PROJECT REPORT

A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

Submitted By

PNT2022TMID31528

Manisha E - 710719205029

Sivagama sundari M - 710719205047

Rajasri M - 710719205039

Britney Sandra M - 710719205011

Abinaya P - 710719205002

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Project Overview

Machine learning is a branch of Artificial Intelligence and computer science which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy.

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.

1.2 Purpose

Hand writing recognition of characters has been around since the 1980s.The task of handwritten digit recognition, using a classifier, has great importance and use such as – online handwriting recognition on computer tablets, recognize zip codes on mail for postal mail sorting, processing bank check amounts, numeric entries in forms filled up by hand (for example - tax forms) and so on.

2. LITERATURE SURVEY

2.1 Existing problem

The different architectures of CNN, hybrid CNN,CNN - RNN and CNNHMM models, and domain - specific recognition system, are not thoroughly inquired and evolutionary algorithms are not clearly explored for optimizing CNN learning

parameters ,the number of layers, learning rate and kernel sizes of convolutional filters.

. The fluctuation of accuracies for handwritten digits was observed for 15 epochs by varying the hidden layers. There is no clear explanation given for observing variation in the overall classification accuracy by varying the number of hidden layers and batch size

2.2 References

S.NO	Author Name	Paper Title	Journal/ Conference title	Page No/ Volume No	Year of Publication	Description

1.	Amit Choudhary, Anand Nayyar, Saurabh Singh and Byungun Yoon.	Improved Handwritten Digit Recognition Using Convolutional Neural Networks (CNN)	IEEE Sensors Journal		2021	In this paper, with the aim of improving the performance of handwritten digit recognition, they valuated variants of a convolutional neural network to avoid complex preprocessing, costly feature extraction and a complex ensemble (classifier combination) approach
						of a traditional recognition system.

2.	Rudraswamimath, Bhavanishankar and Channasandra.	Handwritten Digit Recognition using CNN	International Journal of Innovative Science and Research Technology	Volume -4 Issue6	2018	In this paper, the most widely used Machine learning algorithms, KNN, SVM, RFC and CNN have been trained and tested on the same data in order acquire the comparison between the classifiers
3.	, Shadman Sakib and Md. Abu Bakr Siddique.	Recognition of Handwritten Digit using Convolutional Neural Network in Python with Tensorflow and Comparison of Performance for Various Hidden Layers	5th International Conference on Advances in Electrical Engineering (ICAEE)		2017	In this paper, they observed the variation of accuracies of CNN to classify handwritten digits for 15 epochs using various numbers of hidden layers and epochs and

						to make the comparison between the accuracies. For this performance evaluation of CNN, they performed the experiment using Modified National Institute of Standards and Technology(MN IST) dataset.

4.	Gupta, Ravindra Pratap Narwaria and Madhav Singh	Review on Deep Learning Handwritten Digit Recognition using Convolutional Neural Network	International Journal of Recent Technology and Engineering (IJRTE)	Volume -9 Issue5	2022	In this paper, Object Character Recognition (OCR) is used on printed or documented letters to convert them into text. The database has training image database of 60,000 images and
						testing image database of 10,000 images. The KNN algorithm describes categorical value by making use of majority of votes of K - nearest neighbors, the K value used to differ here.

5.	Anwar Hossain and Md. Mohon Ali	Recognition of Handwritten Digit using Convolutiona l Neural Network (CNN)	Global Journal of Computer Science and Technology: D Neural & Artificial Intelligence	Volume 19 Issue2	2020	The goal of this work will be to create a model that will be able to identify and determine the handwritten digit from its image with better accuracy using using the concepts of Convolution al Neural Network and MNIST

						dataset. Later it can be extended for character recognition and realtime person's handwriting. The results can be made more accurate with more convolution layers and more number of hidden neurons.

2.3 Problem Statement Definition

Handwriting recognition is one of the compelling research works going on because every individual in this world has their own style of writing. It is the capability of the computer to identify and understand handwritten digits or characters automatically. Because of the progress in the field of science and technology, everything is being digitalized to reduce human effort.

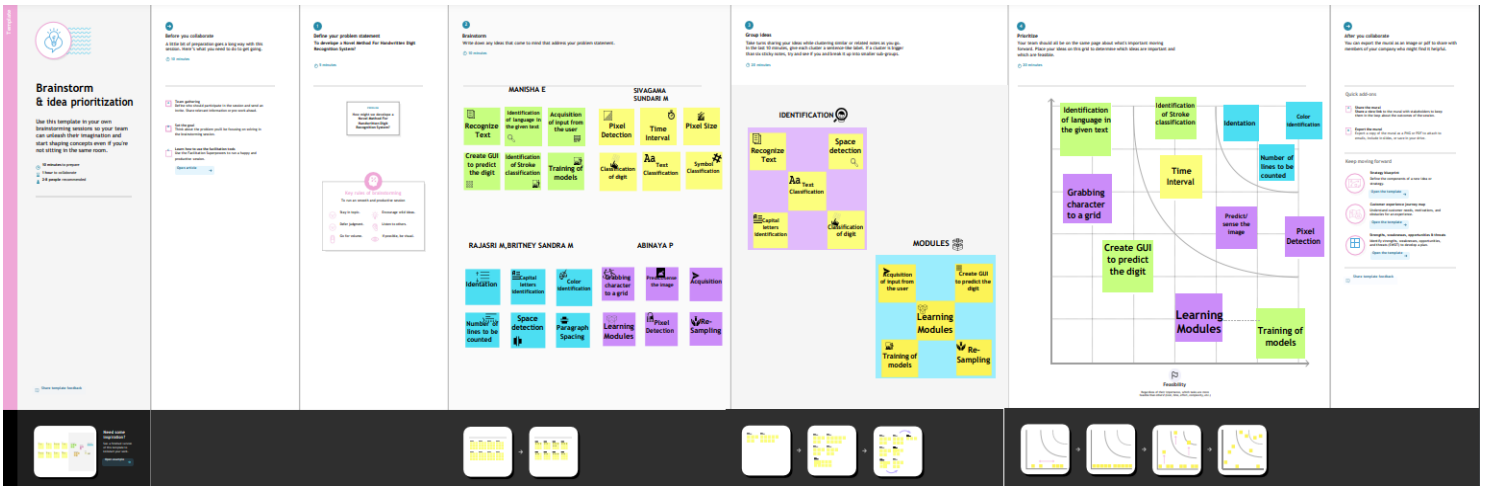
Hence, there comes a need for handwritten digit recognition in many real time applications. MNIST data set is widely used for this recognition process and it has 70000 handwritten digits. We use Artificial neural networks to train these images and build a deep learning model. Web application is created where the user can upload an image of a

handwritten digit. this image is analyzed by the model and the detected result is returned on to UI(User Interface).

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

3.2 Ideation & Brainstorming



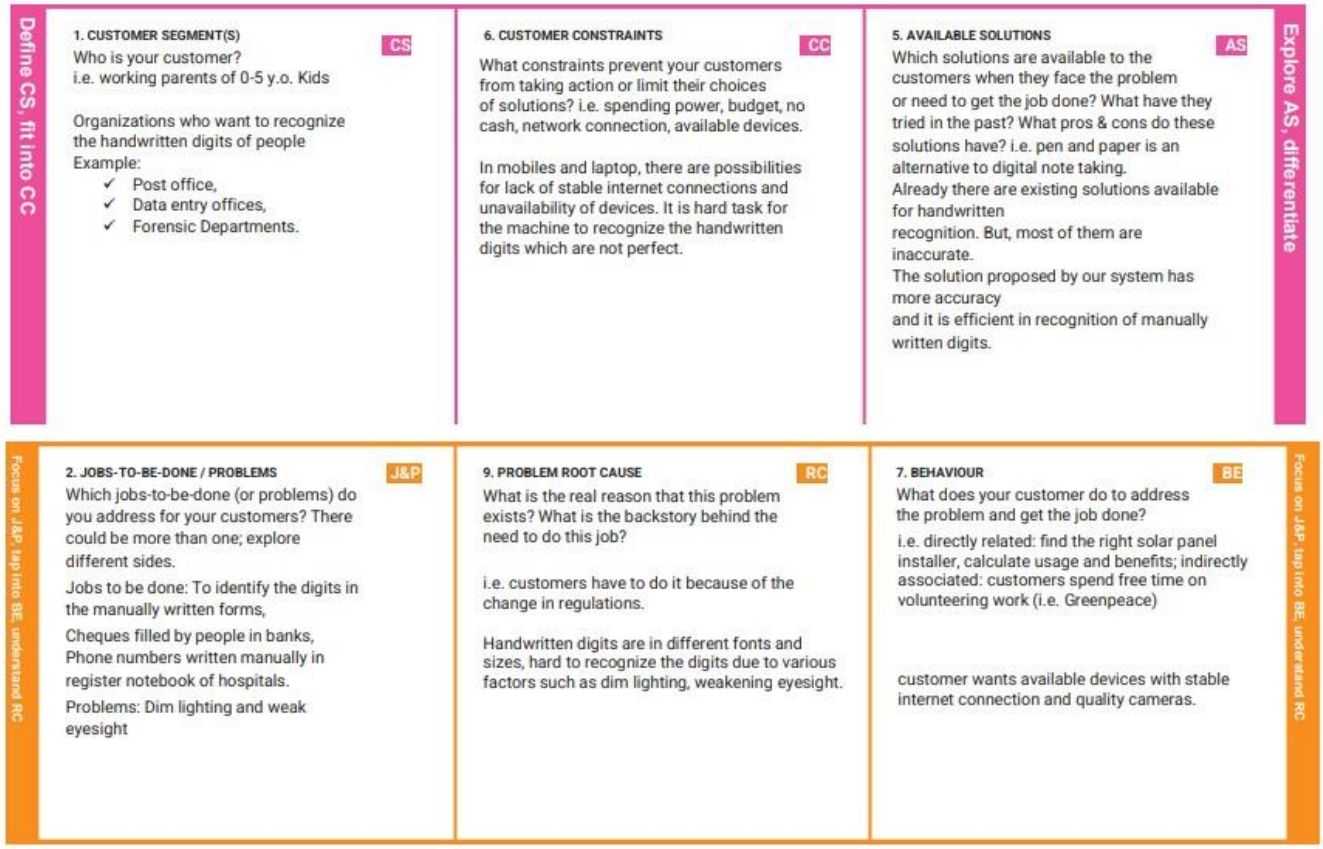
3.3 Proposed Solution

S.No.	Parameter	Description
1	Problem Statement (Problem to be solved)	Statement-The handwritten digit recognition is the capability of computer applications to recognize the human handwritten digits. Description: It is a hard task for the machine because handwritten digits are not perfect and can be made with many different shapes and sizes.
2	Idea / Solution description	1. It is the capability of a computer to fete the mortal handwritten integers from different sources like images, papers, touch defences. 2. It allows user to translate all those signature and notes into electronic words in a text document format and this data only requires far less physical space than the storage of the physical copies.

3	Novelty / Uniqueness	Accurately recognize the digits rather than recognizing all the characters like OCR Handwritten Digit Recognition is the capability of a computer to fete the mortal handwritten integers from different sources like images, papers,touch defenses, etc. And classify them into 10 predefined classes(09).This is the existing method along with this we add some features to make our project unique among them.
4	Social Impact / Customer Satisfaction	1.Artificial Intelligence developed the app called Handwritten digit Recognizer. 2. It converts the written word into digital approximations and utilizes complex algorithms to identify characters before churning out a digital approximation.
5	Business Model (Revenue Model)	• This system can be integrated with traffic surveillance cameras to recognize the vehicle's number plates for effective traffic management. • Can be integrated with Postal system to identify and recognize the pin-code details easily.

		necessarily because of sloppy handwriting, but because people write all over the envelope.We have hard time segmenting handwritten addresses into their components, such as ZIP code or street address, because very few people print addresses neatly in a prescribed format. So, this problem can be solved using Handwritten digit recognition system.
6	Scalability of the Solution	In our model, AlexNet significantly outperformed as it is trained on a GTX 580 GPU with only 3 GB of memory which couldn't fit the entire network. So the network was split across 2 GPUs, with half of the neurons(feature maps) on each GPU. So, a greater accuracy can be attained by allowing multi-GPU training by putting half of the model's neurons on one GPU and the other half on another GPU.

3.4 Problem Solution fit



Identify strong TR & EM

3. TRIGGERS TR

What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news.

Advertisement in the market about the efficient recognition of digits.

Articles about the achievements made by our project.

4. EMOTIONS: BEFORE / AFTER EM

How do customers feel when they face a problem or a job and afterwards?
i.e. lost, insecure > confident, in control - use it in your communication strategy & design.

Defects are common and our project is not an exception

When the system failed to recognize the digit,

Customer Mentality:
Before:(Failure)
We would give guarantee that it would work most of the time
and if any error occurs, they can contact us at any time.
So, customers can feel at ease.
After:(Failure)
They have no need to panic when the failure occurs
They can easily contact us to rectify the error.
We would solve the defect as soon as possible.

10. YOUR SOLUTION SL

If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality.
If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.

Our solution aims to recognize handwritten digits using machine learning techniques thereby saving costs to the organization improving employee productivity.

In our model we use AlexNet , which is one of the CNN architectures . AlexNet allows for multi-GPU training by putting half of the model's neurons on one GPU and the other half on another GPU. Not only does this mean that a bigger model can be trained, but it also cuts down on the training time. It also reduces the overfitting problem by Data Augmentation and Dropout.

8. CHANNELS of BEHAVIOUR CH

8.1 ONLINE

What kind of actions do customers take online?
Extract online channels from #7

Requires Stable internet connection for image processing.

8.2 OFFLINE

What kind of actions do customers take offline?
Extract offline channels from #7 and use them for customer development.

Obtain modern electronic devices and check they are working

Identify strong TR & EM

4. REQUIREMENT ANALYSIS

4.3 Functional requirement

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.

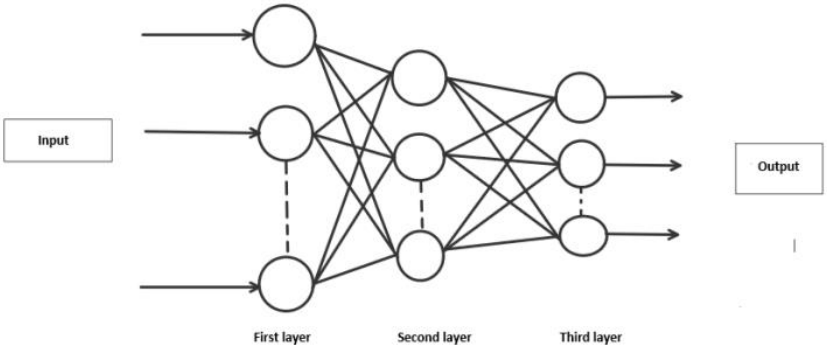
4.4 Non-Functional requirements

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

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Handwritten digit recognition is one of the major important issues in pattern recognition applications. Some of the applications for digit recognition include data entry forms, Bank check processing etc,.
NFR-2	**Security**	The applications of handwritten digit recognition can be used in the banking sector where it can be used to maintain the security pin numbers safely. It can be also used for blind-people by using sound output.
NFR-3	**Reliability**	Reliability indicates the probability that the system will perform its intended function for a larger period of sufficient time and also it will operate in a secured environment without any failures.
NFR-4	**Performance**	The standard implementations of neural networks achieve an accuracy of approximately (98–99)
		percent in correctly classifying the handwritten digits.
NFR-5	**Availability**	The features for handwritten digit recognition have been Acquainted. These features are based on shape analysis of the digit image and extract slant or slope information. They are effective in obtaining good recognition of accuracy.
NFR-6	**Scalability**	The scalability in the task of handwritten digit recognition, using a classifier, has great importance and it makes use of online handwriting recognition on computer tablets, recognizing zip codes on mail for postal mail sorting, processing bank check amounts, numeric entries in forms filled up manually(for example - tax forms) and so on.

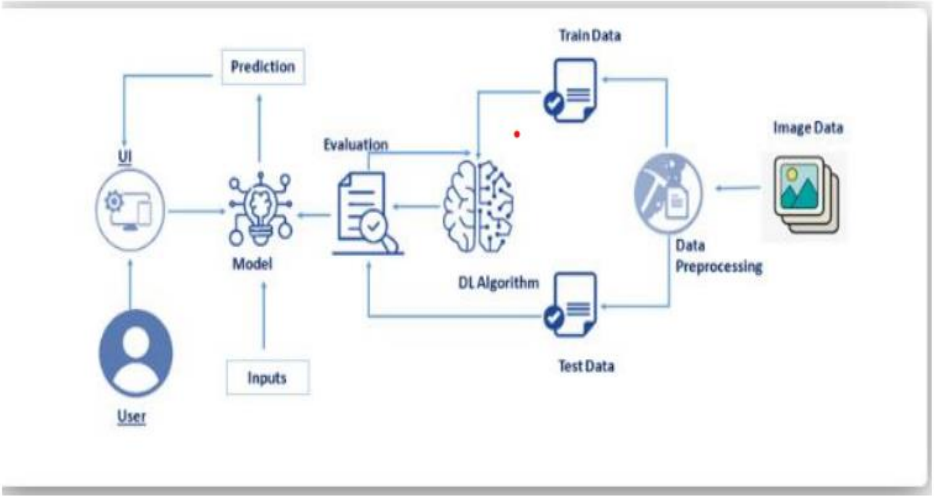
5. PROJECT DESIGN

5.1 Data Flow Diagrams



5.2 Solution & Technical Architecture

Solution Architecture



Technology Architecture

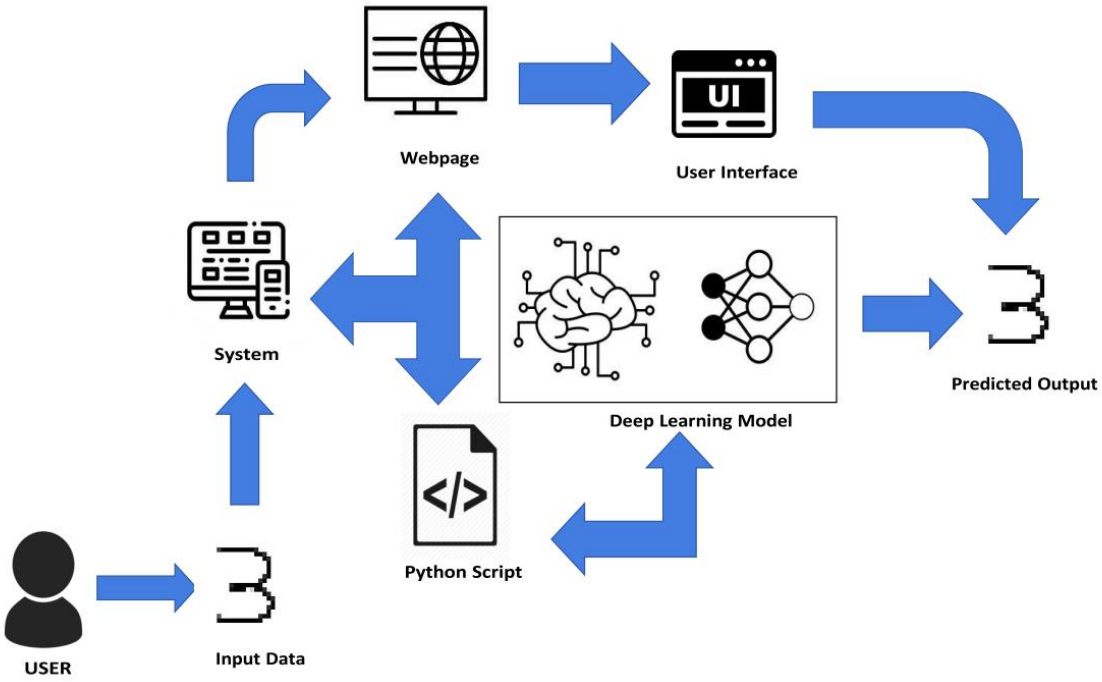


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g. Web UI	HTML, CSS, JavaScript
2.	Application Logic-1	Model is built	Python
3.	Application Logic-2	Python model is deployed	IBM Watson Studio
4.	File Storage	Predicted outputs of the image are stored in a local folder.	Local Filesystem
5.	Machine Learning Model	To predict the image uploaded by the user.	Image Recognition Model
6.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Flask Cloud Server Configuration : IBM Watson Studio	Local, Cloud Foundry.

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used	Flask
2.	Security Implementations	List all the security / access controls implemented, use of firewalls etc.	e.g. SHA-256, Encryptions, IAM Controls, OWASP etc.
3.	Scalable Architecture	High workload can be supported without undergoing any major changes.	Technology used in the architecture is that with Python and the IBM cloud.
4.	Availability	Readily available enables the IT Infrastructure to function when some of the components fail.	Technology used is IBM cloud.
5.	Performance	Performance technology is a field which uses various tools,processes and procedures in a systematic and efficient manner to improve the desired outcomes of individuals and organizations.	Technology used is python.

5.3 User Stories

User Stories

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer	Dashboard	USN-1	As a user, they can see the information regarding the prediction of handwritten digit recognition.	I can see the information regarding digit recognition.	High	Sprint 1
	Launch	USN-2	On clicking the launch button, it will redirect the user to a page where the images to be predicted can be uploaded.	I can see the launch button.	High	Sprint 1
	Upload	USN-3	Users can select the image from the local storage.	I can upload the image.	High	Sprint 2
	Predict	USN-4	Once the image is uploaded, it will predict the respective image.		High	Sprint 3
	Display	USN-5	The predicted image will be displayed with the accuracy chart.	I can see the result with accuracy.	High	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 my password.	2	High	Manisha E
Sprint-1	Login	USN-2	As a user, I can log into the application by entering email & password	1	High	Sivagama sundari M
Sprint-2	Upload Image of digital document	USN-3	As a user, I can able to input the images of digital documents to the application	2	Medium	Rajasri M
Sprint-2	Prediction	USN-4	As a user, I can predict the word	1	Medium	Britney Sandra M
Sprint-2	Prediction	USN-5	As a user,I can predict the word	2	Medium	Abinaya P

6.2 Sprint Delivery Schedule

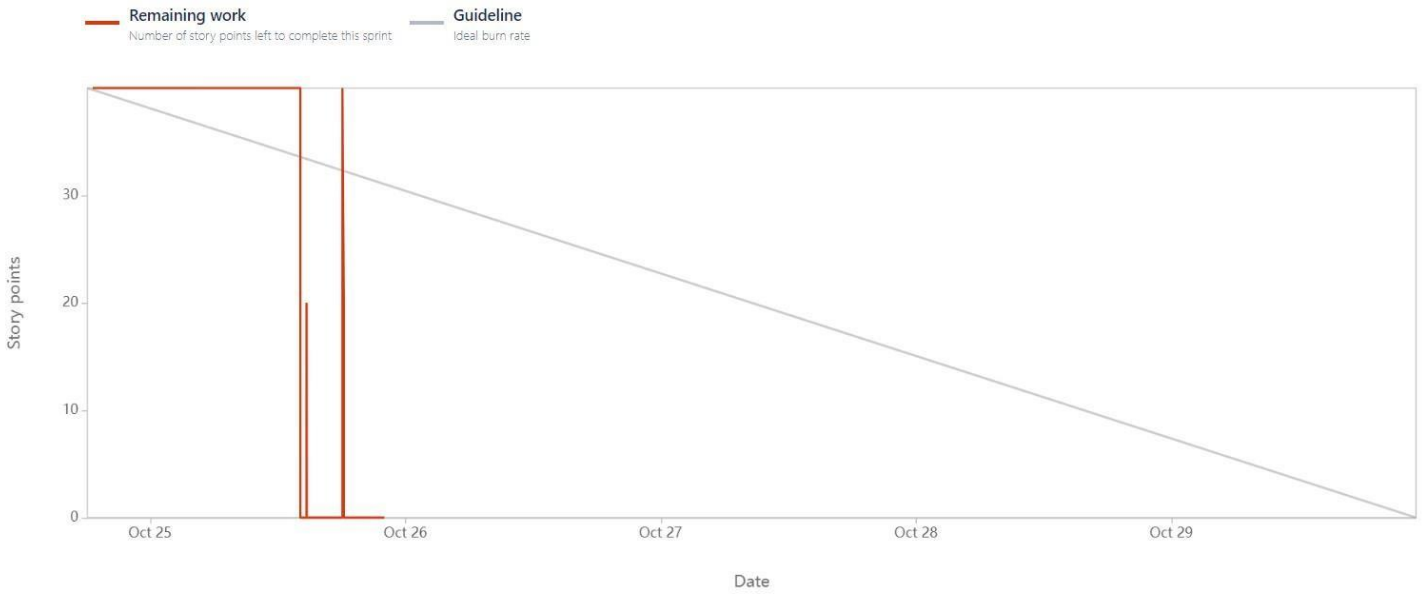
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	2	6 Days	25 Oct 2022	30 Oct 2022	2	30 Oct 2022
Sprint-2	2	6 Days	1 Nov 2022	06 Nov 2022	2	06 Nov 2022
Sprint-3	2	6 Days	08 Nov 2022	13 Nov 2022	2	13 Nov 2022
Sprint-4	2	6 Days	15 Nov 2022	20 Nov 2022	2	20 Nov 2022

6.3 Reports from JIRA

Velocity Report

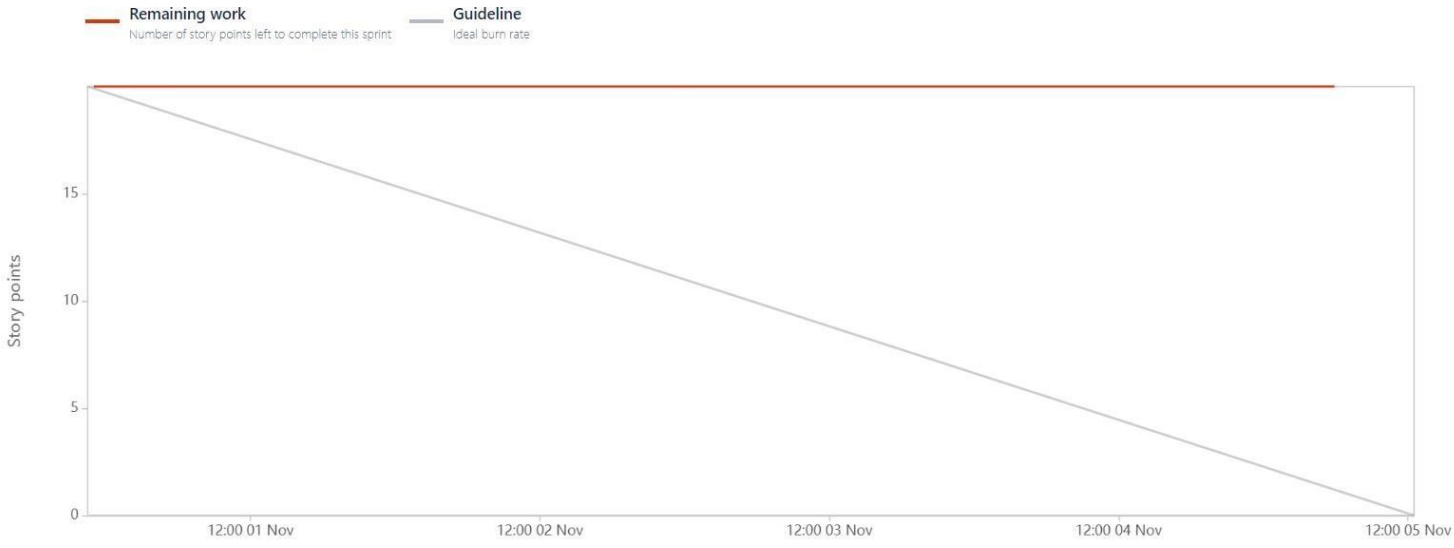
` **Sprint 1**

Date - October 24th, 2022 - October 29th, 2022



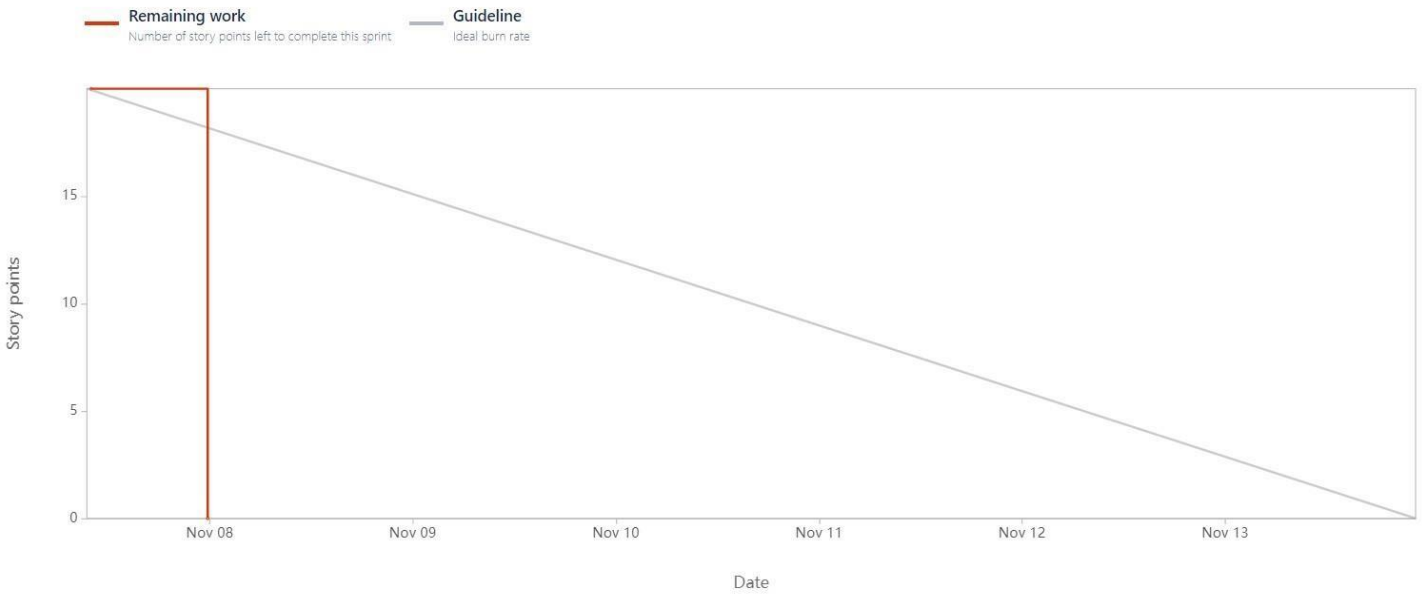
Sprint 2

Date - October 31st, 2022 - November 5th, 2022



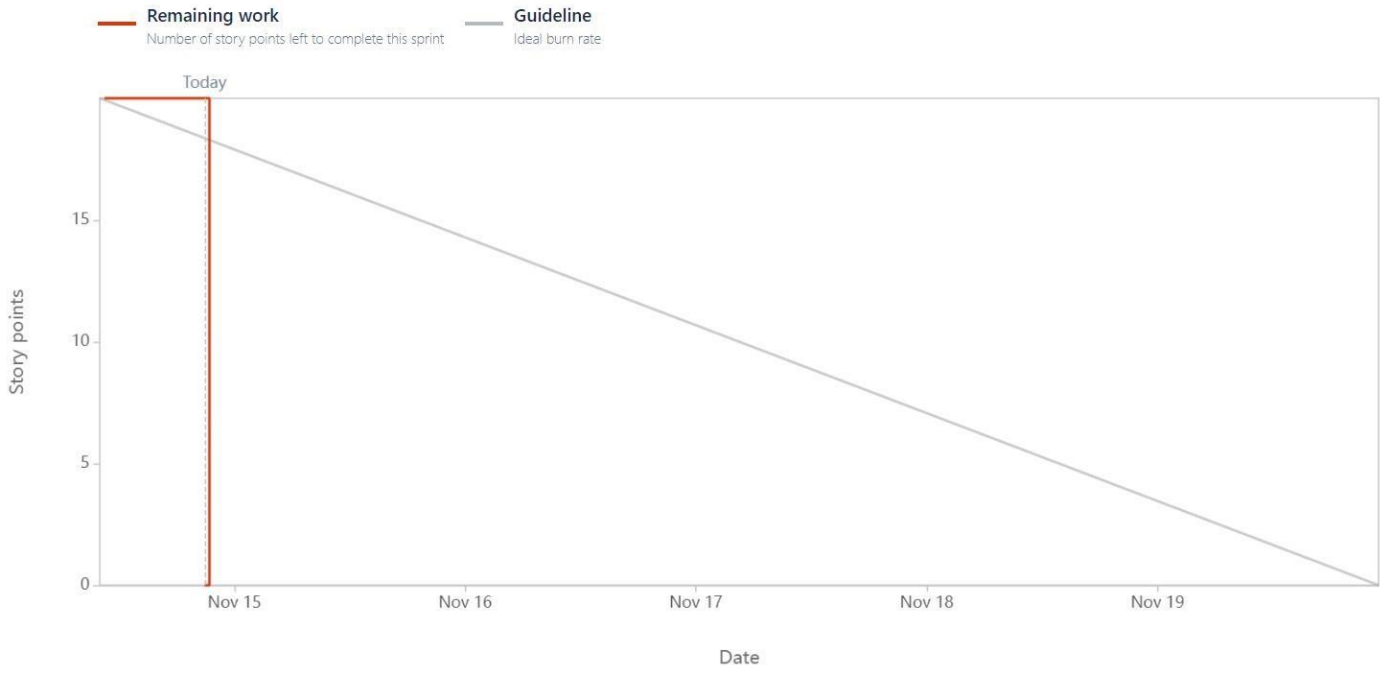
Sprint 3

Date - November 7th, 2022 - November 13th, 2022



Sprint 4

Date - November 14th, 2022 - November 19th, 2022



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

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


def random_name_generator(n: int) -> str:
    """
    Generates a random file name.

    Args:
        n (int): Length the of the file name.

    Returns:
        str: The file name.
    """
    return ''.join(random.choices(string.ascii_uppercase + string.digits, k=n))

def recognize(image: bytes) -> tuple:
    """
    Predicts the digit in the image.

    Args:
        image (bytes): The image data.

    Returns:
        tuple: The best prediction, other predictions and file name
    """

    model=load_model(Path("./model/modelCNN.h5"))

    img = Image.open(image).convert("L")

    # Generate a random name to save the image file.
    img_name = random_name_generator(10) + '.jpg'
    if not os.path.exists(f"./static/data/"):
        os.mkdir(os.path.join('./static/', 'data'))
    img.save(Path(f"./static/data/{img_name}"))

    # Convert the Image to Grayscale, Invert it and Resize to get better prediction.
    img = ImageOps.grayscale(img)
    img = ImageOps.invert(img)
    img = img.resize((28, 28))

    # Convert the image to an array and reshape the data to make prediction.
    img2arr = np.array(img)
    img2arr = img2arr / 255.0
    img2arr = img2arr.reshape(1, 28, 28, 1)

    results  = model.predict(img2arr)
    best = np.argmax(results,axis = 1)[0]

    # Get all the predictions and it's respective accuracy.
    pred = list(map(lambda x: round(x*100, 2), results[0]))
```

```
    values = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
    others = list(zip(values, pred))

    # Get the value with the highest accuracy
    best = others.pop(best)

    return best, others, img_name
```

8. TESTING

8.1 Test Cases

Test case ID	Feature Type	Component	Test Scenario	Expected Result	Actual Result	Status
Homepage_TC_OO1	Functional	Home Page	Verify user is able to see the Homepage when clicked on the link	Home Page should be displayed.	Working as expected	Pass
Homepage_TC_OO2	UI	Home Page	Verify the UI elements in Homepage	Application should show below UI elements: a.choose file button b.predict button c.clear button	Working as expected	Pass
Homepage_TC_OO3	Functional	Home Page	Verify user is able to choose file from the local system and click on predict	Choose file popup screen must be displayed and user should be able to click on predict button	Working as expected	Pass
Homepage_TC_OO4	Functional	Home page	Verify user able to select invalid file format	Application won't allow to attach formats other than ".png, .jiff, .pjp, .jpeg, .jpg, .pjpeg"	Working as expected	Pass

Predict_TC_OO5	Functional	Predict page	Verify user is able to navigate to the predict to and view the predicted result	User must be navigated to the predict page and must view the predicted result	Working as expected	Pass

8.2 User Acceptance Testing

Defect Analysis

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

Test Case Analysis

Section	Total Cases	Not Tested	Fail	Pass
Client Application	5	0	0	5

Security	5	0	0	5
Final Report Output	5	0	0	5
Performance	5	0	0	5

9. RESULTS

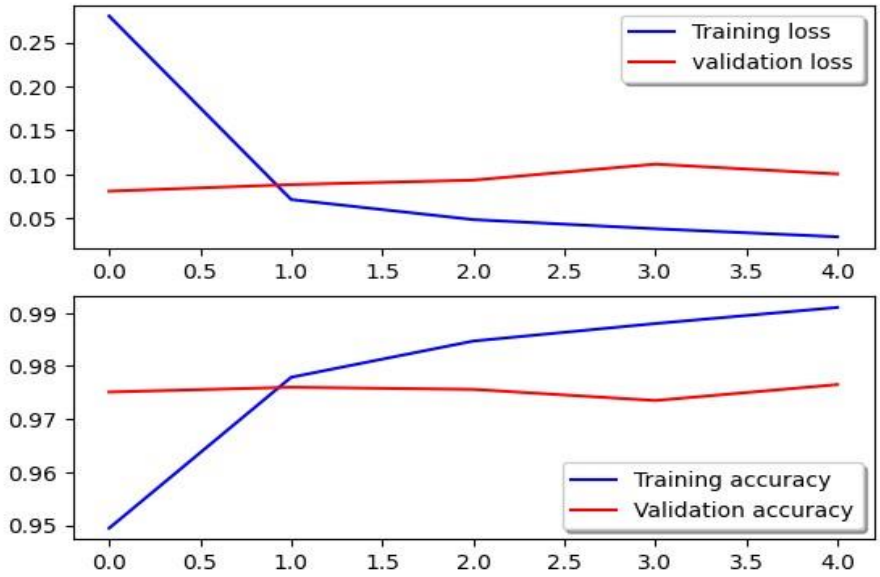
9.1 Performance Metrics

Model Summary:

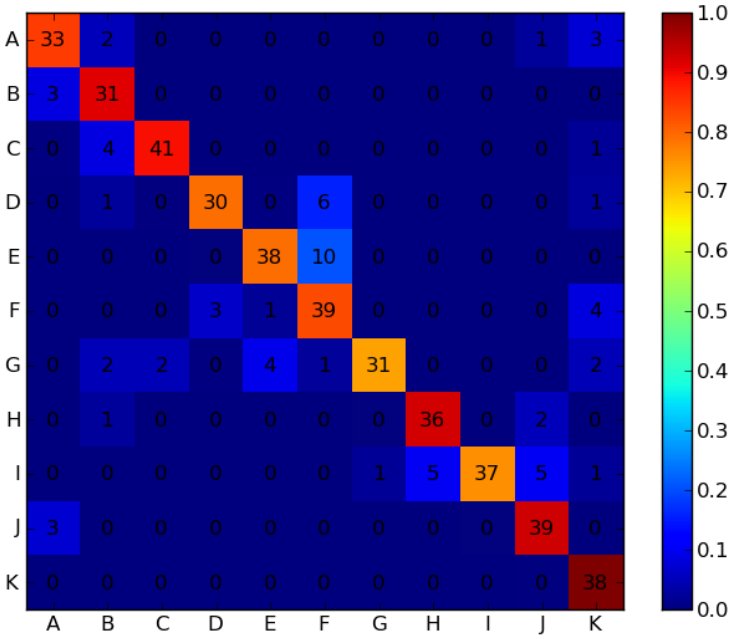
```
model.summary()

Model: "sequential_4"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
sequential_1 (Sequential)    (None, 1280)              410208
_________________________________________________________________
sequential_3 (Sequential)    (None, 4)                 128500
=================================================================
Total params: 538,708
Trainable params: 524,628
Non-trainable params: 14,080
_________________________________________________________________
```

Accuracy:



Confusion Matrix:



Classification Report:

	precision	recall	f1-score	support
1	0.69	0.68	0.68	68
2	0.59	0.71	0.65	184
3	0.67	0.66	0.66	147
4	0.57	0.54	0.55	39
5	0.56	0.40	0.47	70
6	0.65	0.64	0.64	130
7	0.70	0.55	0.62	29
8	0.60	0.58	0.59	112
9	0.79	0.79	0.79	29
avg / total	0.63	0.63	0.63	808

Performance Metrics Result:

Locust Test Report

During: 11/15/2022, 10:52:19 PM - 11/15/2022, 10:56:36 PM

Target Host: http://127.0.0.1:5000/

Script: locustfile.py

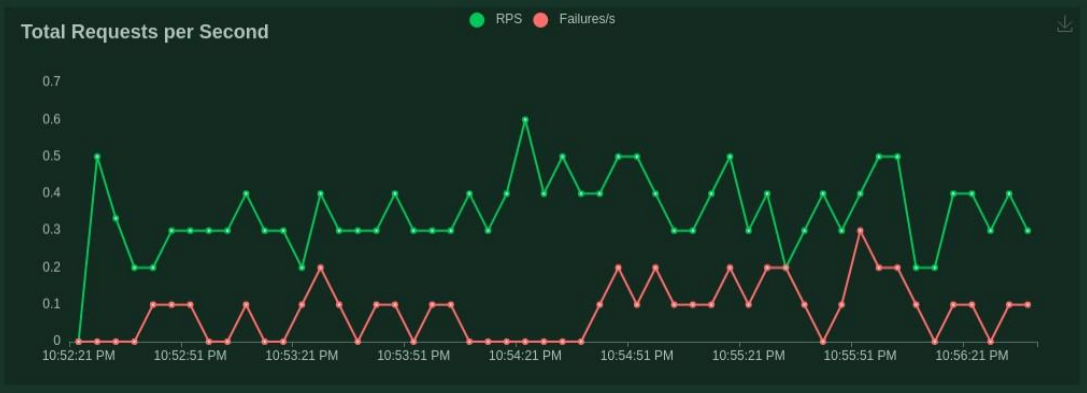
Request Statistics

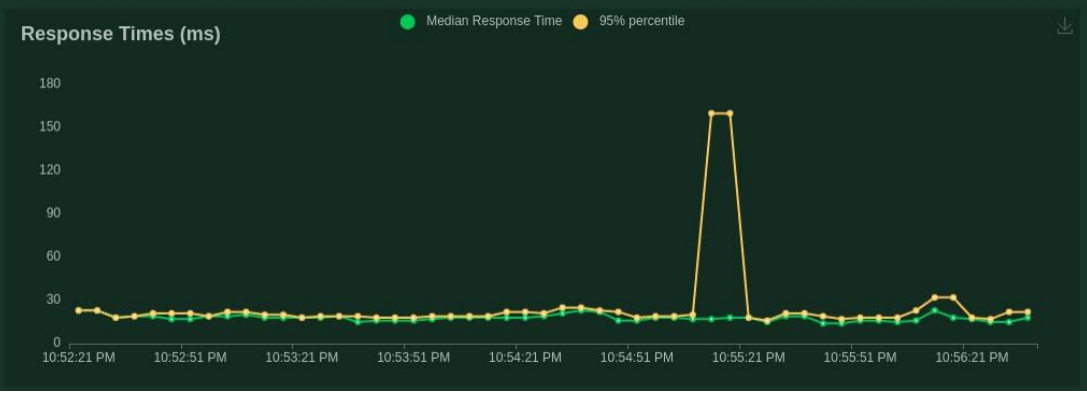
Method	Name	# Requests	# Fails	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	RPS	Failures/s
GET	//	67	0	17	12	24	5875	0.3	0.0
GET	//predict	23	23	21	11	163	265	0.1	0.1
	Aggregated	**90**	**23**	**18**	**11**	**163**	**4441**	**0.4**	**0.1**

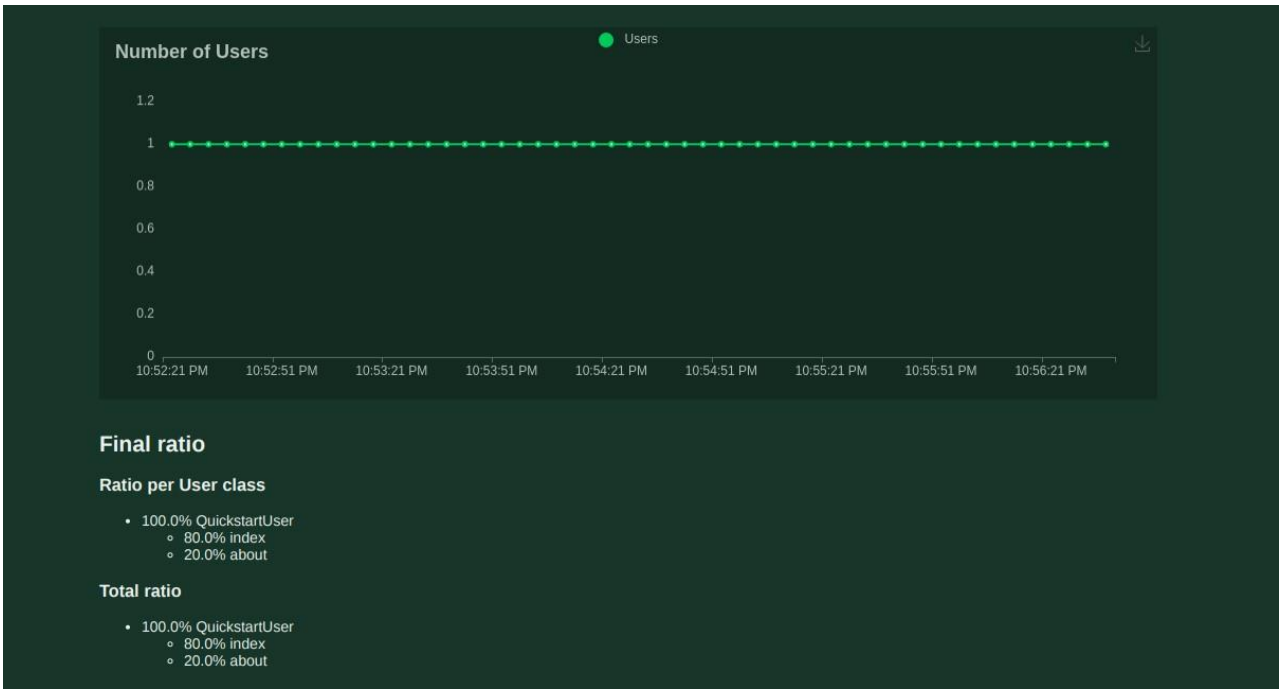
Response Time Statistics

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	//	18	18	19	19	22	23	25	25
GET	//predict	15	15	16	16	17	32	160	160
	Aggregated	**17**	**18**	**18**	**19**	**22**	**23**	**160**	**160**

Charts







Gatling report

10. ADVANTAGES & DISADVANTAGES

Advantages

Reduces manual work.

More accurate than average human.

Capable of handling a lot of data.

Can be used anywhere from any device.

Disadvantages

Cannot handle complex data.

All the data must be in digital format.

Requires high performance server for faster predictions**.**

Prone to occasional errors.

11. CONCLUSION

This project demonstrated a web application that uses machine learning to recognie handwritten numbers. Flask, HTML, CSS, JavaScript, and a few other technologies were used to create this project. The model predicts the handwritten digit using a CNN network. During testing, the model achieved a 99.61% recognition rate. The proposed project is scalable and can easily handle a huge number of users. Since it is a web application, it is compatible with any device that can run a browser. This project is extremely useful in realworld scenarios such as recognizing number plates of vehicles, processing bank cheque amounts, numeric entries in forms filled up by hand (tax forms) and so on. There is so much room for improvement, which can be implemented in subsequent versions.

12. FUTURE SCOPE

This project is far from complete and there is a lot of room for improvement. Some of the improvements that can be made to this project are as follows:

Add support to detect from digits multiple images and save the results

Add support to detect multiple digits

Improve model to detect digits from complex images

Add support to different languages to help users from all over the world

This project has endless potential and can always be enhanced to become better. Implementing this concept in the real world will benefit several industries and reduce the workload on many workers, enhancing overall work efficiency.

13. APPENDIX Source Code

HTML AND CSS: home.html:

```
<html>

<head>
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Handwritten Digit Recognition</title>
   <h3>TEAM ID : PNT2022TMID31528</h3>
    <link rel="icon" type="image/svg" sizes="32x32"
href="{{url_for('static',filename='images/icon.svg')}}" />
    <link rel="stylesheet" href="{{url_for('static',filename='css/main.css')}}" />
    <script src="https://unpkg.com/feather-icons"></script>
    <script defer src="{{url_for('static',filename='js/script.js')}}"></script>
</head>

<body>
    <div class="container">
        <div class="heading">
            <h1 class="heading__main" data-shadow='Handwritten Digit
Recognizer'>Handwritten Digit Recognizer</h1>
      <h2 class="heading__sub">Easily detect handwritten digits and Verfiy it</h2>
      <h2 class="heading__sub">TEAM ID: PNT2022TMID31528</h2>
        </div>
        <div class="upload-container">
            <div class="form-wrapper">
                <form class="upload" action="/predict" method="post"
enctype="multipart/form-data">
                    <label id="label" for="upload-image"><i data-feather="file-
plus"></i>Upload File</label>
                    <input type="file" name="photo" id="upload-image" hidden />
                    <button type="submit" id="up_btn"></button>
                </form>
                <img id="loading"
src="{{url_for('static',filename='images/loading.gif')}}">
            </div>
        </div>
    </div>
</body>

</html>
```

Predict.html:

```
</head>  <style>     body{      background-image: url('static/images/bc1.jpg'); <html>

<head>
   <title>Prediction | Handwritten Digit Recognition</title>
   <link rel="stylesheet" href="{{url_for('static',filename='css/predict.css')}}" />
   <link rel="icon" type="image/svg" sizes="32x32"
href="{{url_for('static',filename='images/icon.svg')}}" />
   <meta name="viewport" content="width=device-width, initial-scale=1.0" />
</head>

<body>
   <div class="container">
      <h1>PREDICTIONS</h1>
      <div class="result-wrapper">
         <div class="input-image-container">
            <img src="{{url_for('static',filename='data/')}}{{img_name}}" />
         </div>
         <div class="result-container">
            <div class="value">{{best.0}}</div>
            <div class="accuracy">{{best.1}}%</div>
         </div>
      </div>
      <h1>OTHER PREDICITONS</h1>
      <div class="other_predictions">
         {% for x in others %}
         <div class="value">
            <h2>{{x.0}}</h2>
            <div class="accuracy">{{x.1}}%</div>
         </div>
         {% endfor %}
      </div>
   </div>
</body>

</html>
```

Main.css

```
@import
url('https://fonts.googleapis.com/c
ss2?family=Open+Sans:wght@400;500;6
00&family=Righteous&family=Sono:wgh
t@400;500;600;700;800&display=swap'
);

* {
   padding: 0;
   margin: 0;
}

body {
   color: #f1faee;
   font-family: 'Brush Script MT',
cursive;
}



@keyframes gradient {
   0% {
      background-position: 0% 50%;
   }

   50% {
      background-position: 100%
50%;
   }

   100% {
      background-position: 0% 50%;
   }
}



.container {
   width: 100%;
   height: 100%;
   display: flex;
   flex-direction: column;
   justify-content: center;
   align-items: center;
   background-image:
url("paper.gif");
   background-attachment:fixed;
   background-size: 100% 100%;
   animation: gradient 5s ease
infinite;
```

```
}

.heading {
   margin-top: -2rem;
   padding-bottom: 2rem;
   width: fit-content;
   text-align: center;
   font-family: 'Brush Script MT',
cursive;
}

.heading .heading__main {
   font-size: 3rem;
   color: #ff0000;
   font-weight: 550;
}

.heading .heading__sub {
   font-size: 1rem;
   color: #ff0000;
   font-family: 'Brush Script MT',
cursive;

}

.upload-container {
   box-shadow: 0 0 20px rgb(33,
195, 183);
   width: 40rem;
   height: 25rem;
   padding: 1.5rem;
}

.form-wrapper {
   background-color: rgba(190, 190,
190, 0.5);
   width: 100%;
   height: 100%;
   display: flex;
   border: 2px dashed black;
   justify-content: center;
   align-items: center;
}

.form-wrapper #loading {
   display: none;
   position: absolute;
}

.form-wrapper .upload {
   display: flex;
   justify-content: center;
   align-items: center;
   width: 9rem;
   height: -webkit-fit-content;
   height: -moz-fit-content;
   height: fit-content;
   font-family: 'Brush Script MT',
cursive;
```

```
   border-radius: 6px;
   color: white;
   background-color: rgb(33, 195,
183);
   box-shadow: 0 5px 10px rgb(33,
195, 183);
}

.form-wrapper .upload #up_btn {
   display: none;
}

.form-wrapper .upload label {
   font-size: 1rem;
   font-weight: 600;
   color: white;
   height: 100%;
   width: 100%;
   padding: 10px;
   display: block;
}

.form-wrapper .upload svg {
   height: 15px;
   width: auto;
   padding-right: 8px;
   margin-bottom: -2px;
}

.buttons {
   margin: 20%;
   text-align: center;
}



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

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

Predict.css

```
@import
url('https://fonts.googleapis.com/c
ss2?family=Open+Sans:wght@400;500;6
00&family=Righteous&family=Sono:wgh
t@400;500;600;700;800&display=swap'
);

body {
   background-image:
url("pred.gif");

   background-size: 100% 100%;
   animation: gradient 15s ease
infinite;
   font-family: 'Brush Script MT',
cursive;
}

@keyframes gradient {
   0% {
      background-position: 0% 50%;
   }

   50% {
      background-position: 100%
50%;
   }

   100% {
      background-position: 0% 50%;
   }
}

h1 {
   padding-top: 2rem;
   color: #c1d8ff;
}

.container {
   display: flex;
   justify-content: center;
   align-items: center;
   flex-direction: column;
}

.result-wrapper {
   width: -webkit-fit-content;
   width: -moz-fit-content;
   width: fit-content;
   height: -webkit-fit-content;
   height: -moz-fit-content;
   height: fit-content;
   box-shadow: 0 0 10px rgb(126,
```

```
125, 125);
   padding: 1.5rem;
   display: flex;
   justify-content: center;
   align-items: center;
   -moz-column-gap: 1rem;
   column-gap: 1rem;
}

.result-wrapper .input-image-
container,
.result-wrapper .result-container {
   width: 15rem;
   height: 15rem;
   border: 1px dashed black;
   justify-content: center;
   display: flex;
   align-items: center;
   flex-direction: column;
   background-color: rgb(209, 206,
206);
   box-shadow: 0 0 20px rgb(17,
211, 255);
}

.result-wrapper .input-image-
container img {
   width: 60%;
   height: 60%;
   background-color: aqua;
   background-size: contain;
}

.result-wrapper .result-container
.value {
   font-size: 6rem;
}

.result-wrapper .result-container
.accuracy {
   margin-top: -1rem;
}

.other_predictions {
   display: flex;
   justify-content: center;
   align-items: center;
   flex-wrap: wrap;
   column-gap: 1rem;
   row-gap: 1rem;
   font-weight: 700;

}

.other_predictions .value {
   display: flex;
   justify-content: center;
   align-items: center;
   flex-direction: column;
```

```
    width: 5rem;
    height: 5rem;
    background-color: #e6e6e6;
    box-shadow: 0 0 7px rgb(23, 236,
255);
}

.other_predictions .value div {
    margin-top: -1.2rem;
}

@media screen and (max-width:
700px) {
    h1 {
        font-size: 2.3rem;
    }

    .result-wrapper .input-image-
container,
    .result-wrapper .result-
container {
        width: 7rem;
        height: 7rem;
    }

    .result-wrapper .result-
container .value {
        font-size: 4rem;
    }
}
```

FLASK

App.py

```python
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()

```

Recognizer.py

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


def random_name_generator(n: int) -> str:
    """
    Generates a random file name.

    Args:
        n (int): Length the of the file name.

    Returns:
        str: The file name.
    """
    return ''.join(random.choices(string.ascii_uppercase + string.digits, k=n))

def recognize(image: bytes) -> tuple:
    """
    Predicts the digit in the image.

    Args:
        image (bytes): The image data.

    Returns:
        tuple: The best prediction, other predictions and file name
```

```
    model=load_model(Path("./model/modelCNN.h5"))

    img = Image.open(image).convert("L")

    # Generate a random name to save the image file.
    img_name = random_name_generator(10) + '.jpg'
    if not os.path.exists(f"./static/data/"):
        os.mkdir(os.path.join('./static/', 'data'))
    img.save(Path(f"./static/data/{img_name}"))

    # Convert the Image to Grayscale, Invert it and Resize to get better prediction.
    img = ImageOps.grayscale(img)
    img = ImageOps.invert(img)
    img = img.resize((28, 28))

    # Convert the image to an array and reshape the data to make prediction.
    img2arr = np.array(img)
    img2arr = img2arr / 255.0
    img2arr = img2arr.reshape(1, 28, 28, 1)

    results  = model.predict(img2arr)
    best = np.argmax(results,axis = 1)[0]

    # Get all the predictions and it's respective accuracy.
    pred = list(map(lambda x: round(x*100, 2), results[0]))

    values = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
    others = list(zip(values, pred))

    # Get the value with the highest accuracy
```

MODEL CREATION:

GitHub & Project Demo Link

GitHub Link https://github.com/IBM-EPBL/IBM-Project-39033-1660389623

Demo Video
https://github.com/IBM-EPBL/IBM-Project-39033-1660389623/tree/main/Final%20Deliverables/Demo%20Video