arr.reshape(1, 28, 28, 1)  # reshaping according to our requirement
        pred = model.predict(im2arr)
        num = np.argmax(pred, axis=1)  # printing our Labels
        return render_template('predict.html', num=str(num[0]))
```

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

Model Creation

```
import numpy as np

import tensorflow #open source used for both ML and DL for computation

from tensorflow.keras.datasets import mnist #mnist dataset

from tensorflow.keras.models import Sequential #it is a plain stack of layers

from tensorflow.keras import layers #A Layer consists of a tensor- in tensor-out computat ion funct
ion

from tensorflow.keras.layers import Dense, Flatten #Dense-Dense Layer is the regular deeply
connected r

from tensorflow.keras.layers import Conv2D #convolutional Layer

from keras.utils import np_utils #used for one-hot encoding

import matplotlib.pyplot as plt   #used for data visualization

(x_train, y_train), (x_test, y_test)=mnist.load_data ()

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'])

x_train = np.asarray(x_train)

y_train = np.asarray(y_train)

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

metrics = model.evaluate(x_test, y_test, verbose=0)

print("Metrics (Test loss &Test Accuracy) : ")

print(metrics)

prediction=model.predict(x_test[6000:6001])

print(prediction)

np.argmax(y_test[6000:6001])

model.save('models/mnistCNN.h5')
```

Github Link:

https://github.com/IBM-EPBL/IBM-Project-50248-1660901034

Project Demo Video Link:

https://drive.google.com/drive/folders/1VRIjRjWzU6BXfy2-AbF5_9GGNUIh3__n?usp=sharing