A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

A NALAIYATHIRAN PROJECT REPORT

SUBMITTED BY

PNT2022TMID46193

VIJI.P	**815119106046**
GAYATHRI.D	**815119106012**
SOWNDHARYA.S	**815119106040**
KEERTHANA.R	**815119106018**

In partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

ELECTRONICS AND COMMUNICATION ENGINEERING

HX8001 -"PROFESSIONAL READINESS FOR INNOVATION, EMPLOYBILITY AND ENTREPRENEURSHIP

DHANALAKSHMI SRINIVASAN INSTITUTE OF TECHNOLOGY
SAMAYAPURAM , TIRUCHIRAPPALLI

ANNA UNIVERSITY;; CHENNAI 600025
2022-2023

TABLE OF CONTENTS

A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

2. LITERATURE SURVEY

c. EXISTING PROBLEM

d. REFERENCES

e. PROBLEM STATEMENT DEFINITION

3. IDEATION & PROPOSED SOLUTION

f. EMPATHY MAP CANVAS

g. IDEATION & BRAINSTORMING

h. PROPOSED SOLUTION

i. PROBLEM SOLUTION FIT

4. REQUIREMENT ANALYSIS

j. FUNCTIONAL REQUIREMENTS

k. NON FUNCTIONAL REQUIREMENTS

5. PROJECT DESIGN

l. DATA FLOW DIAGRAM

m. SOLUTION & TECHNICAL ARCHITECTURE

n. USER STORIES

6. PROJECT PLANNING AND SCHEDULING

o. SPRINT PLANNING AND ESTIMATION

p. SPRINT DELIVERY SCHEDULE

7. CODING & SOLUTIONING

a. FEATURE 1

b. FEATURE 2

c. DATABASE SCHEME

CHAPTER 1

INTRODUCTION

a. PROJECT OVERVIEW

Machine learning and deep learning play an important role in computer technology and artifi cial 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 recognise 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.

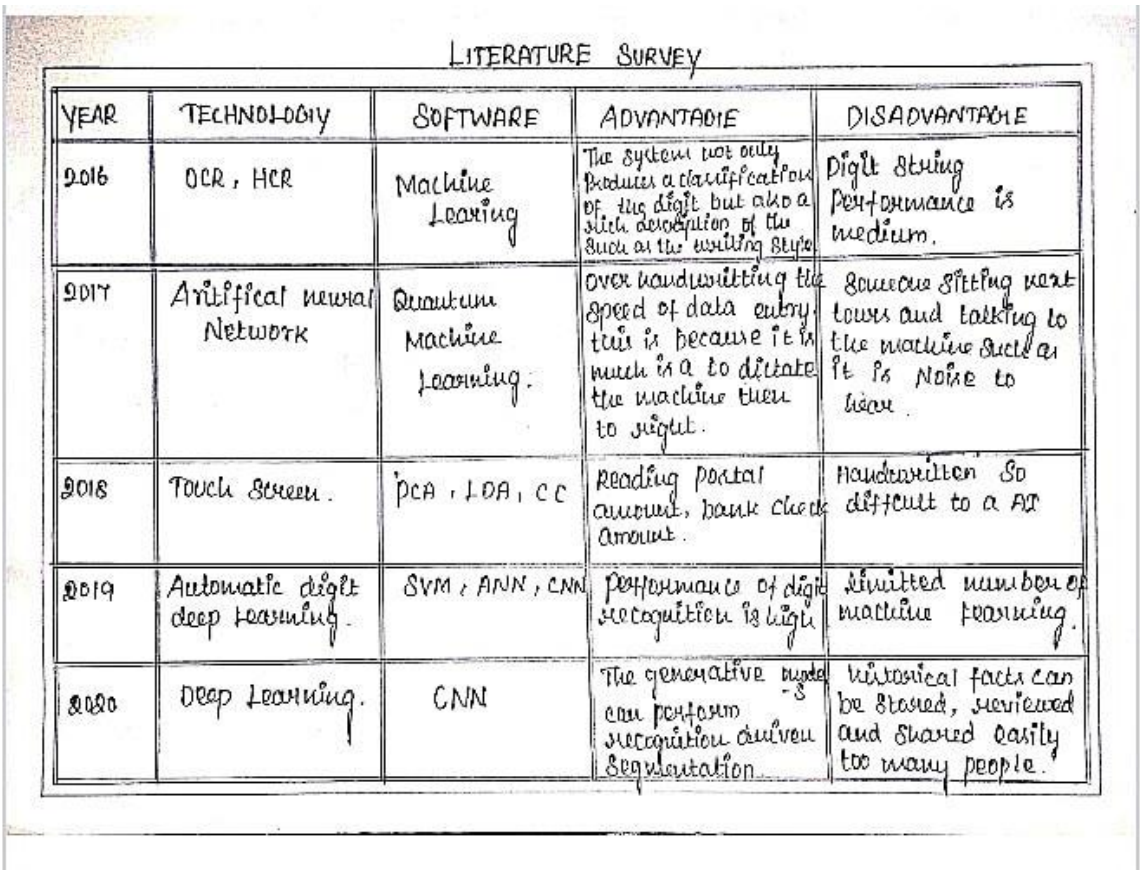
b. PURPOSE

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

CHAPTER 2

LITERATURE SURVEY

A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM



LITERATURE SURVEY

YEAR	TECHNOLOGIY	SOFTWARE	ADVANTAGIE	DISADVANTAGIE
2016	OCR, HCR	Machine Learing	The system not only produces a classification of the digit but also a stick description of the such as the writing style	Digit String Performance is medium.
2017	Artifical neural Network	Quantum Machine Learning.	over handwritting the speed of data entry this is because it is much is a to dictate the machine then to right.	Someone sitting next town and talking to the machine such as it is Noise to hear.
2018	Touch Screen.	PCA, LDA, CC	Reading postal amount, bank check Amount.	Handwritten So difficult to a AI
2019	Automatic digit deep learning.	SVM, ANN, CNN	performance of digit recognition is high	limitted number of machine learning.
2020	Deep Learning.	CNN	The generative model can perform recognition driven Segmentation	historical facts can be stored, reviewed and shared easily too many people.

a. EXISTING PROBLEM

The fundamental problem with handwritten digit recognition is that handwritten digits do not always have the same size, width, orientation, and margins since they vary from person to person. Additionally, there would be issues with identifying the numbers because of similarities between numerals like 1 and 7, 5 and 6, 3 and 8, 2 and 5, 2 and 7, etc. Finally, the individuality and variation of each individual's handwriting infl uence the structure and appearance of the digits.

b. REFERENCES

Improved Handwritten Digit Recognition Using Convolutional Neural Networks (CNN) (2020)

A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

Ahlawat, Savita and Choudhary, Amit and Nayyar, Anand and Singh, Saurabh and Yoon, Byungun

This paper's primary goal was to enhance handwritten digit recognition ability. To avoid diffi cult pre-processing, expensive feature extraction, and a complex ensemble (classifi er combination) method of a standard recognition system, they examined different convolutional neural network variations. Their current work makes suggestions on the function of several hyper-parameters through thorough evaluation utilizing an MNIST dataset. They also confi rmed that optimizing
hyper-parameters is crucial for enhancing CNN architecture performance. With the Adam optimizer for the MNIST database, they were able to surpass many previously published results with a recognition rate of 99.89%. Through the trials, it is made

abundantly evident how the performance of handwritten digit recognition is affected by the number of convolutional layers in CNN architecture. According to the paper, evolutionary algorithms can be explored for optimizing convolutional fi lter kernel sizes, CNN learning parameters, and the quantity of layers and learning rates.

An Efficient And Improved Scheme For Handwritten Digit Recognition Based On Convolutional Neural Network (2019)

Ali, Saqib and Shaukat, Zeeshan and Azeem, Muhammad and Sakhawat, Zareen and Mahmood, Tariq and others

This study uses rectifi ed linear units (ReLU) activation and a convolutional neural network (CNN) that incorporates the Deeplearning4j (DL4J) architecture to recognize handwritten digits. The proposed CNN framework has all the necessary parameters for a high level of MNIST digit classifi cation accuracy. The system's training takes into account the time

factor as well. The system is also tested by altering the number of CNN layers for additional accuracy verifi cation. It is important to note that the CNN architecture consists of two convolutional layers, the fi rst with 32 fi lters and a 5x5 window size and the second with 64 fi lters and a 7x7 window size. In comparison to earlier proposed systems, the experimental fi ndings show that the proposed CNN architecture for the MNIST dataset demonstrates great performance in terms of time and accuracy. As a result, handwritten numbers are detected with a recognition rate of 99.89% and high precision (99.21%) in a short amount of time.

Improved Handwritten Digit Recognition Using Quantum K-Nearest Neighbor Algorithm (2019)

Wang, Yuxiang and Wang, Ruijin and Li, Dongfen and Adu-Gyamfi, Daniel and Tian, Kaibin and Zhu, Yixin

The KNN classical machine learning technique is used in this research to enable quantum parallel computing and superposition. They used the KNN algorithm with quantum acceleration to enhance handwritten digit recognition. When dealing with more complicated and sizable handwritten digital data sets, their suggested method considerably lowered the computational time complexity of the traditional KNN algorithm. The paper offered a theoretical investigation of how quantum concepts can be applied to machine learning. Finally, they established a fundamental operational concept and procedure for machine learning with quantum acceleration.

Handwritten Digit Recognition Using Machine And Deep Learning Algorithms (2021)

Pashine, Samay and Dixit, Ritik and Kushwah, Rishika

In this study, they developed three deep and machine learning-based models for handwritten digit recognition using MNIST datasets. To determine which model was the most accurate, they compared them based on their individual properties.

Support vector machines are among the simplest classifi ers, making them faster than other algorithms and providing the highest training accuracy rate in this situation. However, due to their simplicity, SVMs cannot categorize complicated and ambiguous images as accurately as MLP and CNN algorithms can. In their research, they discovered that CNN produced the most precise outcomes for handwritten digit recognition. This led them to the conclusion that CNN is the most effective

solution for all types of prediction issues, including those using picture data. Next, by comparing the execution times of the algorithms, they determined that increasing the number of epochs without changing the confi guration of the algorithm is pointless due to the limitation of a certain model, and they discovered that beyond a certain number of epochs, the model begins over-fi tting the dataset and provides biased predictions.

a. PROBLEM STATEMENT DEFINITION

For years, the traffi c department has been combating traffi c law violators.

These offenders endanger not only their own lives, but also the lives of other individuals. Punishing these offenders is critical to ensuring that others do not become like them. Identifi cation of these offenders is next to impossible because it is impossible for the average individual to write down the license plate of a reckless driver. Therefore, the goal of this project is to help the traffi c department identify these offenders and reduce traffi c violations as a result

.

CHAPTER 3

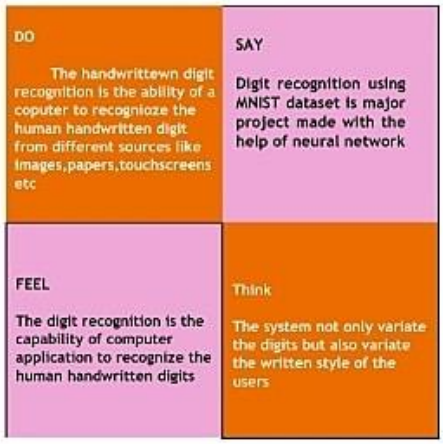
IDEATION AND PROPOSED SOLUTION

a. EMPATHY MAP CANVAS

A Novel Method For Handwritten Digit Recognition System

objective&Goals.

To ensure effective and reliable approaches for recogniton of handwritten digits and make banking operations earier and error free.

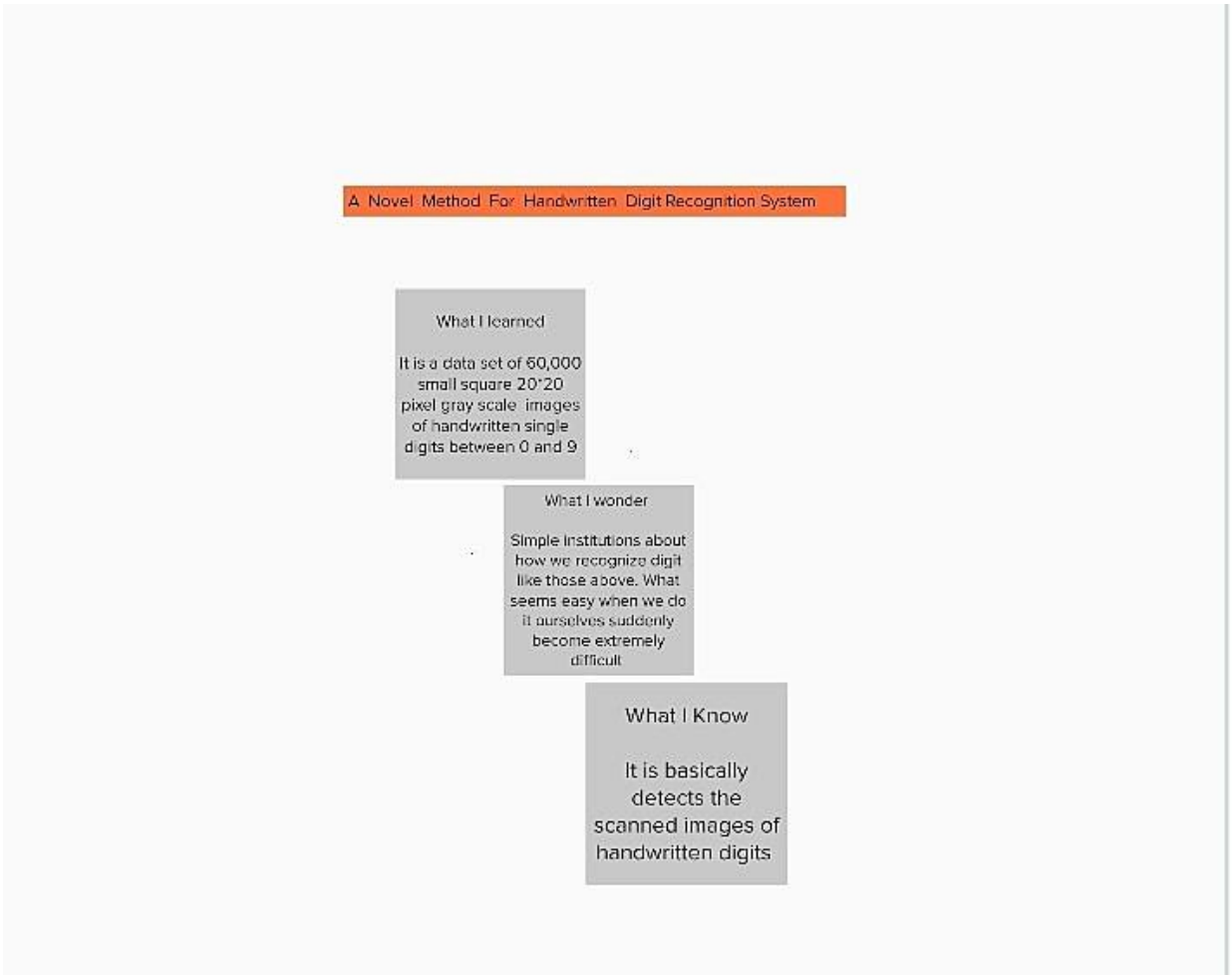


GAINS: The system not only produces a classification of the digit but also a rich description of type instantiation parameter which can yield information such as writing style

PAINS:
It is not done in real time as a person writes and therefore not appropriate for immediate text input
Handwritten number recogintion we face numerous challenges

a. IDEATION & BRAINSTORMING

A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM



A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

Project Design Phase-I
Proposed Solution Template

Date	27 September 2022
Team ID	PNT2022TMID46193
Project Name	A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM
Maximum Marks	2 Marks

Proposed Solution :

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	• It is easy for the human to perform 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 element present in the images. • The **handwritten digit recognition** is the capability of computer applications to **recognize** the human **handwritten digits**. • It is a hard task for the **machine** because **handwritten digits** are not perfect and can be made with many different shapes and sizes. • The **handwritten digit recognition system** is a way to tackle this problem which uses the image of a **digit** and recognizes the **digit** present in the image. • In this competition, the goal is to correctly identify digits from a dataset of tens of thousands of handwritten images and experiment with different algorithms to learn what works well and how techniques compare.

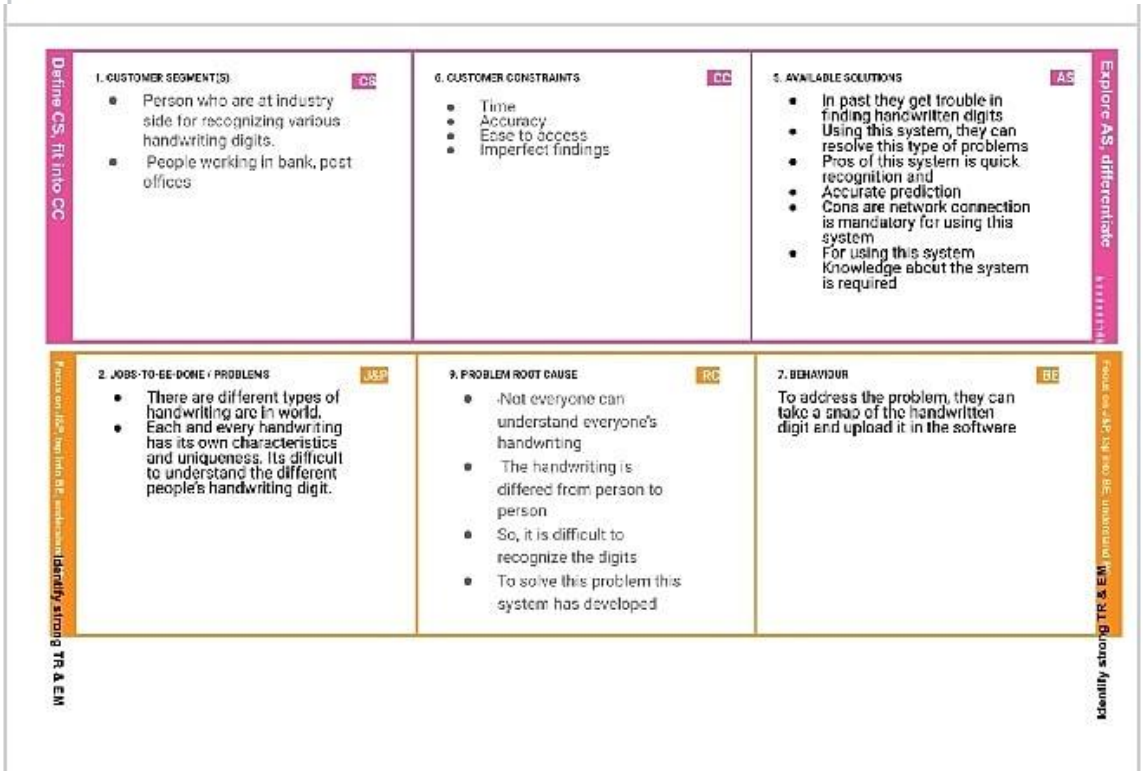
A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

2.	Idea / Solution description	• The algorithm used is Convolution Neural Network(CNN). This will prepare the trained model which will be used to classify the digits present in the test data. Thus, we can classify the digits present in the images as: Class 0,1,2,3,4,5,6,7,8,9. • MNIST is a dataset which is widely used for handwritten digit recognition. The dataset consist of 60,000 training images and 10,000 test images.
		• The artificial neural neworks can all most mimic the human brain and are a key ingredient in image processing field.
3.	Novelty / Uniqueness	• This project introduces an operative strategy for dealing with novelty in the handwritten visual recognition domain. A perfect transcription agent would be able to distinguish known and unknown characters in a picture, as well as determine any aesthetic variations that may occur inside or between texts. The existence of novelty has shown to be a major stumbling block for even the most robust machine learning-based algorithms for these activities. • Novelty in handwritten papers might include, among other things, a change in the writer, character properties, writing attributes, or overall document appearance. Instead of examining each element separately, we believe that an integrated agent capable of processing known characters and novelties concurrently is a superior technique. The handwritten digit recognition problem can be seen as a subtask of the optical character recognition (OCR) problem.

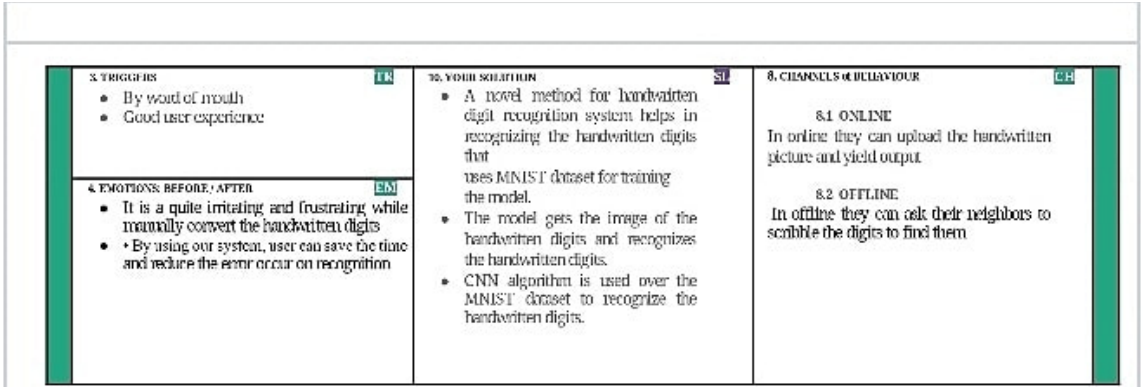
4.	Social Impact / Customer Satisfaction	• There are many benefits associated with the handwriting recognition system. In addition to reading postal addresses and bank check amounts, it is also useful for reading forms. Furthermore, it's used in fraud detection because it makes it easy to compare two texts and determine which one is a copy. As a result, this system fulfills customers' expectations, as it is a novel method for recognizing handwritten digits, ensuring high accuracy for the model and meeting all customer expectations. Users will save a lot of time and effort if the system provides various synonyms for the words recognized. Due to the fact that the users in rural areas will be using their own regional language, this proposed system should be able to detect those digits as well. As the system is being used in socially crowded places such as banks to check amounts, it should be fast and reliable.
		As it is designed to solve real-world problems, it should be highly reliable and trustworthy in every way, and users throughout the world should be able to use it effectively.
5.	Business Model (Revenue Model)	• The applications where these handwritten digit recognition can be used are Banking sector where it can be used to maintain the security pin numbers, it can be also used for blind peoples by using sound output. • Some of the research areas include signature verification, bank check processing, postal address interpretation from envelopes etc.

A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

6.	Scalability of the Solution	• One of the approaches to make the handwritten digit recognition system scalable is to make use of cloud-native methods. For example, one of the cloud solutions for making AI scalable is IBM Cloud. IBM Cloud Build helps run and manage AI models, optimize decisions at scale across any cloud. The advantage of using cloud to make solutions scalable is that we can deploy our AI application on the specific cloud environment that best supports our business needs. We can take advantage of built-in security capabilities and AI model monitoring. We can Automate AI lifecycles with ModelOps pipelines, deploy and run models through oneclick integration and also prepare and build models visually and programmatically. Looking at these advantages, we can drive better business outcomes by optimizing our decisions and also make our solution scalable using cloud



A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM



CHAPTER 4

FUNCTIONAL REQUIREMENTS

Project Design Phase-II
Solution Requirements (Functional & Non-functional)

Team ID	PNT2022TMID46193
Project Name	A Novel Method for Handwritten Digit Recognition System

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.

a. NON FUNCTIONAL REQUIREMENT

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.
NFR-5	**Availability**	

CHAPTER 5
PROJECT DESIGN

DATA FLOW DIAGRAM

P

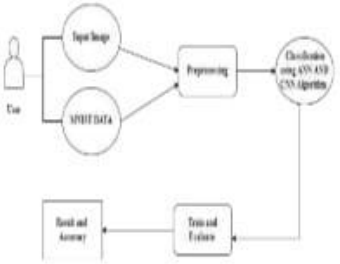
roject Design Phase-II
Data Flow Diagram & User Stories

Date	14 October 2022
Team ID	PNT2022TMID46193
Project Name	Project - A Novel Method For Handwritten Digit Recognition System
Maximum Marks	4 Marks

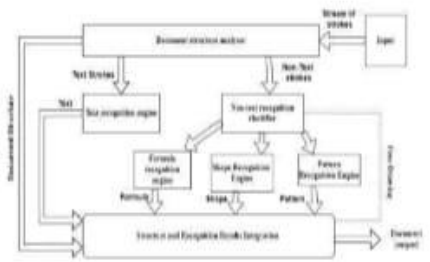
Data Flow Diagrams:

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

Example: (Simplified) FLOW



Example:DFDLevel0(IndustryStandard)



A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

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 (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 user-friendly 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-5	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 (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 user-friendly 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

a. **SOLUTION & TECHNICAL ARCHITECTURE**

A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

Project Design Phase-II
Technology Stack (Architecture & Stack)

Date	14 October 2022
Team ID	PNT2022TMID46193
Project Name	Project - A Novel Handwritten Digit Recognition System
Maximum Marks	4 Marks

Technical Architecture:

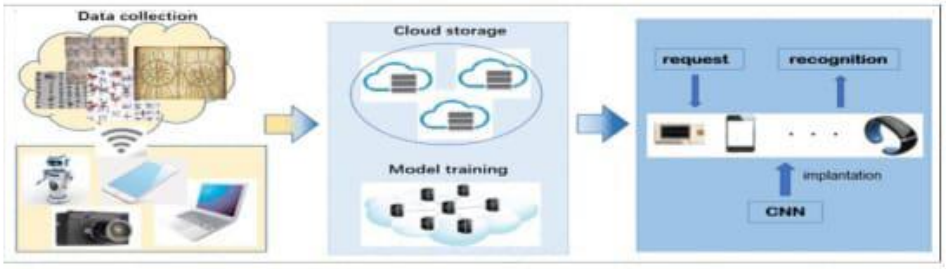


Table-1 : Components & Technologies:

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

Table-2: Application Characteristics:

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	Characteristics	Description	Technology
3.	Scalable Architecture	Justify the scalability of architecture	3 – tier, Micro services
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

CHAPTER 6

PROJECT PLANNING AND SCHEDULING

a. SPRINT PLANNING AND ESTIMATION

A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

Project Planning Phase

Project Planning Template (Product Backlog, Sprint Planning, Stories, Story points)

Date	22 October 2022
Team ID	PNT2022TMID46193
Project Name	Project - A Novel Method For Handwritten Digit Recognition System
Maximum Marks	8 Marks

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	Understanding the data	USN-1	Importing the required library	1	Medium	• D.GAYATHRI • R.KEERTHANA • P.VIJI
Sprint 1		USN-2	Loading the data	2	High	• S.SOWNDHARAYA • P. VIJI
Sprint-1		USN-3	Analyzing the data	2	High	• D.GAYATHRI • R.KEERTHANA
Sprint-1		USN-4	Reshaping the data	1	Medium	• S.SOWNDHARYA • P.VIJI
Sprint 1		USN-5	Applying one hot encoding	1	Low	• P.VIJI
Sprint 2	Model building-1	USN-6	Add CNN layers	5	High	• R.KEERTHANA
Sprint-2		USN-7	Compiling the model	2	Low	• D.GAYATHRI
Sprint-2		USN-8	Train the model	7	High	• S.SOWNDHARYA
Sprint-2		USN-9	Observing the metrix	3	Medium	• D.GAYATHRI
Sprint-2		USN-10	Test the model	3	Medium	• P.VIJI
Sprint-3	Model building -2	USN-11	Observing the metrics	4	High	• S.SOWNDHARYA
Sprint 3		USN 12	Test the model	2	Low	• R.KEERTHANA
Sprint-3		USN-13	Save the model	5	High	• D.GAYATHRI
Sprint-3		USN-14	Test with saved model	2	Low	• P.VIJI
Sprint 3		USN 15	Create an HTML file	8	High	• R.KEERTHANA
Sprint-4	Application building	USN-16	Build python code part1	5	High	• G.SOWNDHARYA

A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

Sprint-4		USN-17	Build python code part2	2	Medium	• S.GOWNDHARYA
Sprint-4		USN-18	Run the application	8	High	• R.KEERTHANA
Sprint-4	Train the model on IBM	USN-19	Register for IBM cloud	11	High	• D.GAYATHRI
Sprint-4		USN-20	Train the model on IBM	4	High	• P.VIJI

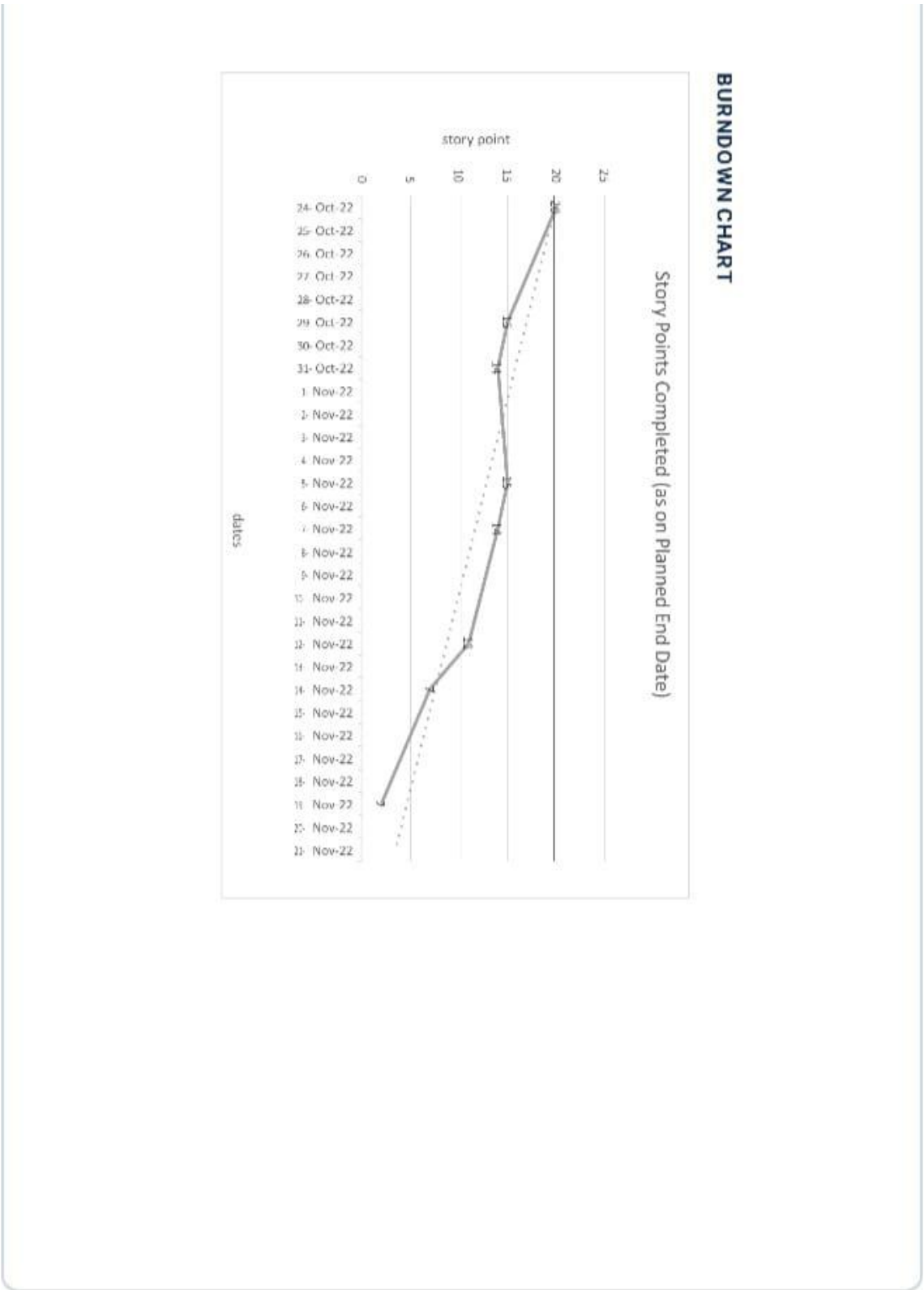
Project Tracker, Velocity & Burndown Chart: (4 Marks)

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:

$$AV = \frac{sprint\ duration}{velocity}$$

AV = 6/3 = 2



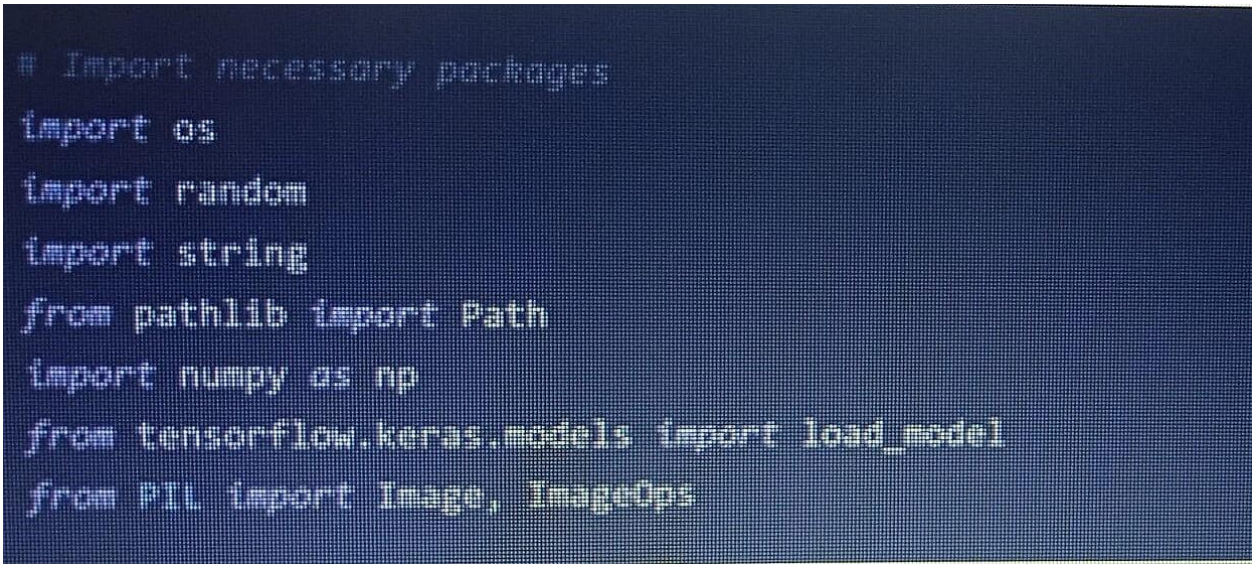
a. **SPRINT DELIVERY SCHEDULE**

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

CHAPTER 7

CODING & SOLUTIONING

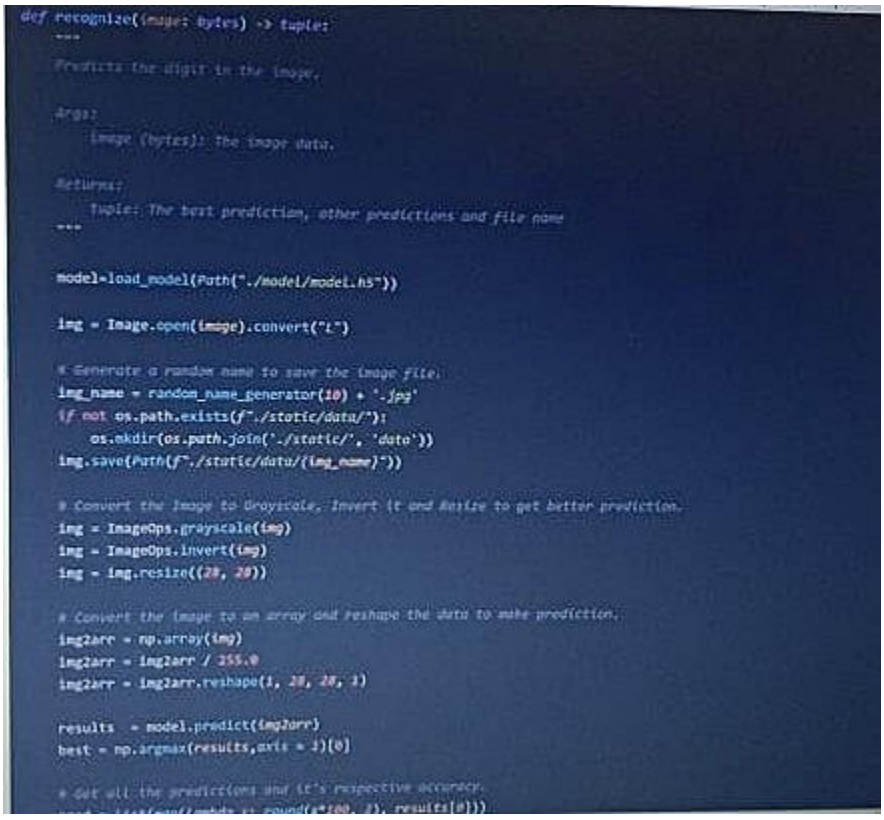


```
# Import necessary packages
import os
import random
import string
from pathlib import Path
import numpy as np
from tensorflow.keras.models import load_model
from PIL import Image, ImageOps
```

```python
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))
```



```python
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/model.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.
```

CHAPTER 8

TESTING

TEST CASES

Test case ID	Feature Type	Component	Test Scenario	Expected Result	Actual Result	Status
HP_TC_001	UI	Home Page	Verify UI elements in the Home Page	The Home page must be displayed properly	Working as expected	PASS
HP_TC_002	UI	Home Page	Check if the UI elements are displayed properly in different screen sizes	The Home page must be displayed properly in all sizes	The UI is not displayed properly in screen size 2560 x 1801 and 768 x 630	FAIL
HP_TC_003	Functional	Home Page	Check if user can upload their fi le	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 fi les	The application should not allow user to select a non image fi le	User is able to upload any fi le	FAIL
HP_TC_005	Functional	Home Page	Check if the page redirects to the result page once the input is given	The page should redirect to the results page	Working as expected	PASS

BE_TC_001	Functional	Backend	Check if all the routes are working properly	All the routes should properly work	Working as expected	PASS
M_TC_001	Functional	Model	Check if the model can handle various image sizes	The model should rescale the image and predict the results	Working as expected	PASS
M_TC_002	Functional	Model	Check if the model predicts the digit	The model should predict the number	Working as expected	PASS
M_TC_003	Functional	Model	Check if the model can handle complex input image	The model should predict the number in the complex image	The model fails to identify the digit since the model is not built to handle such data	FAIL
RP_TC_001	UI	Result Page	Verify UI elements in the Result Page	The Result page must be displayed properly	Working as expected	PASS
RP_TC_002	UI	Result Page	Check if the input image is displayed properly	The input image should be displayed properly	The size of the input image exceeds the display container	FAIL
RP_TC_003	UI	Result Page	Check if the result is displayed properly	The result should be displayed properly	Working as expected	PASS

RP_TC_004	UI	Result Page		The other predictions should be displayed properly	Working as expected	PASS

a. USER ACCEPTANCE TESTING

i. DEFECT ANALYSIS

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

i. TEST CASE ANALYSIS

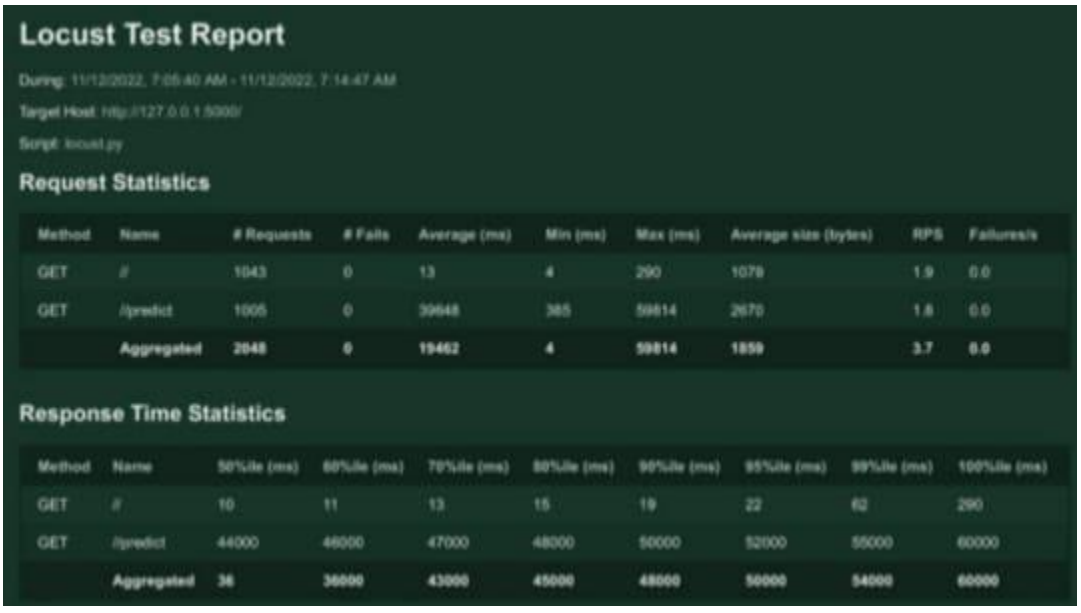
Section	Total	Not Tested	Fail	Pass

	Cases			
Client Application	10	0	3	7
Security	2	0	1	1
Performance	3	0	1	2
Exception Reporting	2	0	0	2

CHAPTER 9

RESULTS

a. PERFORMANCE METRICS



Locust Test Report

During: 11/12/2022, 7:05:40 AM - 11/12/2022, 7:14:47 AM

Target Host: http://127.0.0.1:5000/

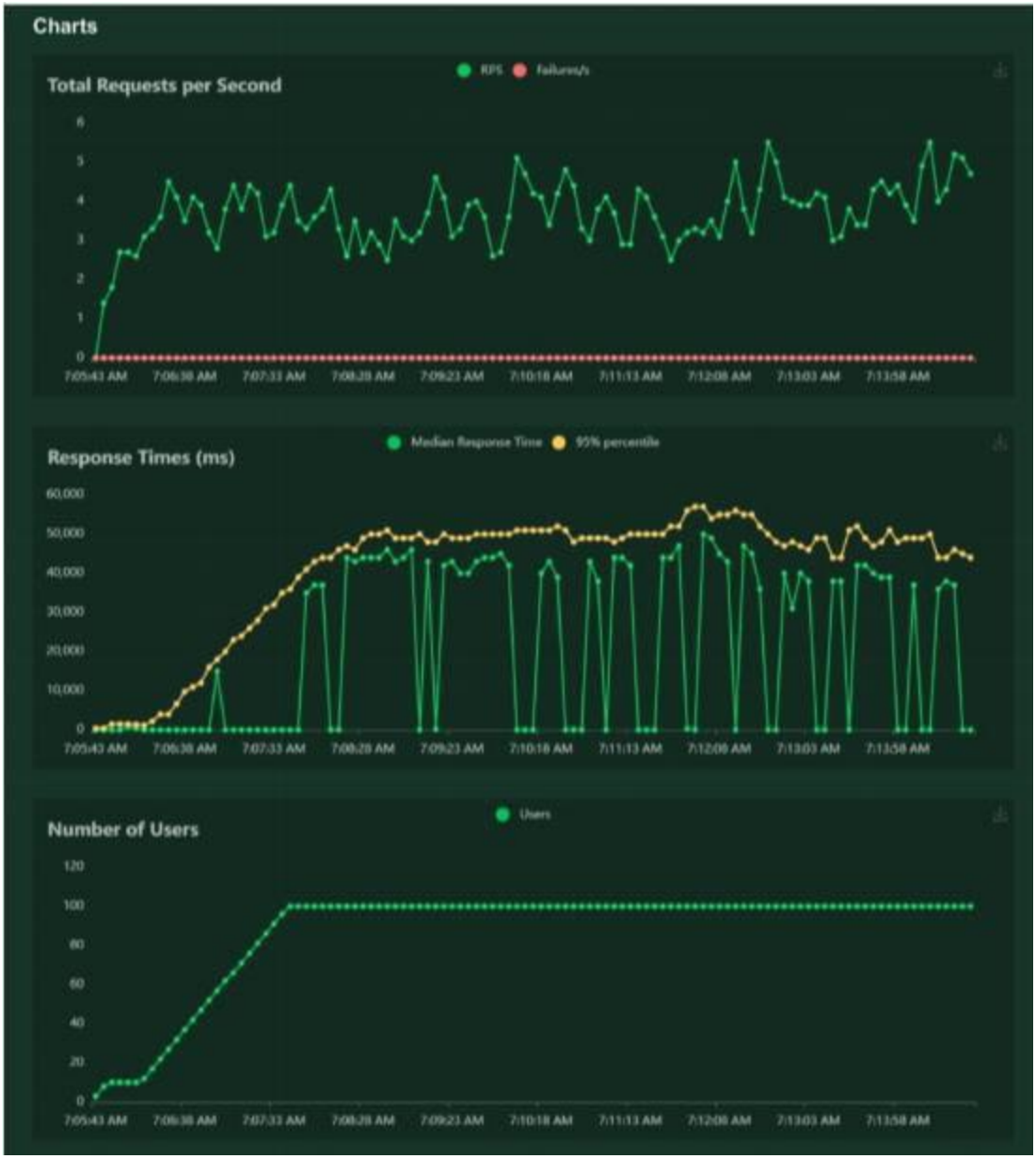
Script: locust.py

Request Statistics

Method	Name	# Requests	# Fails	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	RPS	Failures/s
GET	/	1043	0	13	4	290	1079	1.9	0.0
GET	/predict	1005	0	39648	385	59814	2670	1.8	0.0
	Aggregated	2048	0	19462	4	59814	1859	3.7	0.0

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	/	10	11	13	15	19	22	62	290
GET	/predict	44000	46000	47000	48000	50000	52000	55000	60000
	Aggregated	36	36000	43000	45000	48000	50000	54000	60000



CHAPTER 10

ADVANTAGES & DISADVANTAGES

ADVANTAGES

i. Reduces manual work

ii. More accurate than average human
iii. Capable of handling a lot of data
iv. Can be used anywhere from any device

DISADVANTAGES

i. Cannot handle complex data
ii. All the data must be in digital format
iii. Requires a high performance server for faster predictions
iv. Prone to occasional errors

CHAPTER 11

CONCLUSION

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

CHAPTER 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:

i. Add support to detect from digits multiple images and save the results
ii. Add support to detect multiple digits
iii. Improve model to detect digits from complex images
iv. 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 benefi t several industries and reduce the workload on many workers, enhancing overall work effi ciency.

APPENDIX

SOURCE CODE

MODEL CREATION

A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

```python
# Load the necessary packages
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from keras.utils import np_utils
from tensorflow.keras.datasets import mnist
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, Dense, Flatten
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.models import load_model
from PIL import Image, ImageOps

# Load the data
(X_train, y_train), (X_test, y_test) = mnist.load_data()

# Data pre-processing
X_train = X_train.reshape(60000, 28, 28, 1).astype('float32')
X_test = X_test.reshape(10000, 28, 28, 1).astype('float32')

number_of_classes = 10
Y_train = np_utils.to_categorical(y_train, number_of_classes)
Y_test = np_utils.to_categorical(y_test, number_of_classes)
```

```python
# Create the model
model = Sequential()
model.add(Conv2D(64, (3, 3), input_shape=(28, 28, 1), activation="relu"))
model.add(Conv2D(32, (3, 3), activation="relu"))
model.add(Flatten())
model.add(Dense(number_of_classes, activation="softmax"))

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

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

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

# Save the model
model.save("model.h5")
```

```python
# Test the saved model
model=load_model("model.h5")

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

FLASK APP

A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

```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

```python
# Import necessary packages
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
```

```python
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))
```

A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

```python
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/model.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
```

HOME PAGE (HTML)

A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

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

* {
    padding: 0;
    margin: 0;
}

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

```
            <div class="heading">
                <h1 class="heading__main">Handwritten Digit Recognizer</h1>
                <h2 class="heading__sub">Easily analyze and detect handwritten digits</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>Select 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>
```

A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

```css
.form-wrapper .upload {
    display: flex;
    justify-content: center;
    align-items: center;
    width: 8rem;
    height: -webkit-fit-content;
    height: -moz-fit-content;
    height: fit-content;
    border-radius: 6px;
    color: white;
    background-color: rgb(114, 96, 182);
    box-shadow: 0 5px 10px rgb(146, 135, 247);
}

.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;
}

@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;
    }
}
```

A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

```css
.container {
    width: 100%;
    height: 100%;
    display: flex;
    flex-direction: column;
    justify-content: center;
    align-items: center;
    background-color: white;
}

.heading {
    margin-top: -2rem;
    padding-bottom: 2rem;
    width: fit-content;
    text-align: center;
}

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

.heading .heading__sub {
    font-size: 1rem;
    color: rgb(90, 88, 88);
}

.upload-container {
    box-shadow: 0 0 20px rgb(172, 170, 170);
    width: 40rem;
    height: 25rem;
    padding: 3.5rem;
}

.form-wrapper {
    background-color: rgba(190, 190, 190, 0.5);
    width: 100%;
    height: 100%;
    display: flex;
    border: 1px dashed black;
    justify-content: center;
    align-items: center;
}

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

HOME PAGE (JS)

A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

```
feather.replace(); // Load feather icons

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

select.addEventListener("change", (e) => {
    e.preventDefault();

    form.submit()
    form.style.visibility = "hidden";
    loading.style.display = 'flex';
});
```

PREDICT PAGE (HTML)

A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

```
<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>Prediction</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 Predictions</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>
```

A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

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

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

h1 {
    padding-top: 2rem;
}

.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);
}
```

A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

```css
.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;
    box-shadow: 0 0 7px rgb(158, 157, 157);
}

.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;
    }
}
```

CERTIFICATES

A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

This is to certify that

GAYATHRI D

successfully completed and received a passing grade in

Data Science Tools

(DS0105EN, provided by IBM)

A course on ictacademy.skillsnetwork.site
Powered by IBM Developer Skills Network.

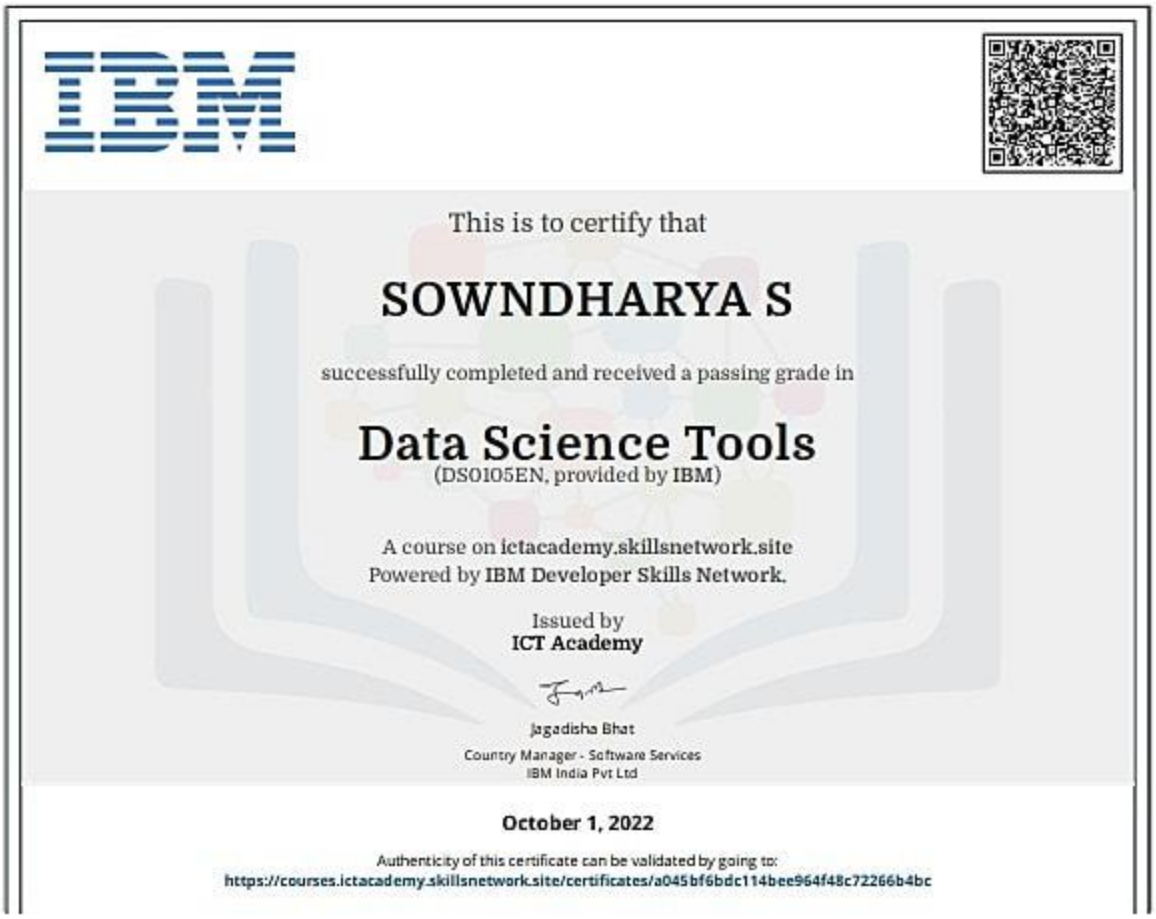
Issued by
ICT Academy

Jagadisha Bhat
Country Manager - Software Services
IBM India Pvt Ltd

October 10, 2022

Authenticity of this certificate can be validated by going to:
https://courses.ictacademy.skillsnetwork.site/certificates/16aabb832daa433db6616cc7a0135b13

A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM



This is to certify that

SOWNDHARYA S

successfully completed and received a passing grade in

Data Science Tools
(DS0105EN, provided by IBM)

A course on ictacademy.skillsnetwork.site
Powered by IBM Developer Skills Network.

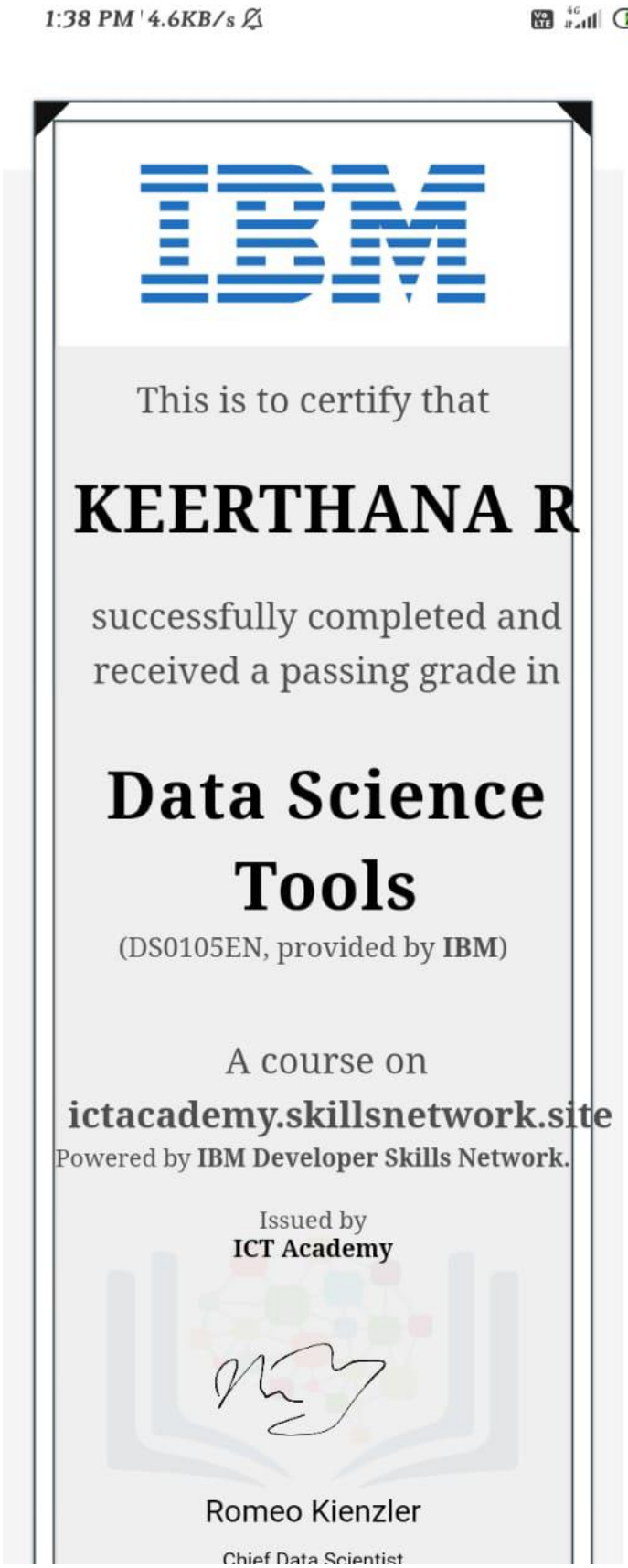
Issued by
ICT Academy

Jagadisha Bhat
Country Manager - Software Services
IBM India Pvt Ltd

October 1, 2022

Authenticity of this certificate can be validated by going to:
https://courses.ictacademy.skillsnetwork.site/certificates/a045bf6bdc114bee964f48c72266b4bc

A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM



A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

This is to certify that

VIJI P

successfully completed and received a passing grade in

Data Science Tools
(DS0105EN, provided by IBM)

A course on ictacademy.skillsnetwork.site
Powered by IBM Developer Skills Network.

Issued by
ICT Academy

Jagadisha Bhat
Country Manager - Software Services
IBM India Pvt Ltd

October 1, 2022

Authenticity of this certificate can be validated by going to:
https://courses.ictacademy.skillsnetwork.site/certificates/dac16fd683d14685a2aa746d732fd1f7

GITHUB

https://github.com/IBM-EPBL/IBM-Project-31149-166019845

PROJECT DEMO LINK

https://youtu.be/0q3FxM9EpZo

A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM