PROJECT REPORT

A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

Submitted By

PNT2022TMID26942

Hariharan K - 310819104031

Gunalan E - 310819104030

Logeshwaran K - 310819104047

Kumaari Narendra - 310819104045

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Project Overview

Machine learning and deep learning play 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 in many more areas.

Handwritten Digit Recognition is the ability of Computer systems to recognize handwritten digits from various sources, such as images, documents, and so on. This project aims to let users take advantage of machine learning to reduce manual tasks in recognizing digits.

1.2 Purpose

Digit Recognition system are capable of recognizing the digits from different sources like emails, bank cheques, papers, images, etc. and in different real-world scenarios for online handwriting recognition on computer, tablets or systems, recognize number plates of vehicles, processing bank cheque amounts, numeric enteries in forms filled up by hand (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

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

	Vijayalaxmi R Rudraswamimath, Bhavanishankar and Channasandra.	Handwritten Digit Recognition using CNN	International Journal of Innovative Science and Research Technology	Volume -4 Issue6	2019	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
	Fathma Siddique, 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)		2019	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.

	Akanksha 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	2021	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.

	Md. 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	2019	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.

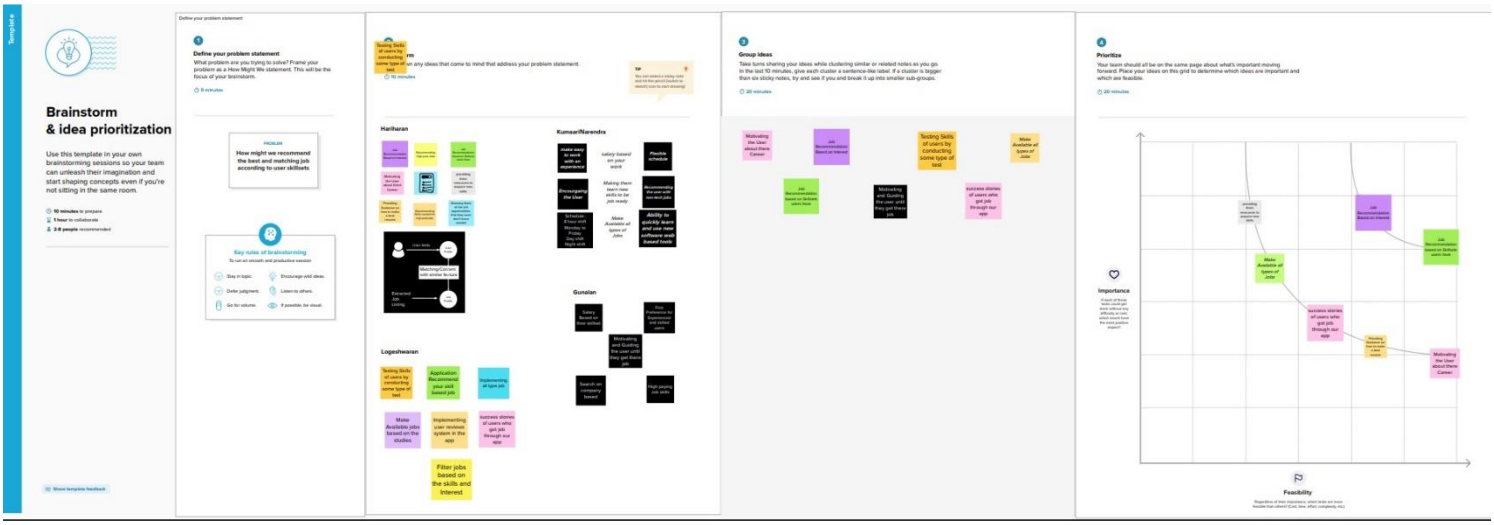
Hence, there comes a need for handwritten digit recognition in many real time applications. MNIST data set is widely used for this recognition process and it has 70000 handwritten digits. We use Artificial neural networks to train these images and build a deep learning model. Web application is created 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)	Manually written digits are of a different size, thickness, position and direction. In this manner, various difficulties must be considered to determine the issue of handwritten digit recognition.
2.	Idea / Solution description	To solve this problem, we are going to implement a classification algorithm to recognize the handwritten digits. This algorithm will be effective in order to recognize digits which are of different compositions.
3.	Novelty / Uniqueness	• Strategy for perceiving and arranging transcribed digits. • Can be used offline • Provided more data sets for more accuracy • The uniqueness and assortment in the composition styles of various individuals additionally influence the example and presence of the digits.

4.	Social Impact / Customer Satisfaction	The main social impact of this work is to ensure effective and reliable approaches for recognition of handwritten digits and make banking operations easier and error free. Customers will feel at ease, as it is easy and convenient to use. As the accuracy is acceptable, this can have many applications.
5.	Business Model (Revenue Model)	This novel method for Handwritten Digit Recognition System can be approached by many industries which needs this application including, programmed bank checks, postal locations and tax documents and so on. Humans recognizing the handwritten digits with their naked eye can be difficult at times as it of different sizes, thickness, direction and ca also lead to making errors due to these factors. This is when our proposed solutions comesinto help. We provide different data sets which helps in recognition with accuracy, so that human making errors can be avoided respectively.
6.	Scalability of the Solution	Financial and other business organizations such as banks are facing issues in Recognizing written digits such as in cheques etc. This can be handled by our handwritten digit recognition project as they expand into different business domains without impacting performance. Our proposed solution is scalable as it is dynamic and also trained using AI and deep learning Models.

3.4 Problem Solution fit

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Who is your customer?
i.e. working parents of 0-5 y.o. Kids

Organizations who want to recognize the handwritten digits of people
Example:

- ✓ Post office,
- ✓ Data entry offices,
- ✓ Forensic Departments.

6. CUSTOMER CONSTRAINTS CC

What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices.

In mobiles and laptop, there are possibilities for lack of stable internet connections and unavailability of devices. It is hard task for the machine to recognize the handwritten digits which are not perfect.

5. AVAILABLE SOLUTIONS AS

Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper is an alternative to digital note taking.
Already there are existing solutions available for handwritten
recognition. But, most of them are inaccurate.
The solution proposed by our system has more accuracy
and it is efficient in recognition of manually written digits.

Explore AS, differentiate

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

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

Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides.

Jobs to be done: To identify the digits in the manually written forms,

Cheques filled by people in banks, Phone numbers written manually in register notebook of hospitals.

Problems: Dim lighting and weak eyesight

9. PROBLEM ROOT CAUSE RC

What is the real reason that this problem exists? What is the backstory behind the need to do this job?

i.e. customers have to do it because of the change in regulations.

Handwritten digits are in different fonts and sizes, hard to recognize the digits due to various factors such as dim lighting, weakening eyesight.

7. BEHAVIOUR BE

What does your customer do to address the problem and get the job done?

i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)

customer wants available devices with stable internet connection and quality cameras.

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

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

Functional Requirements:

Following are the 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

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.

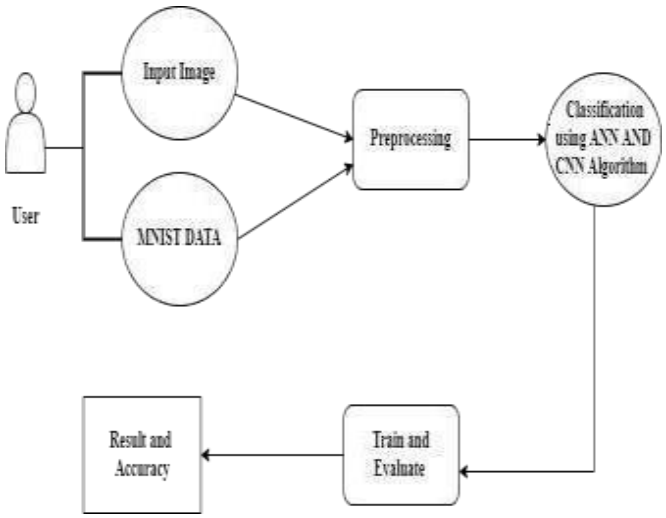
Non-functional Requirements:

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

		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.
NFR-5	**Availability**	

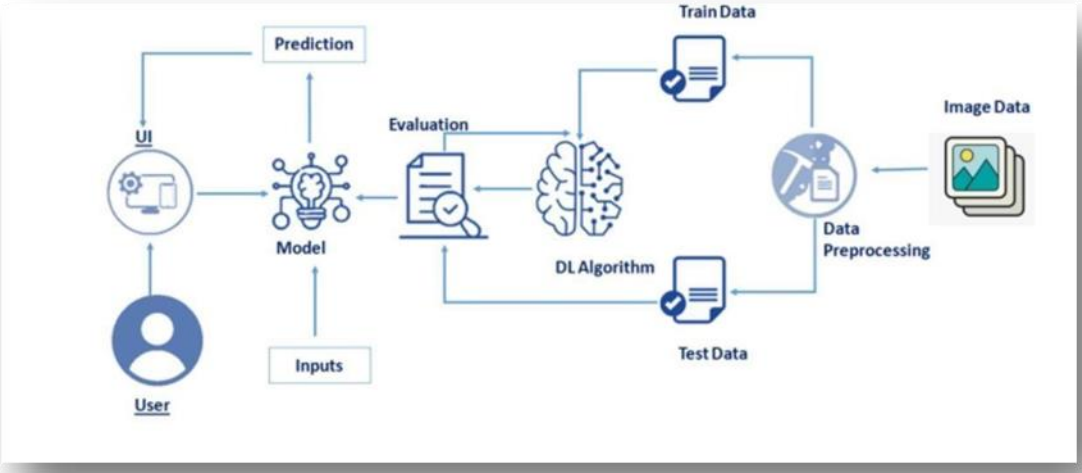
5. PROJECT DESIGN

5.1 Data Flow Diagrams



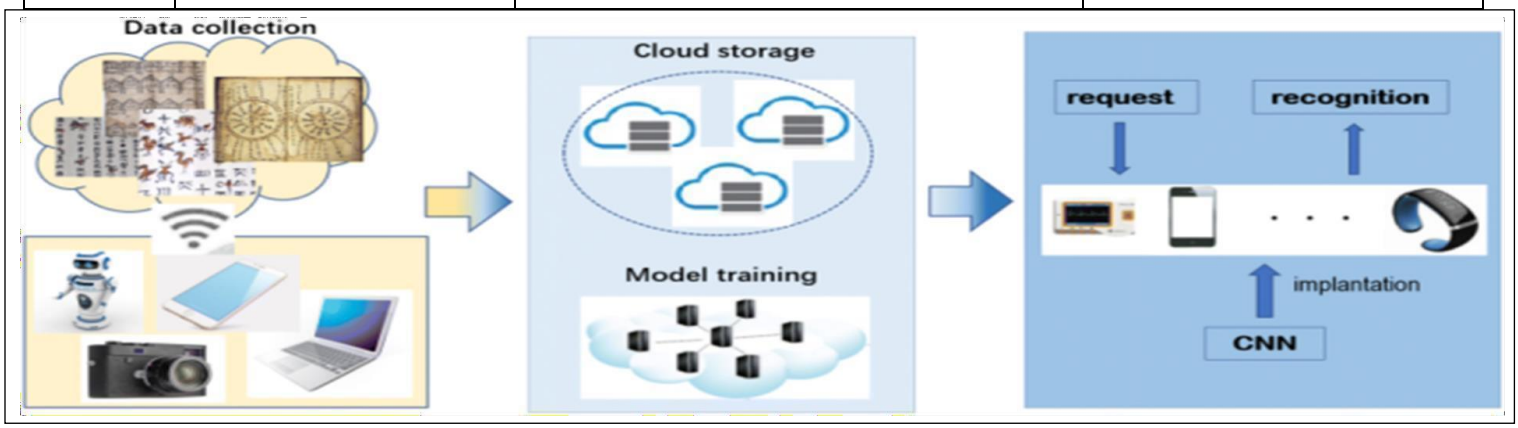
5.2 Solution & Technical Architecture

Solution Architecture



Technology Architecture

S.No	Characteristics	Description	Technology
3.	Scalable Architecture	Justify the scalability of architecture	3 – tier, Micro-services



Components & Technologies:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used	Technology of Opensource framework
2.	Security Implementations	List all the security / access controls implemented, use of firewalls etc.	SHA-256, Encryptions, IAM Controls, OWASP

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g. Web UI, Mobile App, Chatbot etc.	HTML, CSS, JavaScript
2.	Application Logic-1	Logic for a process in the application	Python
3.	Application Logic-2	Logic for a process in the application	IBM Watson STT service
4.	Application Logic-3	Logic for a process in the application	IBM Watson Assistant
5.	Database	Data Type, Configurations etc.	MySQL, NoSQL, etc.
6.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant
7.	File Storage	File storage requirements	IBM Block Storage
8.	External API-1	Purpose of External API used in the application	IBM Weather API
9.	External API-2	Purpose of External API used in the application	Aadhar API
10.	Machine Learning Model	Purpose of Machine Learning Model	Object Recognition Model
11.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration Cloud Server Configuration	Local, Cloud Foundry

Application Characteristics:

4.	Availability	Abstract and Figures. The features for handwritten digit recognition have been introduced. These features are based on shape analysis of the digit image and extract slant or slope information. They are effective in obtaining good recognition accuracies	Distributed servers, IBM cloud
5.	Performance	The standard implementations of neural networks achieve an accuracy of ~ (98–99) percent in correctly classifying the handwritten digits.	number of requests per sec, use of Cache, use of CDN's

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, I can view the guide and awareness to use this application.	I can view the awareness to use this application and its limitations.	Low	Sprint-1
		USN-2	As a user, I'm allowed to view the guided video to use the interface of this 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 userfriendly method.	Low	Sprint-2
	Recognize	USN-4	As a user, In this prediction page I get to choose the image.	I can choose the image from our local system and predict the output.	High	Sprint-2
	Predict	USN-6	As a user, I'm Allowed to upload and choose the image to be uploaded	I can upload and choose the image from the system storage and also in any virtual storage.	Medium	Sprint-3
		USN-7	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-8	As a user, I can access the MNIST data set	I can access the MNIST data set to produce the accurate result.	Medium	Sprint-3
Customer (Web user)	Home	USN-9	As a user, I can view the guide to use the web app.	I can view the awareness of this application and its limitations.	Low	Sprint-1

User Type	Functional Requirement	User Story	User Story / Task	Acceptance criteria	Priority	Release

	(Epic)	Number				
Customer (Mobile user)	Home	USN-1	As a user, I can view the guide and awareness to use this application.	I can view the awareness to use this application and its limitations.	Low	Sprint-1
		USN-2	As a user, I'm allowed to view the guided video to use the interface of this 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 userfriendly method.	Low	Sprint-2
	Recognize	USN-10	As a user, I can use the web application virtually anywhere.	I can use the application portably anywhere.	High	Sprint-1
		USN-11	As it is an open source, can use it cost freely.	I can use it without any payment to be paid for it to access.	Medium	Sprint-2
		USN-12	As it is a web application, it is installation free	I can use it without the installation of the application or any software.	Medium	Sprint-4
	Predict	USN-13	As a user, I'm Allowed to upload and choose the image to be uploaded	I can upload and choose the image from the system storage and also in any virtual storage.	Medium	Sprint-3

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	Data Collection & pre processing	USN-1	As a user, I can upload any kind of image with the pre-processing step is involved in it.	10	High	Kumari Narendra, Logeshwaran.K
Sprint-1		USN-2	As a user, I can upload the image in any resolution.	10	High	Hariharan.K, Gunalan.E

Sprint-2	Building the Machine learning model	USN-3	As a user, I will get a application with ML model which provides high accuracy of recognized handwritten digit	3	Medium	Hariharan.K, Logeshwaran.K
Sprint-2		USN-4	As a user, I can pass the handwritten digit image for recognizing the digit.	2	Medium	Hariharan.K, Logeshwaran.K
Sprint	**Functional Requirement (Epic)**	**User Story Number**	**User Story / Task**	**Story Points**	**Priority**	**Team Members**
Sprint-2		USN-5	As a user, I can get the most suitable recognized digit.	10	High	Logeshwaran.K, Gunalan.E
Sprint-3	Building User Interface Application	USN-6	As a user, I will upload the handwritten digit image to the application by clicking a upload button.	8	Medium	Kumari Narendra, Hariharan.K
Sprint-3		USN-7	As a user, I can know the details of the fundamental usage of the application.	2	High	Gunalan.E, Kumari Narendra
Sprint-3		USN-8	As a user, I can see the predicted / recognized digits in the application	10	Medium	Hariharan.K, Logeshwaran.K
Sprint-4	Train and deployment of model in IBM Cloud	USN-9	As a user, I can access the web application and make the use of the product from anywhere	20	High	Hariharan.K, Gunalan.E, Logeshwaran.K, Kumari Narendra

6.2 Sprint Delivery Schedule

Project Tracker, Velocity & Burndown Chart: (4 Marks)

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

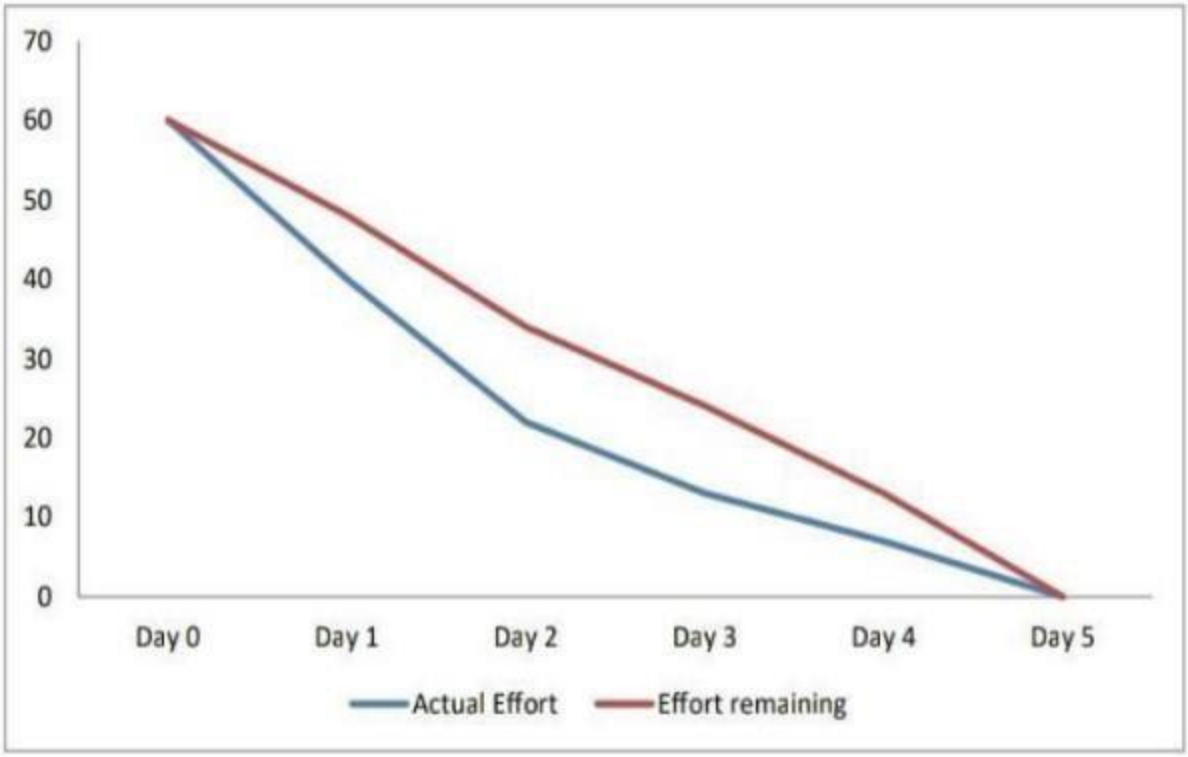
Velocity:

Imagine we have a 6-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)

$$AV = \text{sprint duration} / \text{Velocity}$$
$$= 20/6 = 3.33$$

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.

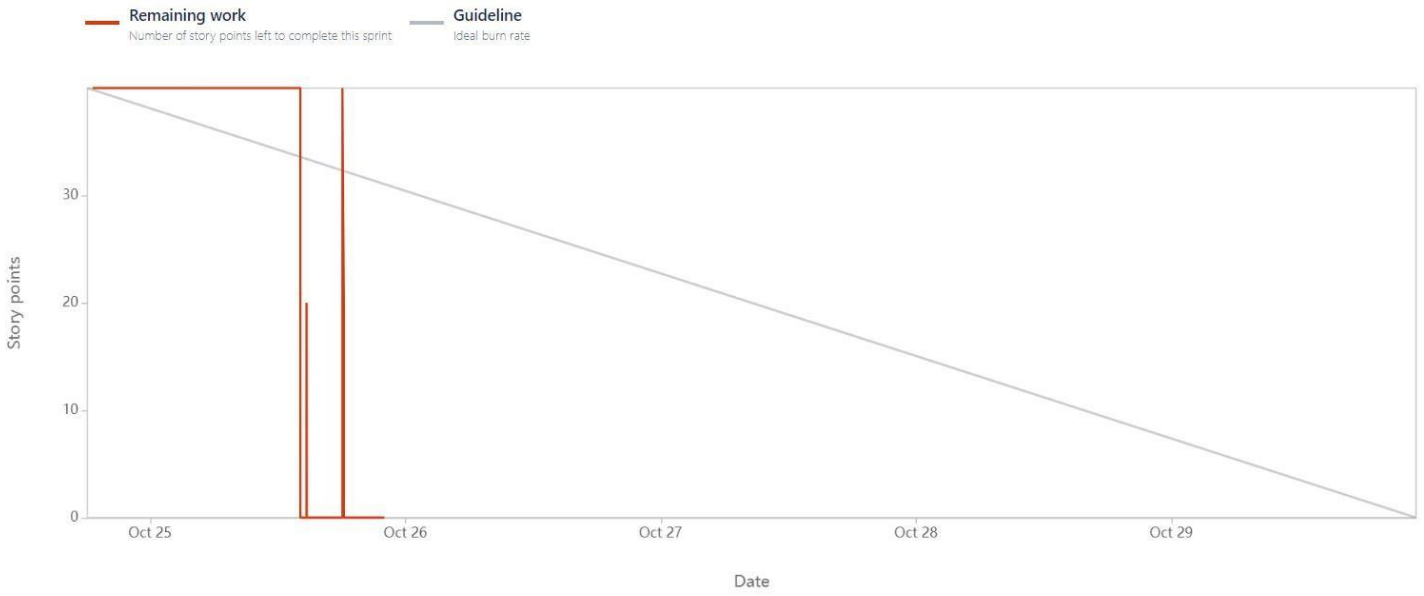


6.3 Reports from JIRA

Velocity Report

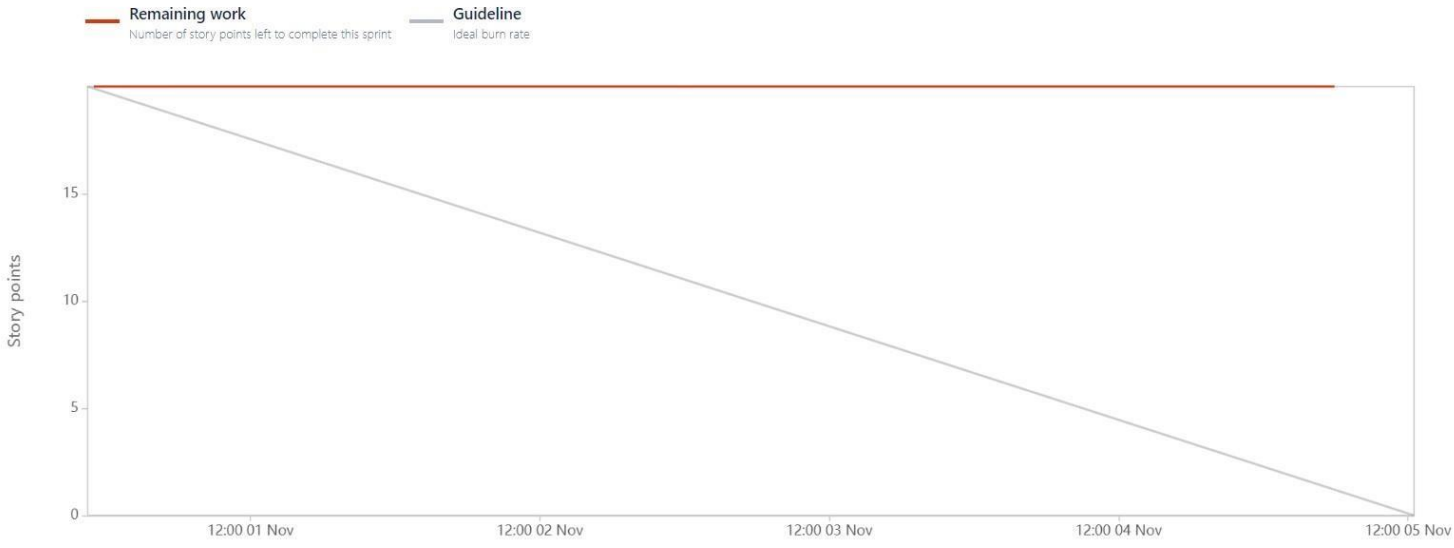
` **Sprint 1**

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



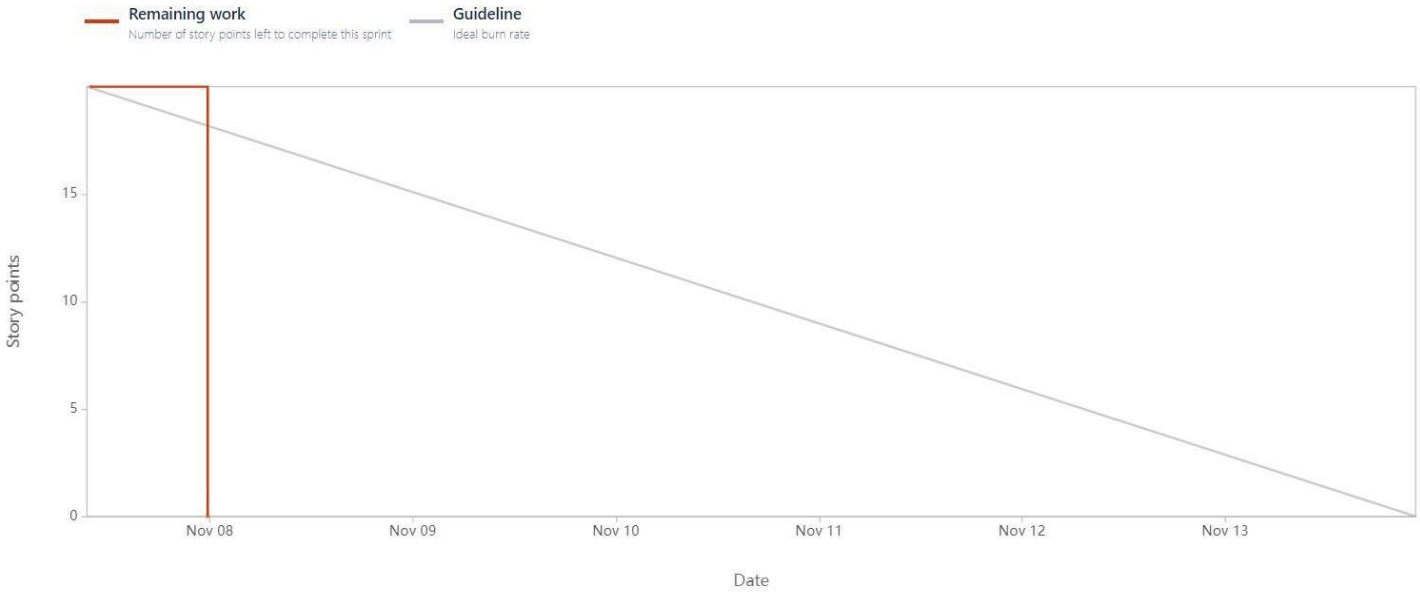
Sprint 2

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

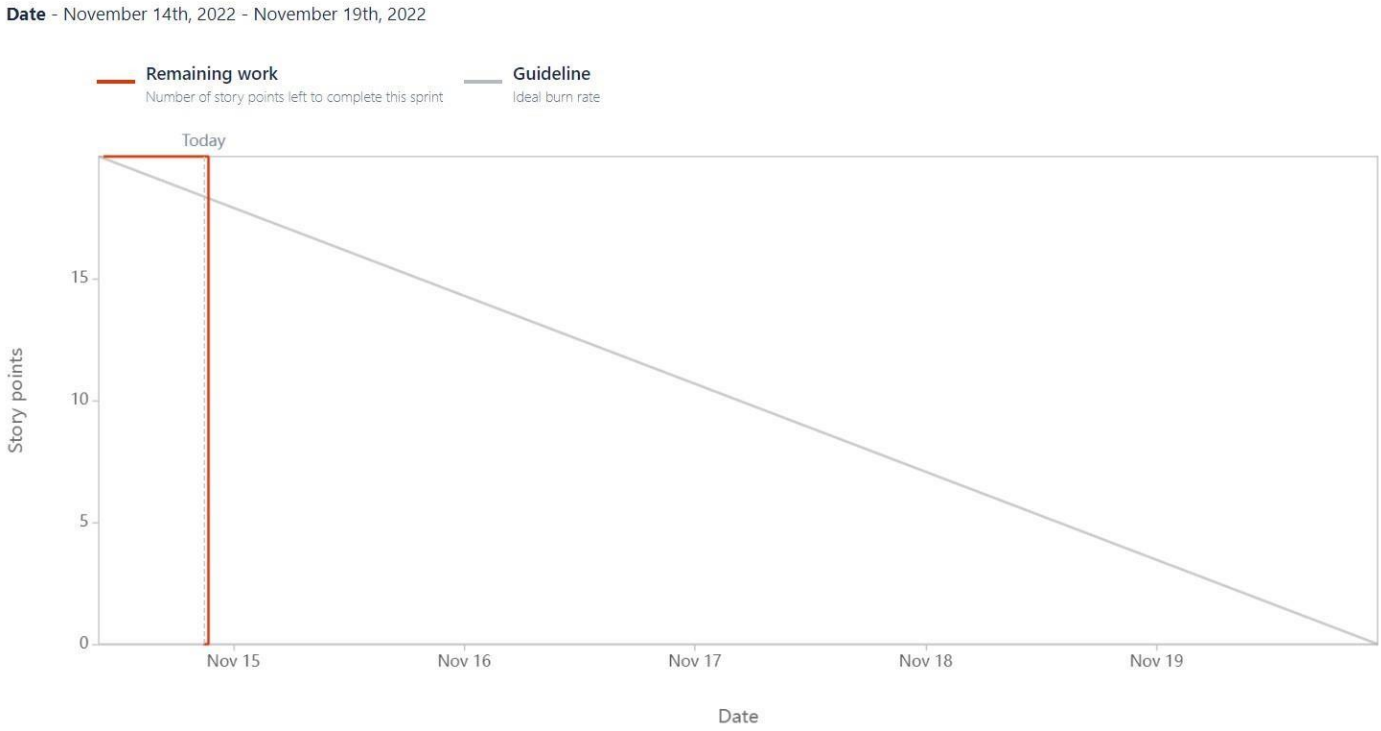


Sprint 3

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



Sprint 4



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

```
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 = 'D:/ibm/data'


app = Flask(__name__)
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER

model = load_model("./models/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)
```

8. TESTING

8.1 Test Cases

Test case ID	Feature Type	Component	Test Scenario	Expected Result	Actual Result	Status
Homepage_TC_001	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_002	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_003	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_004	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_005	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

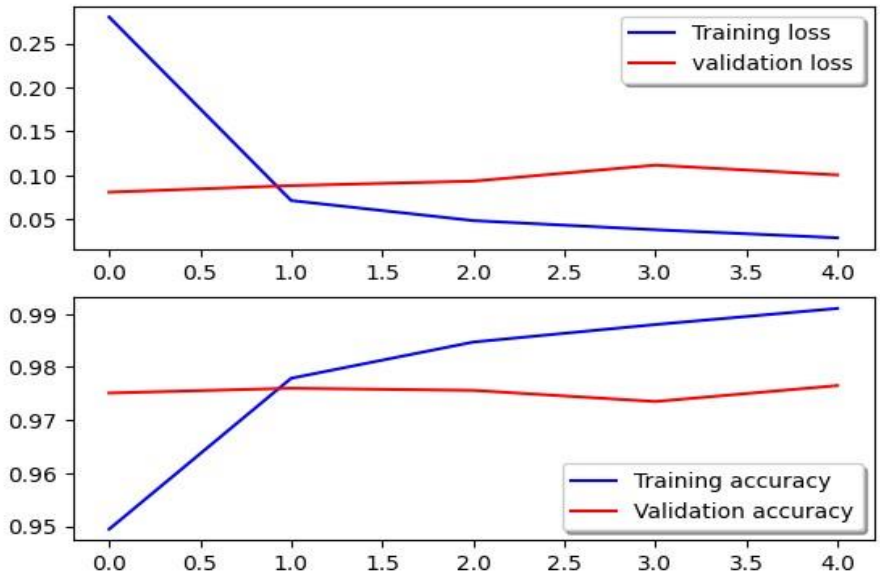
9. RESULTS

9.1 Performance Metrics

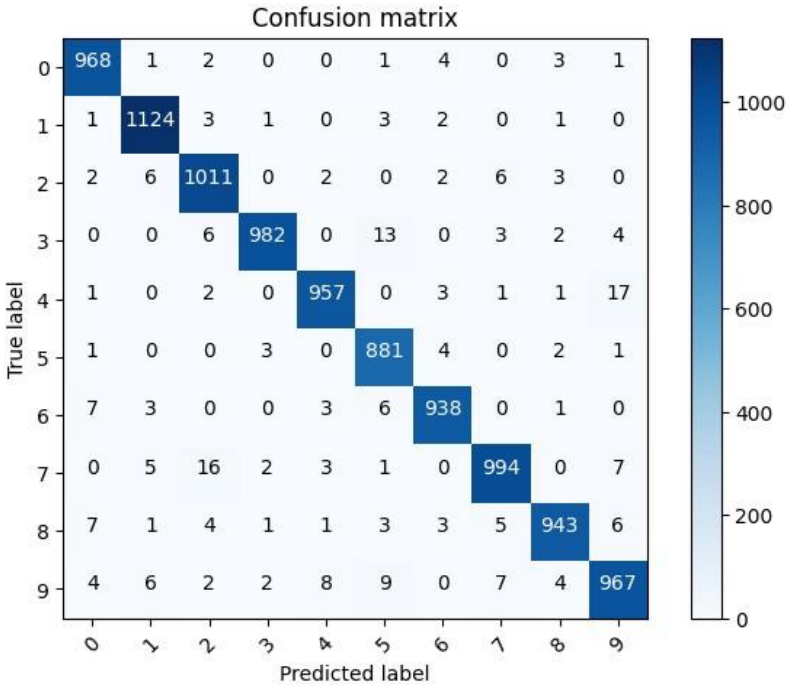
Model Summary:

```
Model: "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
```

Accuracy:



Confusion Matrix:



Classification Report:

	precision	recall	f1-score	support
0	0.98	0.99	0.98	980
1	0.98	0.99	0.99	1135
2	0.97	0.98	0.97	1032
3	0.99	0.97	0.98	1010
4	0.98	0.97	0.98	982
5	0.96	0.99	0.97	892
6	0.98	0.98	0.98	958
7	0.98	0.97	0.97	1028
8	0.98	0.97	0.98	974
9	0.96	0.96	0.96	1009
accuracy			0.98	10000
macro avg	0.98	0.98	0.98	10000
weighted avg	0.98	0.98	0.98	10000

Performance Metrics Result:

Locust Test Report

During: 11/20/2022, 07:37:46 PM - 11/20/2022, 07:42:14 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**

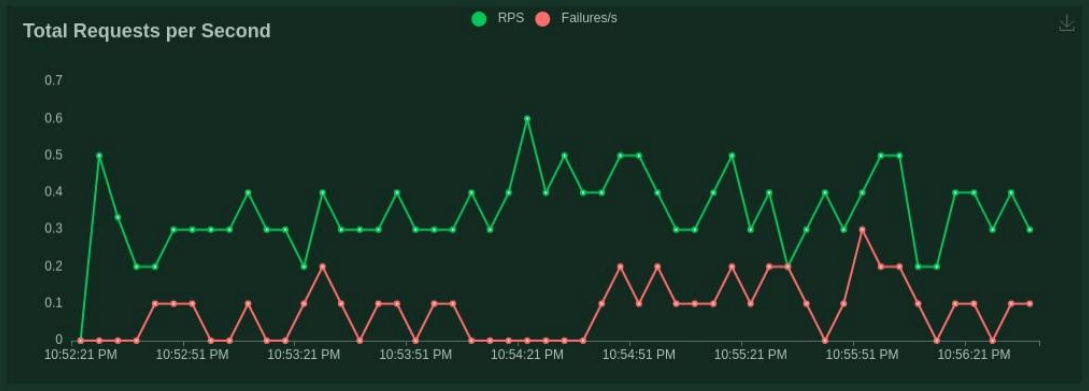
Response Time Statistics

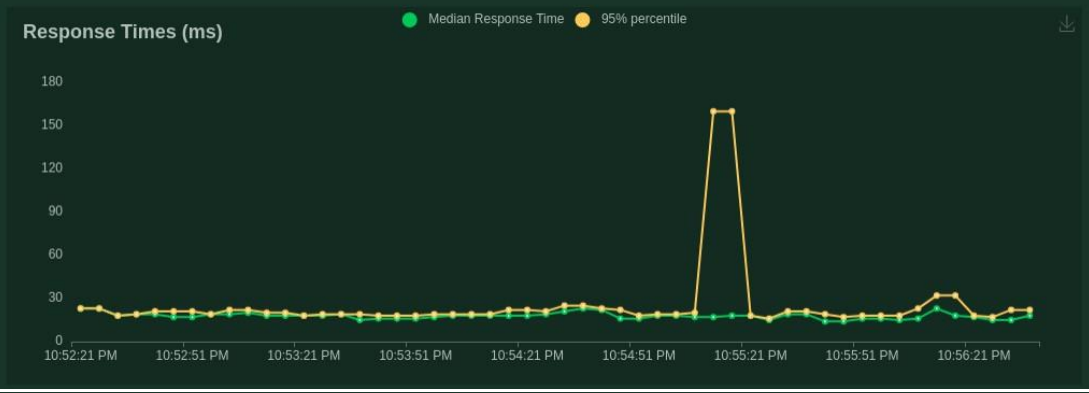
Method	Name	50%ile (ms)	60%ile (ms)	70%ile (ms)	80%ile (ms)	90%ile (ms)	95%ile (ms)	99%ile (ms)	100%ile (ms)
GET	//	18	18	19	19	22	23	25	25
GET	//predict	15	15	16	16	17	32	160	160
	Aggregated	**17**	**18**	**18**	**19**	**22**	**23**	**160**	**160**

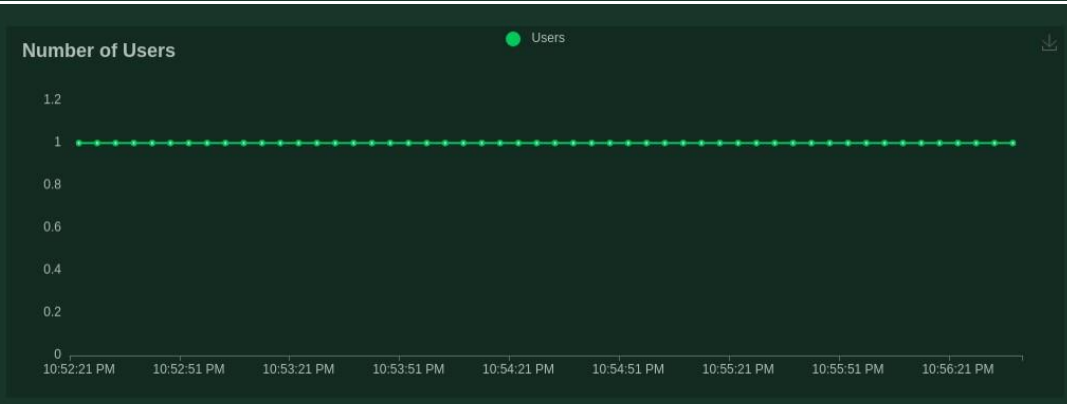
Failures Statistics

Method	Name	Error	Occurrences
GET	//predict	500 Server Error: INTERNAL SERVER ERROR for url: http://127.0.0.1:5000//predict	23

Charts







Final ratio

Ratio per User class

- 100.0% QuickstartUser
  - 80.0% index
  - 20.0% about

Total ratio

- 100.0% QuickstartUser
  - 80.0% index
  - 20.0% about

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: index.html:

```
  <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') }}">

  <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="sha384q8i/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>
  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.1/dist/css/bootstrap.min.css">
  <script src="https://cdn.jsdelivr.net/npm/jquery@3.6.0/dist/jquery.slim.min.js"></script>
  <script
src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js"></script>
  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.1/dist/js/bootstrap.bundle.min.js"></script>
</head> <style>     body{      background-image:
url('static/images/bc1.jpg');      background-
repeat: no-repeat;      background-size: cover;
    }
</style>
<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>HandWritten Digit Recognition System</h1>
        <div class="container p-3 my-3 bg-dark text-white">
            <p>Handwritten Digit Recognition is a technology that is much needed in this
world as of Today.This Digit Recognition System is used to recognize the digits from
different sources like email, posts, cheque etc. Before proper implementation of this
technology we have relied on writing text with our own hands which can result in error.It's
difficult to store and access physical data with efficiency.The project presents in
representing the recognization of handwritten digits (0 - 9) from the famous MNIST dataset.
Here we will be using Convolutional Neural Network for the prediction.</p>
</div>
        <section id="content">

            <div class="leftside">
            <form action="/predict" method="POST" enctype="multipart/form-data">
            <label>Select a image:</label>
            <input id="image" type="file" name="image" accept="image/png, image/jpeg"
onchange="preview()"><br><br>
              <img id="frame" width="100px" height="100px"/>
              <div class="buttons_div">
                <button type="submit" class="btn btn-light">Predict</button>
                <button type="button" class="btn btn-light">  Clear  </button>
              </div>
            </form>
            </div>
      </section>


</body>
</html>
```

Predict.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Prediction</title>
</head>  <style>     body{      background-image: url('static/images/bc1.jpg');
```

```
 background-repeat: no-repeat;
background-size: cover;
}
```

```
  #clear_button{     margin-
left: 15px;     font-weight:
bold;     color: rgb(0, 174,
255);
  }
```

```
  #confidence{     font-family: 'Josefin
Sans', sans-serif;
```

```
    #rectangle{
 width:600px;     height:150px;
 background-color: #000000;
```

```
     border-radius: 25px;
position:absolute;      box-shadow: 0px
0px 10px 5px white;      top:25%;
left:50%;      transform:translate(-
50%,-50%);
    }

    #num{   text-
align: center;
font-size: 30px;
margin: 0 auto;
padding: 3% 5%;
padding-top: 8%;
color: white;
    }

</style>

<body>

    <div id="rectangle">
        <h1 id="num">Predicted Number is {{num}}</h1>
    </div>

</body>
</html>
```

Style.css

```
    margin-top: 7.5%;
  }
    #content{
 margin: 0 auto;
```

```
padding: 2% 15%;
padding-bottom: 0;
 }
```

```
    .welcome{      text-align: center;
position: relative;      color: rgb(0, 32, 112);
background-color: skyblue;      padding-top: 1%;
padding-bottom: 1%;      font-weight: bold;
font-family: 'Bookman', 'URW Bookman L', serif;
  }
    #team_id{
text-align: right;
font-size: 25px;
padding-right: 3%;
color: rgb(255, 0, 0);
    background-color:
rgb(0, 0, 0);
  }
  }

  #predict_button{     margin-
right: 15px;     color:
rgb(44, 241, 0);     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;
  }
```

```
.btn {
```

```
     font-size: 15px;
    padding: 10px;
     /* -webkit-appearance: none; */
background: #eee;      border: 1px
solid #888;      margin-top: 20px;
margin-bottom: 20px;
 }
```

```
  .buttons_div{     margin-
bottom: 30px;     margin-
right: 80px;
  }
     .heading{     font-family:"American
Typewriter", 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%; */
  }
h1{
    text-align: center;     color:
aliceblue;     padding: 100px 50px
65px 100px;
}

  @media (min-width: 720px) {
    .leftside{
```

```
        padding-left: 10%;
    }
}
```

FLASK:

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 = 'D:/ibm/data'


app = Flask(__name__) app.config['UPLOAD_FOLDER']
= UPLOAD_FOLDER
 model =
load_model("./DigitRecog_IBM_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
#faltten -used fot flattening the input or change the dimension from tensorflow.keras.layers
import Conv2D #onvoLutiona l Layer from keras.optimizers import Adam #opt imizer 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 ()
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  #storing the no of classes in a
variable y_train = np_utils.to_categorical (y_train, number_of_classes) #converts the output
in binary format y_test = np_utils.to_categorical (y_test, number_of_classes)
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) history =
model.fit(x_train, y_train, validation_data=(x_test, y_test), epochs=5, batch_size=32)

from sklearn.model_selection import
train_test_split from sklearn.metrics import
confusion_matrix import itertools

fig, ax = plt.subplots(2,1) ax[0].plot(history.history['loss'], color='b',
label="Training loss") ax[0].plot(history.history['val_loss'], color='r',
label="validation loss",axes =ax[0]) legend = ax[0].legend(loc='best', shadow=True)
```

```
ax[1].plot(history.history['accuracy'], color='b', label="Training accuracy")
ax[1].plot(history.history['val_accuracy'], color='r',label="Validation accuracy") legend
= ax[1].legend(loc='best', shadow=True)
 def plot_confusion_matrix(cm, classes,
normalize=False,
title='Confusion matrix',
cmap=plt.cm.Blues):
    """
    This function prints and plots the confusion matrix.
    Normalization can be applied by setting `normalize=True`.
    """     plt.imshow(cm, interpolation='nearest',
cmap=cmap)     plt.title(title)     plt.colorbar()
tick_marks = np.arange(len(classes))
plt.xticks(tick_marks, classes, rotation=45)
plt.yticks(tick_marks, classes)
     if
normalize:
        cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
     thresh = cm.max() / 2.     for i, j in
itertools.product(range(cm.shape[0]), range(cm.shape[1])):
        plt.text(j, i, cm[i, j],
horizontalalignment="center",                 color="white" if
cm[i, j] > thresh else "black")
     plt.tight_layout()
plt.ylabel('True label')
plt.xlabel('Predicted label')

# Predict the values from the validation dataset
Y_pred = model.predict(x_test)
# Convert predictions classes to one hot vectors
Y_pred_classes = np.argmax(Y_pred,axis = 1)
# Convert validation observations to one hot vectors
Y_true = np.argmax(y_test,axis = 1)  # compute the
confusion matrix confusion_mtx =
confusion_matrix(Y_true, Y_pred_classes)
# plot the confusion matrix
plot_confusion_matrix(confusion_mtx, classes = range(10))
 import
sklearn
print(sklearn.metrics.classification_report(Y_true, Y_pred_classes))

print(model.summary())
```

```
# Final evaluation of the model metrics =
model.evaluate(x_test, y_test, verbose=0)
print("Metrics (Test loss &Test Accuracy) : ")
print(metrics)
 plt.imshow(x_test[5100])
 import numpy as np
print(np.argmax(prediction, axis=1))
 np.argmax(y_test[5100:5101]) #printing the actual
labels
# Save the model model.save('models/mnistCNN.h5')
```

GitHub & Project Demo Link

GitHub Link : https://github.com/IBM-EPBL/IBM-Project-899-1658329455

Demo Video : https://drive.google.com/file/d/127h-inurh6KV55ThNbArY3Xqeryo3sHS/view?usp=share_link