PROJECT REPORT

A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION

submitted by

PNT2022TMID32578

Arunkumar S - 810019205301

Subashkumar S - 810019205098

Santhoshkumar R - 810019205089

Natesh T -810019205501

TABLE OF CONTENTS

1. **INTRODUCTION**
   - 1.1 Project Overview
   - 1.2 Purpose
2. **LITERATURE SURVEY**
   - 2.1 Existing problem
   - 2.2 References
   - 2.3 Problem Statement Definition
3. **IDEATION & PROPOSED SOLUTION**
   - 3.1 Empathy Map Canvas
   - 3.2 Ideation & Brainstorming
   - 3.3 Proposed Solution
   - 3.4 Problem Solution fit
4. **REQUIREMENT ANALYSIS**
   - 4.1 Functional requirement
   - 4.2 Non-Functional requirements
5. **PROJECT DESIGN**
   - 5.1 Data Flow Diagrams
   - 5.2 Solution & Technical Architecture
   - 5.3 User Stories
6. **PROJECT PLANNING & SCHEDULING**
   - 6.1 Sprint Planning & Estimation
   - 6.2 Sprint Delivery Schedule
   - 6.3 Reports from JIRA
7. **CODING & SOLUTIONING (Explain the features added in the project along with code)**
   - 7.1 Sprint 1
   - 7.2 Sprint 2
   - 7.3 Sprint 3

8. TESTING

8.1 Test Cases

8.2 User Acceptance Testing

9. RESULTS

9.1 Performance Metrics

10. ADVANTAGES & DISADVANTAGES

11. CONCLUSION

12. FUTURE SCOPE

13. APPENDIX

Source Code

GitHub & Project Demo Link

CHAPTER 1
INTRODUCTION

1.1 PROJECT OVERVIEW

Handwritten Digit Recognition is the capacity of a computer to interpret the manually written digits from various sources like messages, bank cheques, papers, pictures, and so forth and in various situations for web-based handwriting recognition on PC tablets, identifying number plates of vehicles, handling bank cheques, digits entered in any forms etc. Machine Learning provides various methods through which human efforts can be reduced in recognizing the manually written digits.

Deep Learning is a machine learning method that trains computers to do what easily falls into place for people: learning through examples. With the utilization of deep learning methods, human attempts can be diminished in perceiving, learning, recognizing and in a lot more regions. Using deep learning, the computer learns to carry out classification works from pictures or contents from any document. Deep Learning models can accomplish state-of-art accuracy, beyond the human level performance. The digit recognition model uses large datasets in order to recognize digits from distinctive sources.

1.2 PURPOSE

The main objective was to actualize a pattern characterization method to perceive the handwritten digits provided in the MINIST data set of images of handwritten digits (0-9). The goal of our work is to create a model that will be able to recognize and classify the handwritten digits from images by using concepts of Convolution Neural Network. Though the goal of our research is to create a model for digit recognition and classification, it can also be extended to letters and an individual's handwriting. With high accuracy rates, the model can solve a lot of real life problems.

The main applications are vehicle license-plate recognition, postal letter-sorting services, Cheque truncation system (CTS) scanning and historical document preservation in archaeology departments, old documents automation in libraries and banks, etc. All these areas deal with large databases and hence demand high recognition accuracy, lesser computational complexity and consistent performance of the recognition system.

CHAPTER 2

LITERATURE SURVEY

2.1 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. People can struggle to read others' handwriting. The handwritten digits are not always of the same size, width, orientation as they differ from writing of person to person, so the general problem would be while classifying the digits.

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 influence the structure and appearance of the digits.

2.2 REFERENCES

[1]This paper's primary goal was to enhance handwritten digit recognition ability. To avoid difficult pre-processing, expensive feature extraction, and a complex ensemble (classifier 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 confirmed 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 filter kernel sizes, CNN learning parameters, and the quantity of layers and learning rates.[2]This study uses rectified 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 classification 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 verification. It is important to note that the CNN architecture consists of two convolutional layers, the first with 32 filters and a 5x5 window size and the second with 64 filters and a 7x7 window size. In comparison to earlier proposed systems, the experimental findings 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.[3]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 o ered 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. The KNN algorithm, however, is a method for handling handwritten digit recognition. The challenges mentioned in this study can be solved more e ectively using the deep learning neural network approach.[4]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 classifiers, 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 e ective 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 configuration 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 overfitting the dataset and provides biased predictions.

[1]Improved Handwritten Digit Recognition Using Convolutional Neural Networks (CNN) (2020) Ahlawat, Savita and Choudhary, Amit and Nayyar, Anand and Singh, Saurabh and Yoon, Byungun.
[2]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.
[3] 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, and Zhu, Yixin. [4]Handwritten Digit Recognition Using Machine And Deep Learning Algorithms (2021) Pashine, Samay and Dixit, Ritik and Kushwah, Rishika

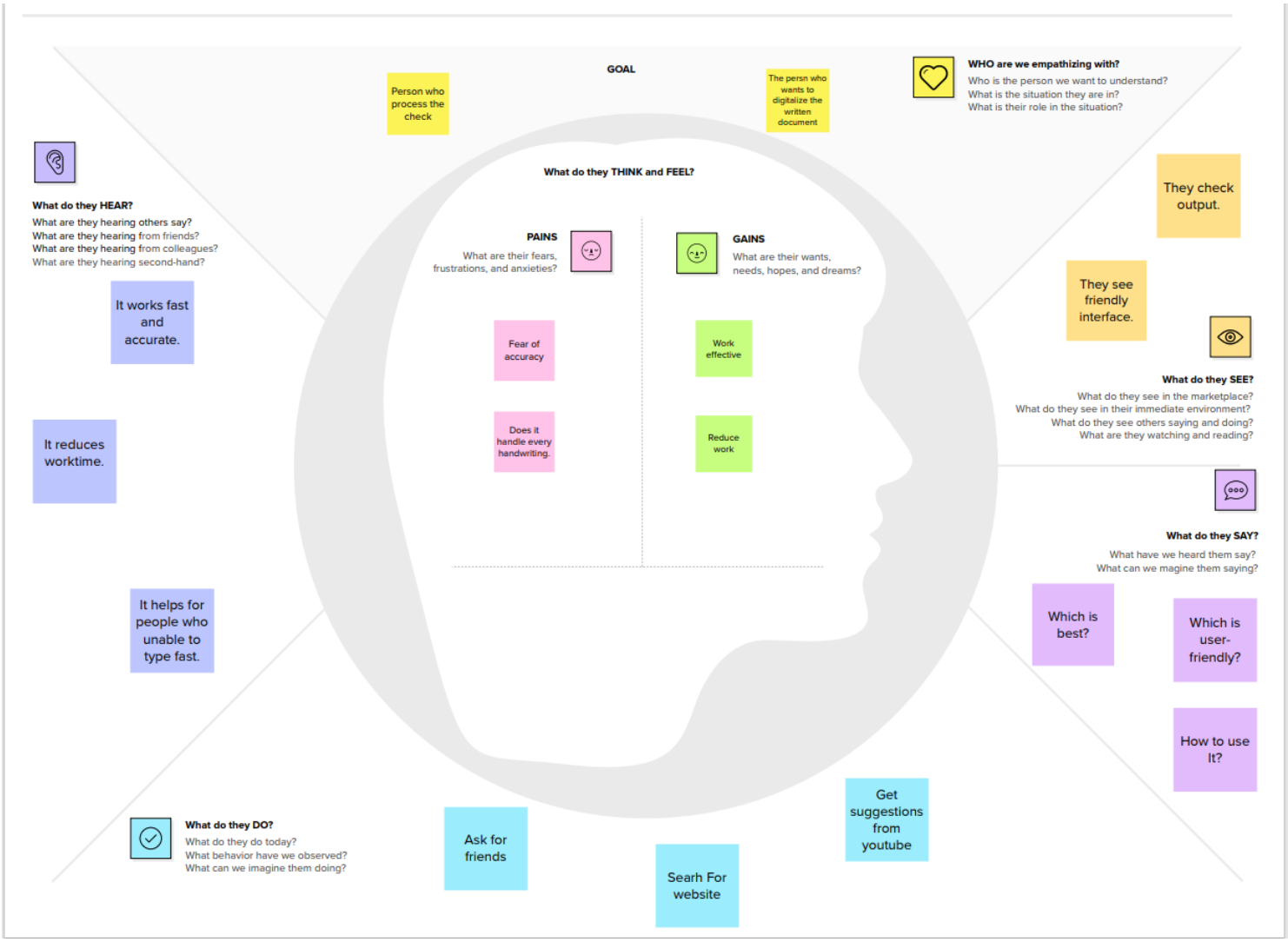
2.3 PROBLEM STATEMENT DEFINITION

It is easy for the human to perform a task accurately by practicing it repeatedly and memorizing it for the next time. Human brain can process and analyse images easily. Also, recognize the different elements present in the images. 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 first-hand what works well and how techniques compare.

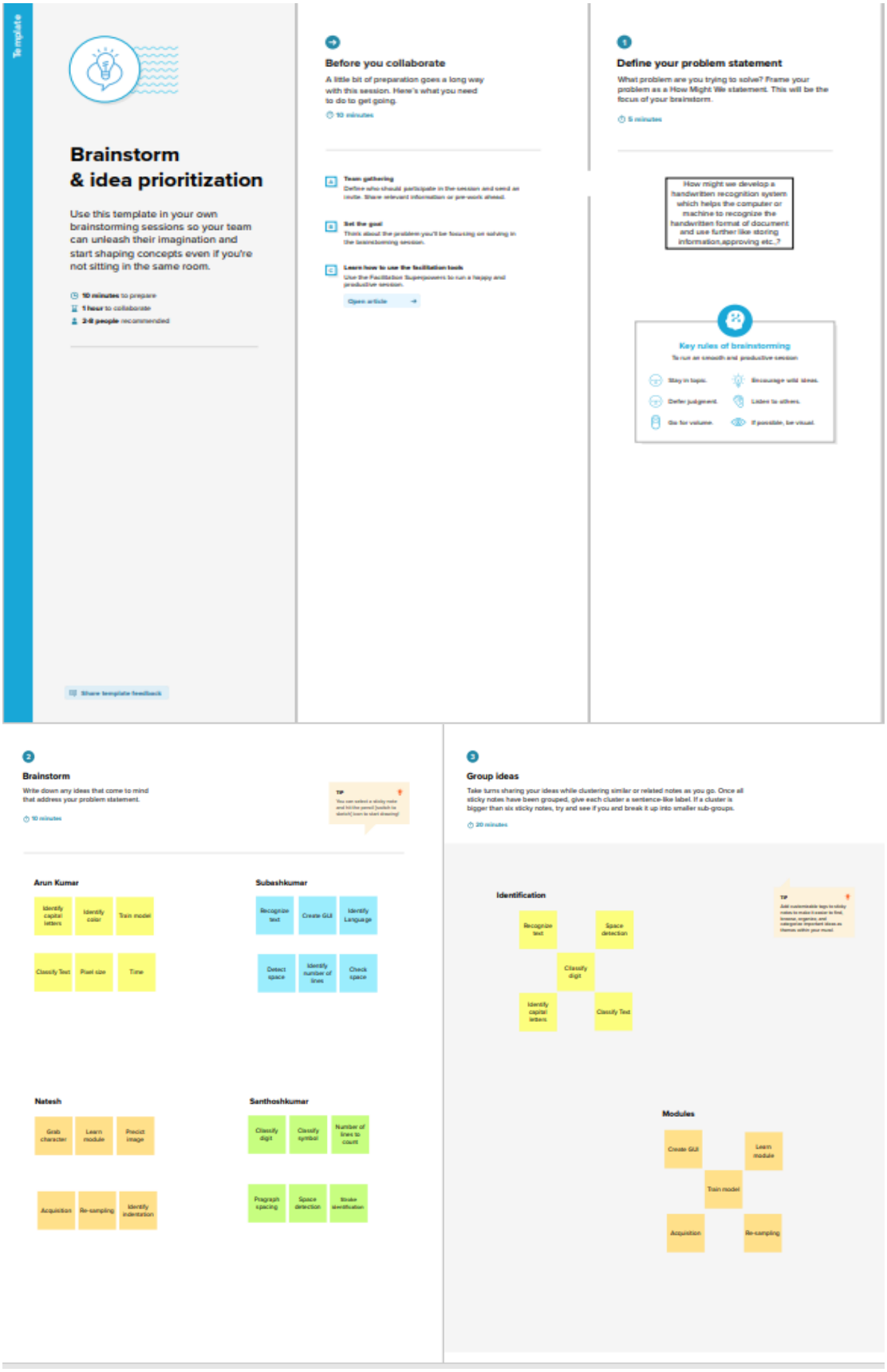
CHAPTER 3

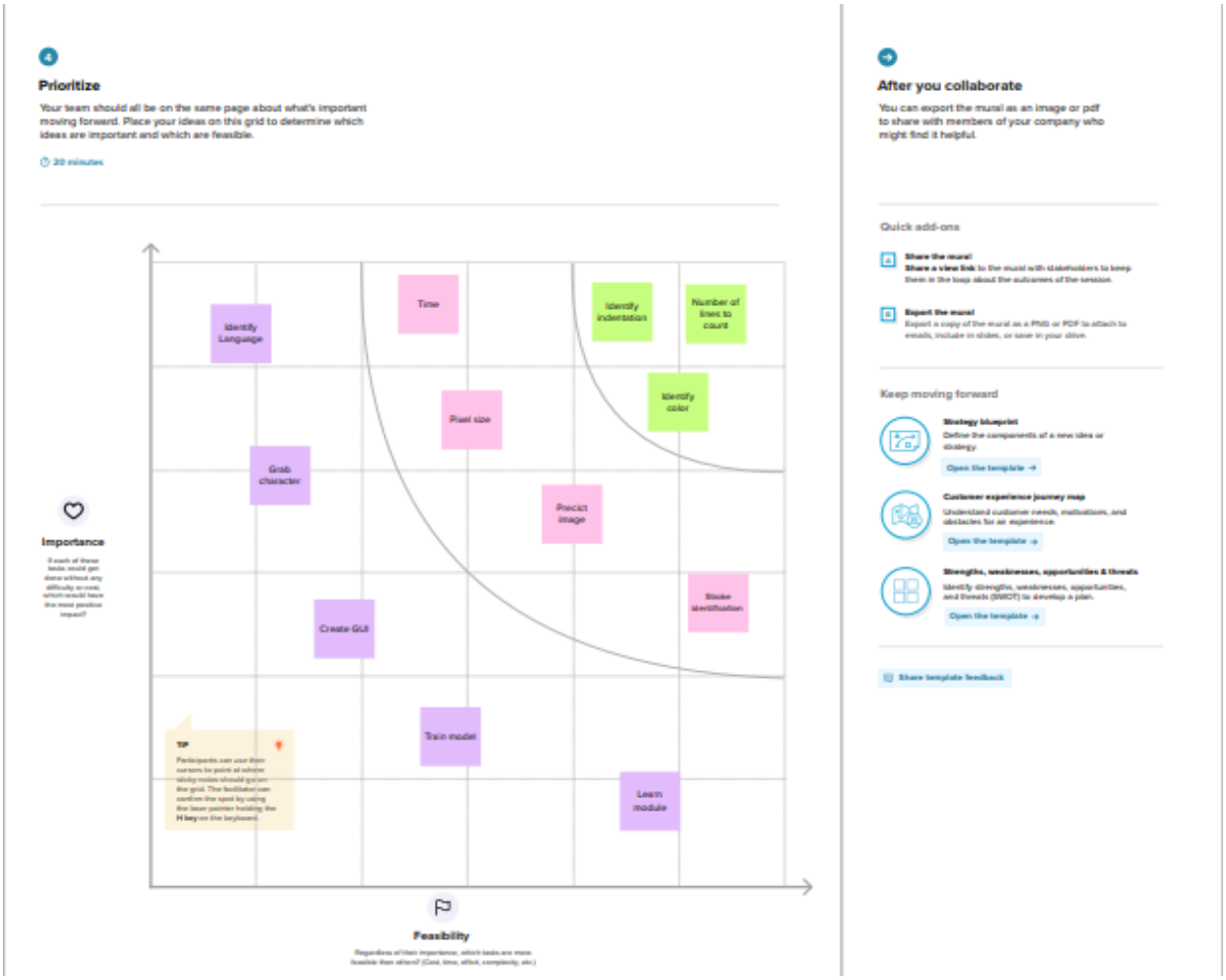
IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS



3.2 IDEATION & BRAINSTORMING



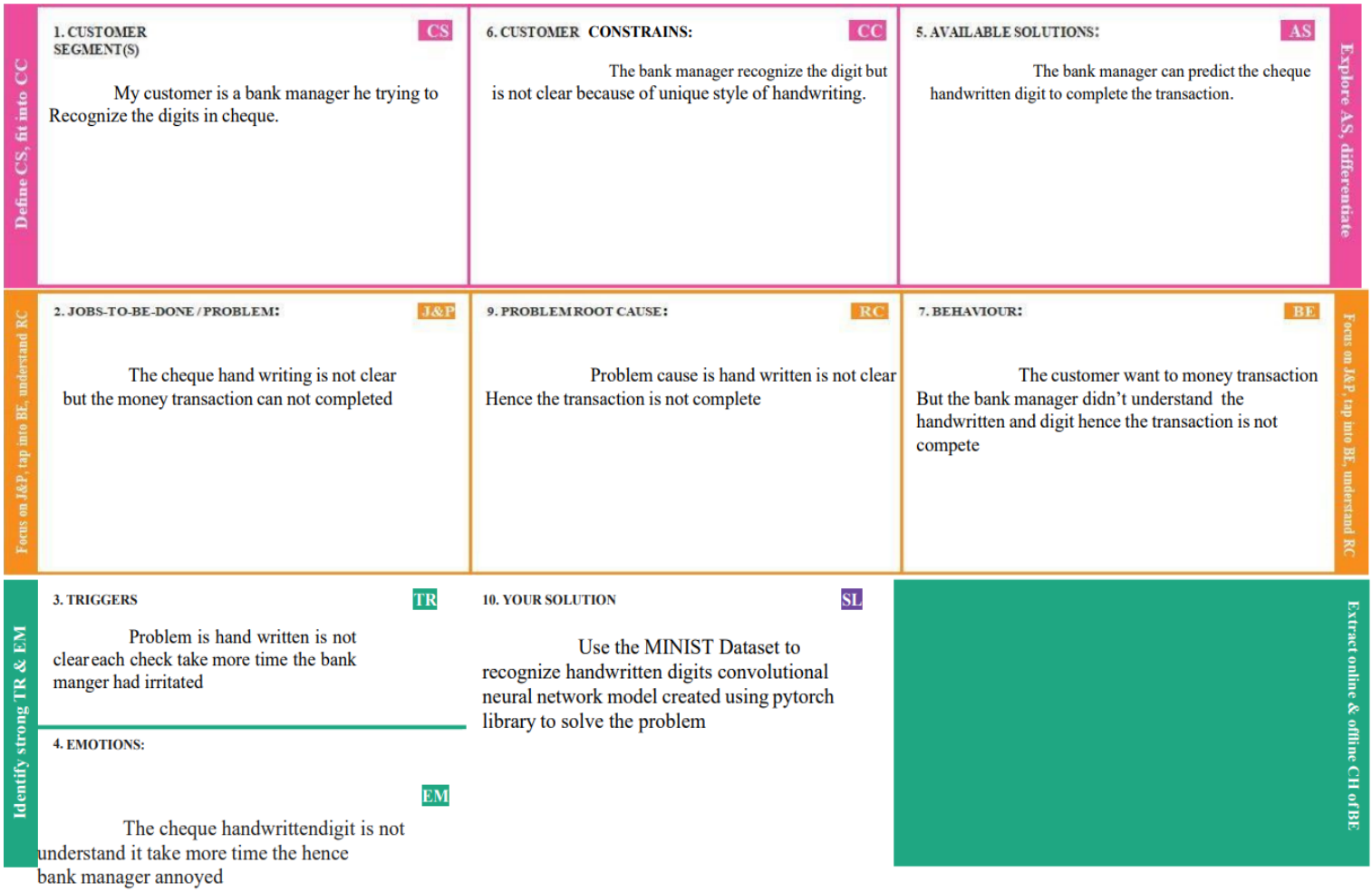

4
Prioritize
Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.
20 minutes
Identify Language
Time
Identify indentation
Number of lines to count
Identify color
Pixel size
Grab character
Predict image
Stroke identification
Create GUI
Train model
Learn module
TIP
Importance
Feasibility
After you collaborate
You can export the mural as an image or pdf to share with members of your company who might find it helpful.
Quick add-ons
Share the mural
Share a view link to the mural with stakeholders to keep them in the loop about the outcomes of the session.
Export the mural
Keep moving forward
Strategy blueprint
Define the components of a new idea or strategy
Open the template
Customer experience journey map
Understand customer needs, motivations, and obstacles for an experience.
Open the template
Strengths, weaknesses, opportunities & threats
Identify strengths, weaknesses, opportunities, and threats (SWOT) to develop a plan.
Open the template
Share template feedback

3.3 PROPOSED SOLUTION

S.NO	Parameter	Description
1.	Problem Statement (Problem to be solved)	Statement: The handwritten digit recognition is the capability of computer applications to recognize the human handwritten digits. Description: It is a hard task for the machine because handwritten digits are not perfect and can be made with many different shapes and sizes
2.	Idea / Solution description	1. It is the capability of a computer to fete the mortal handwritten integers from different sources like images, papers, and touch defenses. 2. It allows user to translate all those signature and notes into electronic words in a text document format and this data only requires far less physical copies.
3.	Novelty / Uniqueness	Accurately recognize the digits rather than recognizing all the characters like OCR.
4.	Social Impact / Customer Satisfaction	1. Artificial Intelligence developed the app called handwritten digit Recognizer. 2. It converts the written word into digital approximations and utilizes complex algorithms to identify characters before churning out a digital approximation
5.	Business Model (Revenue Model)	1. This system can be integrated with traffic surveillance cameras to recognize the vehicle's number plates for effective traffic management.

		2. Can be integrated with Postal system to identify and recognize the pin-code details easily.
6.	Scalability of the Solution	1. Ability to recognize digits in more noisy environments. 2. There is no limit in the number of digits it can be recognized.

3.4 PROBLEM SOLUTION FIT



CHAPTER 4

REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

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 categorize 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 and get the training and validation data first.
FR-4	Cloud: The cloud offers a range of IT services, including virtual storage, networking, servers, databases, and applications. In plain English, cloud computing is described as a virtual platform that enables unlimited storage and access to your data over the internet.
FR-5	Modified National Institute of Standards and Technology dataset: The abbreviation MNIST stands for the MNIST dataset. It is a collection of 60,000 tiny square grayscale photographs, each measuring 28 by 28, comprising handwritten single digits between 0 and 9.

4.2 NON_FUNCTIONAL REQUIREMENTS

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 recognize 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**	The features for handwritten digit recognition have been Acquainted. These features are based on shape analysis of the digit image and extract slant or slope information. They are effective in obtaining good recognition of accuracy.
NFR-6	**Scalability**	The scalability in the task of handwritten digit recognition, using a classifier, has great importance and it makes use of online handwriting recognition on computer tablets, recognizing zip codes on mail for postal mail sorting, processing bank check amounts, numeric entries in forms filled up manually(for example - tax forms) and so on.

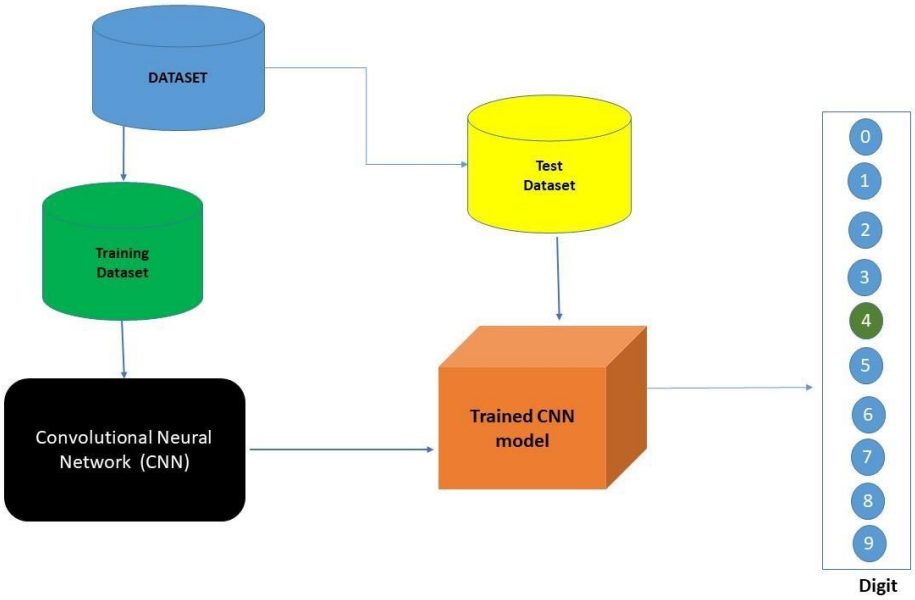
CHAPTER 5

PROJECT DESIGN

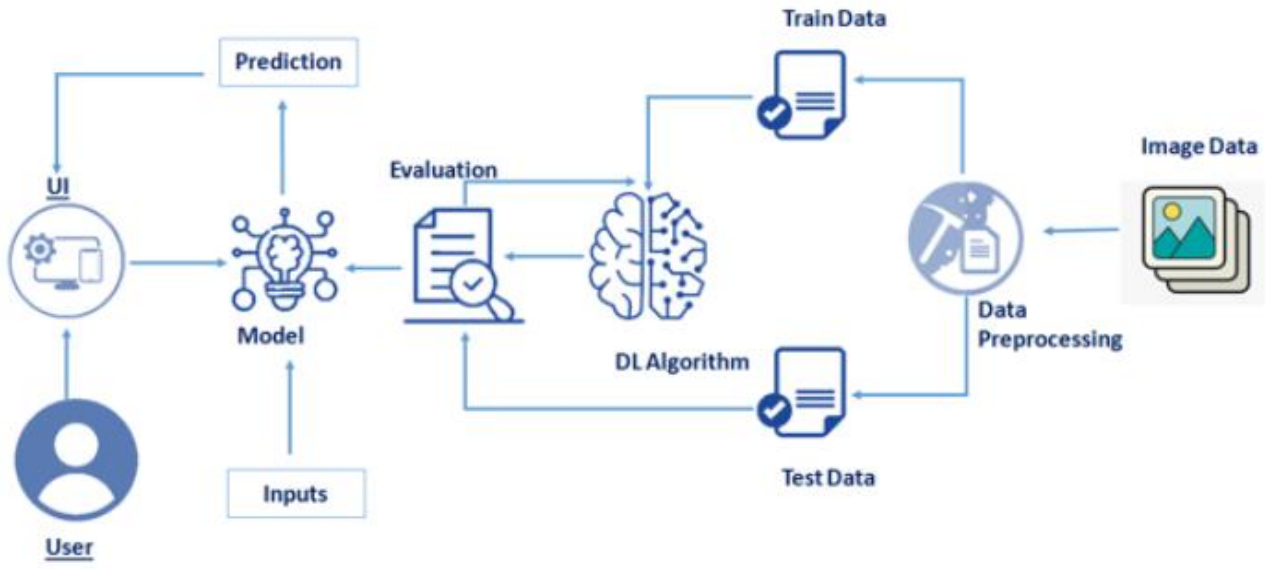
5.1 DATA FLOW DIAGRAM

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.

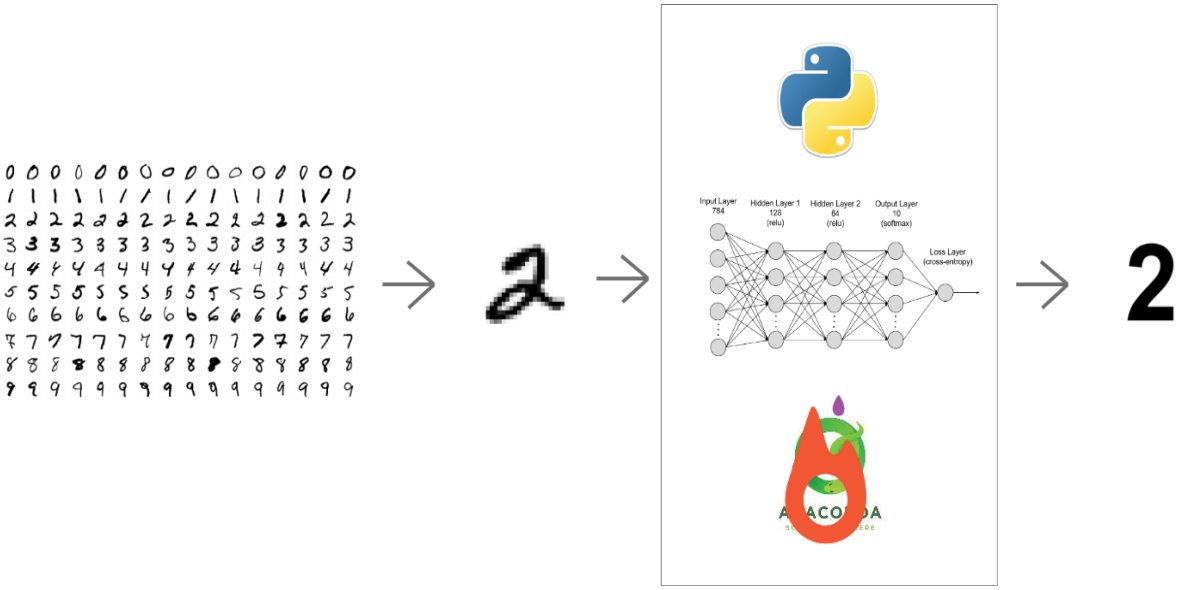
Data Flow Diagrams: A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear
DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where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



5.2 SOLUTION & TECHNICAL ARCHITECTURE



MNIST DATASET PROCESSING WITH PYTHON



5.3 COMPONENTS & TECHNOLOGIES:

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g., Mobile Application	HTML, CSS, JavaScript
2.	Application Logic-1	Logic for a process in the application	Java / 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 AI in cloud	IBM DB2
7.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or local file system
8.	External API-1	Purpose of External API used in the application	IBM Weather API, etc.
9.	Internet of Things Model	Purpose of AI Model is for integrating the sensors with a user interface	IBM AI Platform
10.	Machine Learning Model	Purpose of Machine Learning Model	Digit Recognition Model

5.4 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'm able to use the application in mobile.	I can view the application in mobile.	Low	Sprint-1
		USN-2	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-3	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-4	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-5	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-7	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-8	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-9	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
User Type	**Functional Requirement (Epic)**	**User Story Number**	**User Story / Task**	**Acceptance criteria**	**Priority**	**Release**
Customer (PC user)	Home	USN-10	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-11	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-12	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-1
	Recognize	USN-13	As a user, I can use the web application virtually anywhere.	I can use the application portably anywhere.	High	Sprint-2
		USN-14	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-15	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-3

	Predict	USN-16	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-4

CHAPTER 6

PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNING AND ESTIMATION

SPRINT	USER STORY / TASK	STORY POINTS	PRIORITY	TEAM MEMBERS
Sprint - I	Get the dataset	3	High	Subashkumar S
	Explore the data	2	Medium	Santhoshkumar R
	Data Pre-Processing	3	High	Arunkumar S
	Prepare training and testing data	3	High	Natesh T
Sprint - II	Create the model	3	High	Arunkumar S
	Train the model	3	High	Subashkumar S
	Test the model	3	High	Santhoshkumar R
Sprint - III	Improve the model	2	Medium	Natesh T
	Save the model	3	High	Natesh T Arunkumar S
	Build the Home Page	3	High	Subashkumar S
	Setup a database to store input images	2	Medium	Santhoshkumar R

Sprint - IV	Build the results page	3	High	Santhoshkumar R
	Integrate the model with the application	3	High	Subashkumar S
	Test the application	3	High	Natesh T, Arunkumar S

6.2 SPRINT DELIVERY SCHEDULE

SPRINT	TOTAL STORY POINTS	DURATION	SPRINT START DATE	SPRINT END DATE (PLANNED)	STORY POINTS COMPLETED (AS ON PLANNED DATE)	SPRINT RELEASE DATE (ACTUAL)
Sprint - I	11	6 Days	24 Oct 2022	29 Oct 2022	11	29 Oct 2022
Sprint - II	9	6 Days	31 Oct 2022	05 Nov 2022	9	05 Nov 2022
Sprint - III	10	6 Days	07 Oct 2022	12 Nov 2022	10	12 Nov 2022
Sprint - IV	9	6 Days	14 Nov 2022	19 Nov 2022	9	19 Nov 2022

6.3 REPORT FROM JIRA

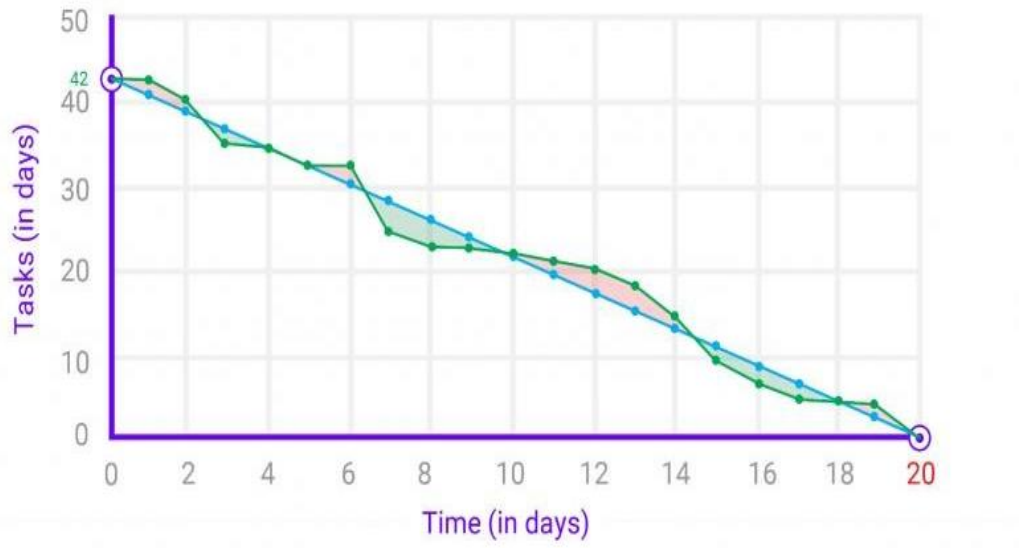
Velocity:

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

Average Velocity = 20 / 6 = 3.33

Burndown Chart:

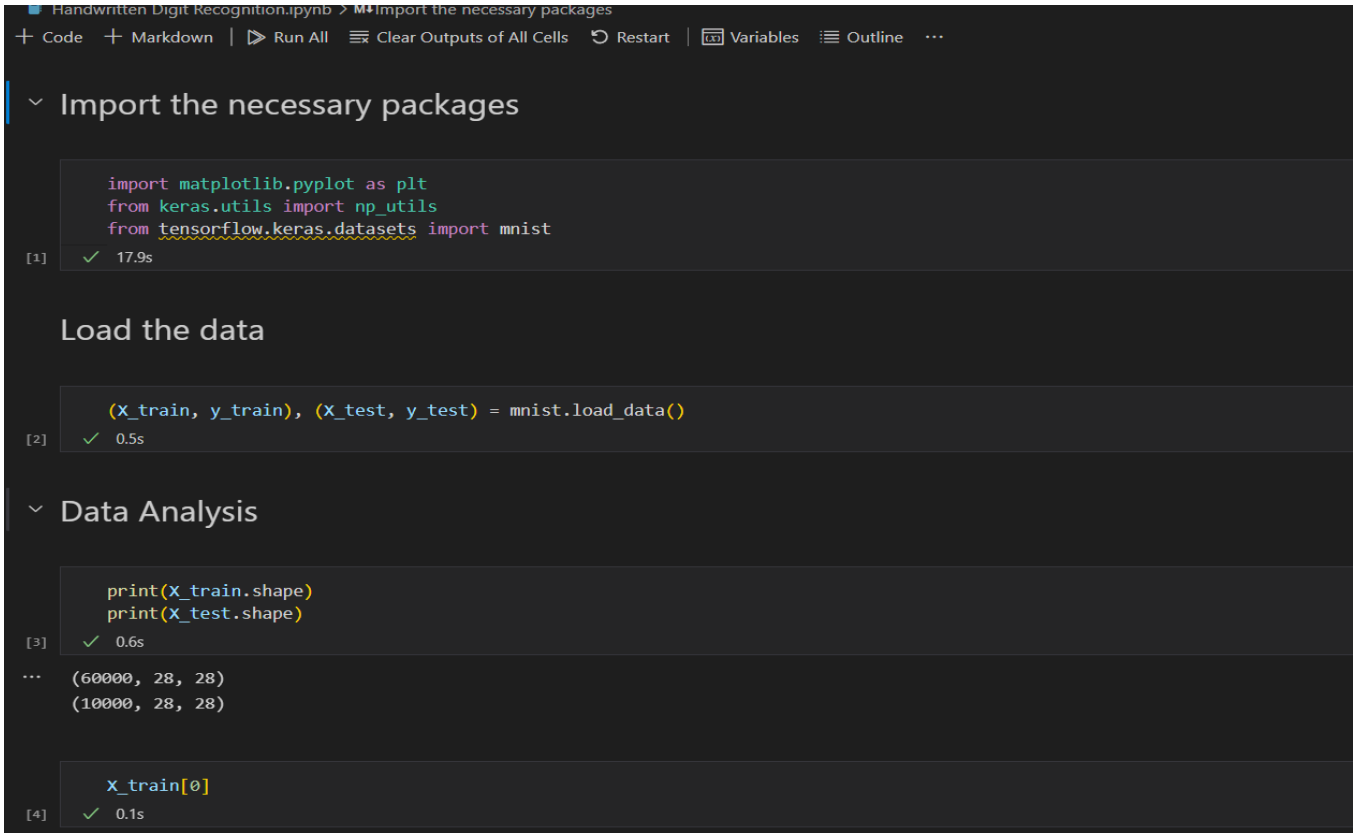
A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.



CHAPTER 7

CODING & SOLUTION

7.1 SPRINT 1



```
… Output exceeds the size limit. Open the full output data in a text editor
array([[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   3,
         18,  18,  18, 126, 136, 175,  26, 166, 255, 247, 127,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,  30,  36,  94, 154, 170,
        253, 253, 253, 253, 253, 225, 172, 253, 242, 195,  64,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,  49, 238, 253, 253, 253, 253,
        253, 253, 253, 253, 251,  93,  82,  82,  56,  39,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,  18, 219, 253, 253, 253, 253,
...
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
```

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')
```

[7] ✓ 0.1s

```
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)
```

[8] ✓ 0.4s

```
Y_train[0]
```

[9] ✓ 0.5s

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

7.2 SPRINT 2

Create 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"))
```

[10] ✓ 0.3s

compile: Any

compile: Any

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

[11] ✓ 0.6s

Train the model

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

[12] ✓ 9m 6.1s

```
Epoch 1/5
1875/1875 [==============================] - 114s 60ms/step - loss: 0.2773 - accuracy: 0.9506 - val_loss: 0.1015 - val_accuracy: 0.9698
Epoch 2/5
1875/1875 [==============================] - 107s 57ms/step - loss: 0.0697 - accuracy: 0.9787 - val_loss: 0.0694 - val_accuracy: 0.9797
Epoch 3/5
1875/1875 [==============================] - 108s 58ms/step - loss: 0.0497 - accuracy: 0.9843 - val_loss: 0.1044 - val_accuracy: 0.9763
Epoch 4/5
1875/1875 [==============================] - 110s 59ms/step - loss: 0.0381 - accuracy: 0.9877 - val_loss: 0.0844 - val_accuracy: 0.9791
Epoch 5/5
1875/1875 [==============================] - 107s 57ms/step - loss: 0.0297 - accuracy: 0.9903 - val_loss: 0.1028 - val_accuracy: 0.9768
```

Test the model

+ Code + Markdown

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

[13] ✓ 2.9s

```
Metrics (Test Loss & Test Accuracy):
[0.1028306856751442, 0.9768000245094299]
```

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

[14] ✓ 0.2s

```
1/1 [==============================] - 0s 138ms/step
[[1.0741620e-11 2.2714581e-17 5.8157077e-08 1.3988182e-07 3.9624286e-16
  1.9408322e-14 4.9455917e-20 9.9999988e-01 2.2219671e-08 2.5562844e-11]
 [2.9521760e-11 2.1299244e-11 1.0000000e+00 8.4716071e-12 7.1529199e-17
  1.3787961e-19 3.1256714e-10 1.1063413e-16 2.0238765e-12 5.0704795e-16]
 [4.1316980e-06 9.8946989e-01 6.2848478e-05 1.1830205e-09 9.7428793e-03
  9.2782386e-07 2.3566345e-08 3.2235835e-08 7.1932178e-04 5.4061683e-10]
 [9.9999893e-01 7.0810903e-13 1.0412999e-06 9.8047904e-13 1.5475894e-10
  2.5959249e-10 6.4390622e-09 1.6189073e-11 1.2669723e-08 1.2285047e-08]]
```

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

[15] ✓ 0.4s

```
[7 2 1 0]
```

7.3SPRINT 3

```
<html>

<head>
  <title>Digit Recognition Webpage</title>

  <meta name="viewport" content="width=device-width">
  <!-- G
  <link                                                    lay=swap" rel="stylesheet">
  <link href="https://fonts.googleapis.com/css2?family=Varela+Round&display=swap" rel="stylesheet">
  <link href="https://fonts.googleapis.com/css2?family=Source+Code+Pro:wght@500&display=swap" rel="stylesheet">
  <link href="https://fonts.googleapis.com/css?family=Calistoga|Josefin+Sans:400,700|Pacifico&display=swap" rel="stylesheet">
  <!-- bootstrap -->
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
  integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">
  <link rel="stylesheet" type= "text/css" href= "{{ url_for('static',filename='css/style.css') }}">
  <!-- fontawesome -->
  <script src="https://kit.fontawesome.com/b3aed9cb07.js" crossorigin="anonymous"></script>

  <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH
  +8abtTE1Pi6jizo" crossorigin="anonymous"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"
  integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1" crossorigin="anonymous"></script>
  <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"
  integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM" crossorigin="anonymous"></script>
  <script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@latest"></script>

</head>

<script>
  function preview() {
    frame.src=URL.createObjectURL(event.target.files[0]);
}

    $(document).ready(function() {
          $('#clear_button').on('click', function() {
              $('#image').val('');
              $('#frame').attr('src',"");
            });
```

This attribute specifies the URL of the linked resource. A URL can be absolute or relative.

```
</script>

<body>


  <section id="title">
    <h4 class="heading">Handwritten Digit Recognition Website</h4>
    <br><br>
      <p>
        The website is designed to predict the handwritten digit.
      </p>
    <p>
        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.</p>

      <br>
    <p> 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</p>

  </section>

  <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" src="" width="100px" height="100px"/>
          <div class="buttons_div">
            <button type="submit" class="btn btn-dark" id="predict_button">Predict</button>
            <button type="button" class="btn btn-dark" id="clear_button">  Clear  </button>
          </div>
        </form>
  </section>

  <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" src="" width="100px" height="100px"/>
          <div class="buttons_div">
            <button type="submit" class="btn btn-dark" id="predict_button">Predict</button>
            <button type="button" class="btn btn-dark" id="clear_button">  Clear  </button>
          </div>
        </form>
        </div>
  </section>

</body>

</html>
```

CHAPTER 8

TESTING

8.1 TEST CASES

Test caseID	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	FAIL
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 file	The input image should be uploaded to the application successfully	Working as expected	PASS
HP_TC_004	Functional	Home Page	Check if user cannot upload unsupported files	The application should not allow user to select a non image file	User is able to upload any file	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 theroutes 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	Check if the other predictions are displayed properly	The other predictions should be displayed properly	Working as expected	PASS

8.2 USER ACCEPTANCE TESTING

8.2.1 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

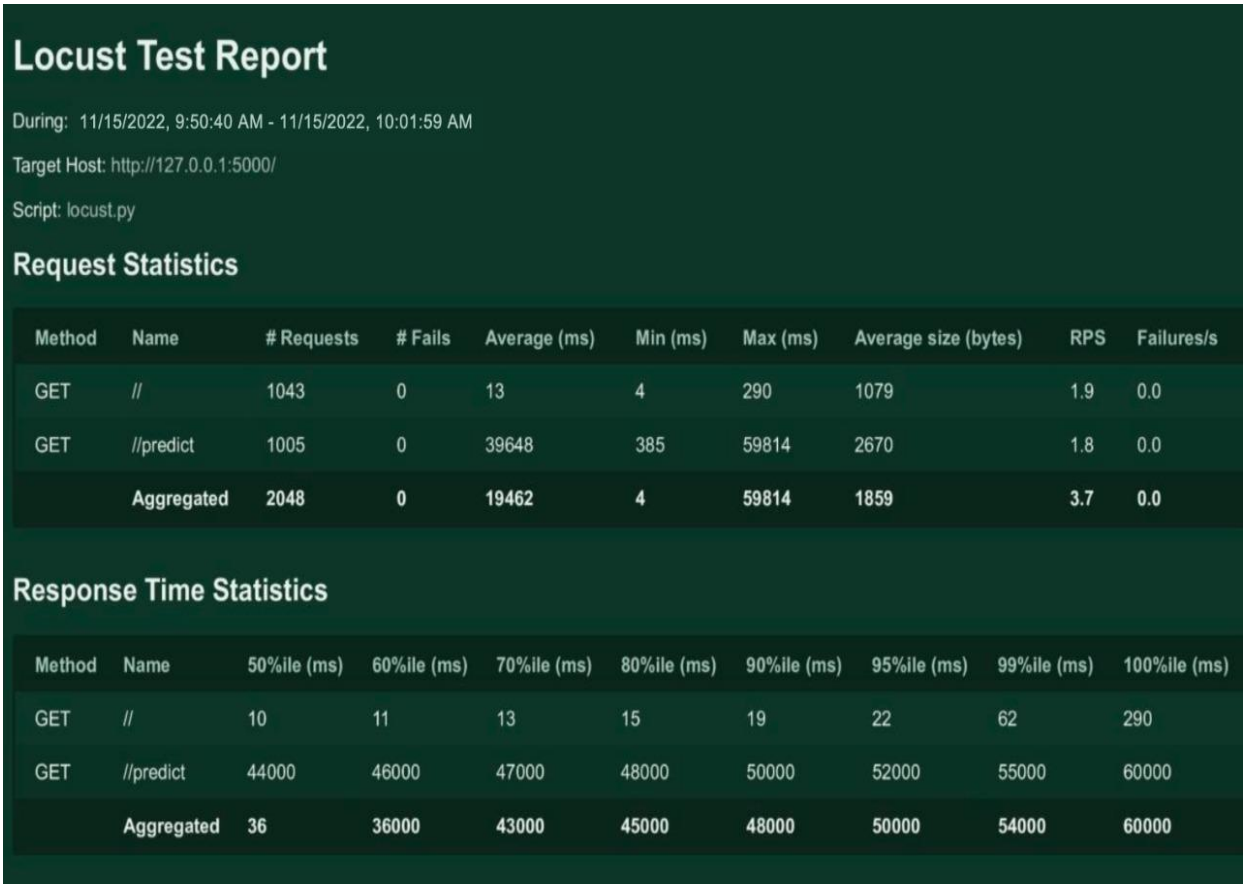
8.2.2 TEST CASE ANALYSIS

Section	Total Cases	Not Tested	Fail	Pass
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

9.1 PERFORMANCE METRICS



Locust Test Report

During: 11/15/2022, 9:50:40 AM - 11/15/2022, 10:01:59 AM

Target Host: http://127.0.0.1:5000/

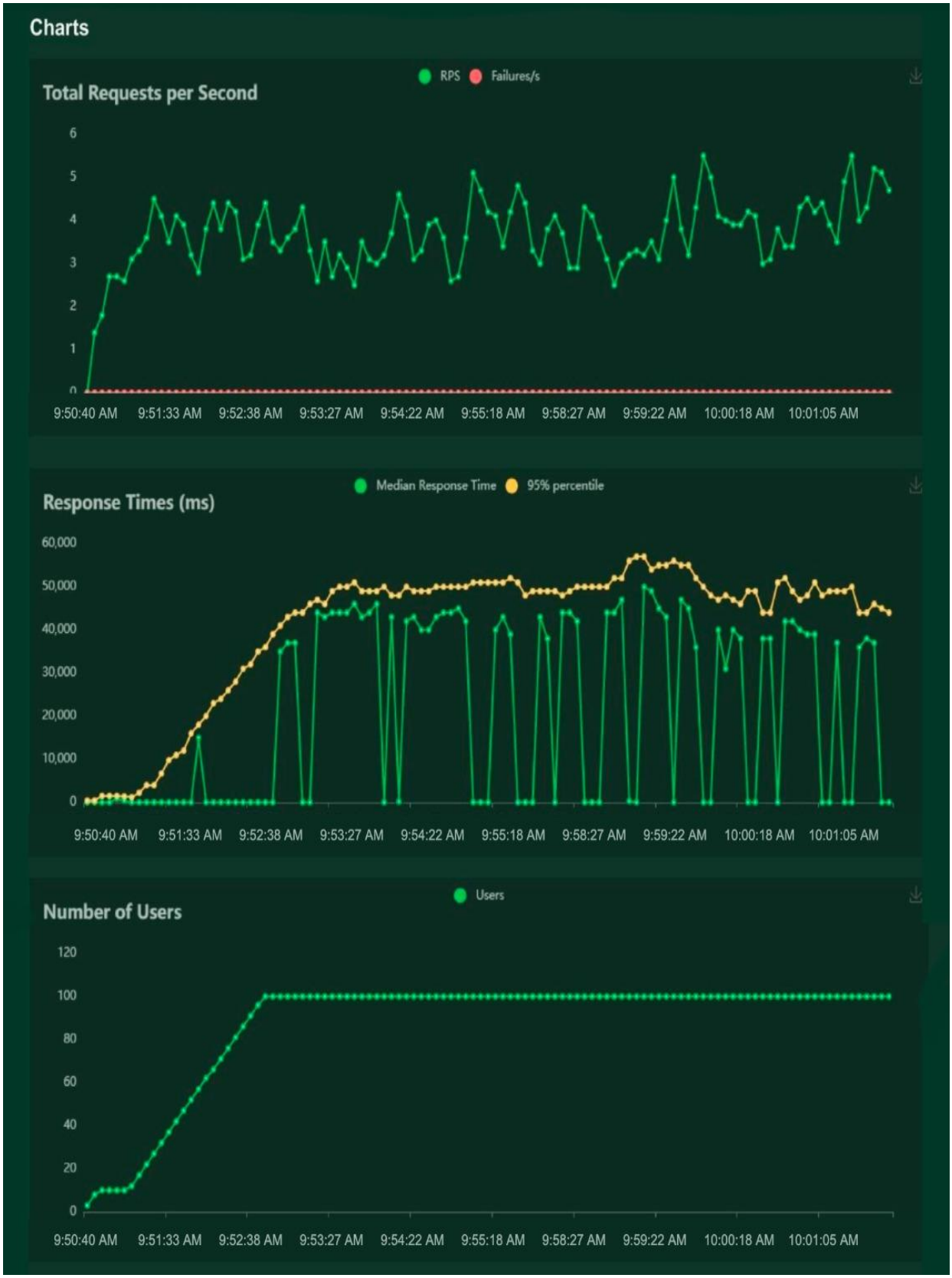
Script: locust.py

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**


Charts
Total Requests per Second
RPS
Failures/s
6
5
4
3
2
1
0
9:50:40 AM
9:51:33 AM
9:52:38 AM
9:53:27 AM
9:54:22 AM
9:55:18 AM
9:58:27 AM
9:59:22 AM
10:00:18 AM
10:01:05 AM
Response Times (ms)
Median Response Time
95% percentile
60,000
50,000
40,000
30,000
20,000
10,000
0
9:50:40 AM
9:51:33 AM
9:52:38 AM
9:53:27 AM
9:54:22 AM
9:55:18 AM
9:58:27 AM
9:59:22 AM
10:00:18 AM
10:01:05 AM
Number of Users
Users
120
100
80
60
40
20
0
9:50:40 AM
9:51:33 AM
9:52:38 AM
9:53:27 AM
9:54:22 AM
9:55:18 AM
9:58:27 AM
9:59:22 AM
10:00:18 AM
10:01:05 AM

CHAPTER 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
- Neural Network is used to train and identify written digits for greater efficiency.
- The accuracy rate is very high.
- Speed of data entry
- It is much easier to dictate the machine than to write
- Easier data retrieval

DISADVANTAGES

- Cannot handle complex data
- All the data must be in digital format
- Requires a high performance server for faster predictions
- Prone to occasional errors
- There is a wide range of handwriting – good and bad.
- It is tricky for programmers to provide enough examples of how every character might look.
- Customers must try with clear image and neat handwriting to get accuracy in digits.
- Unclear image will not give accurate results.

CHAPTER 11
CONCLUSION

Convolutional Neural Network (CNN) adds its significant improvement to the Manuscript Document Recognition System. This paper tells us the effectiveness of CNN-based classification of data and pre-processing methods. Our model clearly sees handwriting and achieves outgoing predictions of up to 82.16% and accurate predictions of up to 69.16%. However the model can be continuously developed using multiple training samples. This will help the model to learn as well as the generalize better. There are many images in the training set that are completely invisible to the human eye.

This project demonstrated a web application that uses machine learning to recognize 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 numberplates of vehicles, processing bank cheque amounts, numeric entries in forms filledup by hand (tax forms) and so on.

Through extensive evaluation using a MNIST dataset, the present work suggests the role of various hyper-parameters. Fine tuning of hyper-parameters is essential in improving the performance of CNN architecture. We achieved a recognition rate of 99.89% with the Adam optimizer for the MNIST database, which is better than all previously reported results. The effect of increasing the number of convolutional layers in CNN architecture on the performance of handwritten digit recognition is clearly presented through the experiments.

CHAPTER 12

FUTURE SCOPE

This project can be enhanced with a great field of machine learning and artificial intelligence. The world can think of a software which can recognize the text from a picture and can show it to the others, for example a shop name detector. Or this project can be extended to a greater concept of all the character sets in the world. This project has not gone for the total English alphabet because there will be more and many more training sets and testing values that the neural network model will not be enough to detect. Think of a AI modeled car sensor going with a direction modeling in the roadside, user shall give only the destination.

All of these enhancement is an application of the texture analysis where advanced image processing, Neural network model for training and advanced AI concepts will come. These applications can be modeled further .As this project is fully done by free and available resources and packages this can be also a limitation of the project. The fund is very important because all machine learning libraries and advanced packages are not available for free. Unless of those the most of the visualizing platforms like on which developers are doing some works like Watson Studio or Aws. These all are mainly paid platforms where a lot of ML projects are going on.

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:

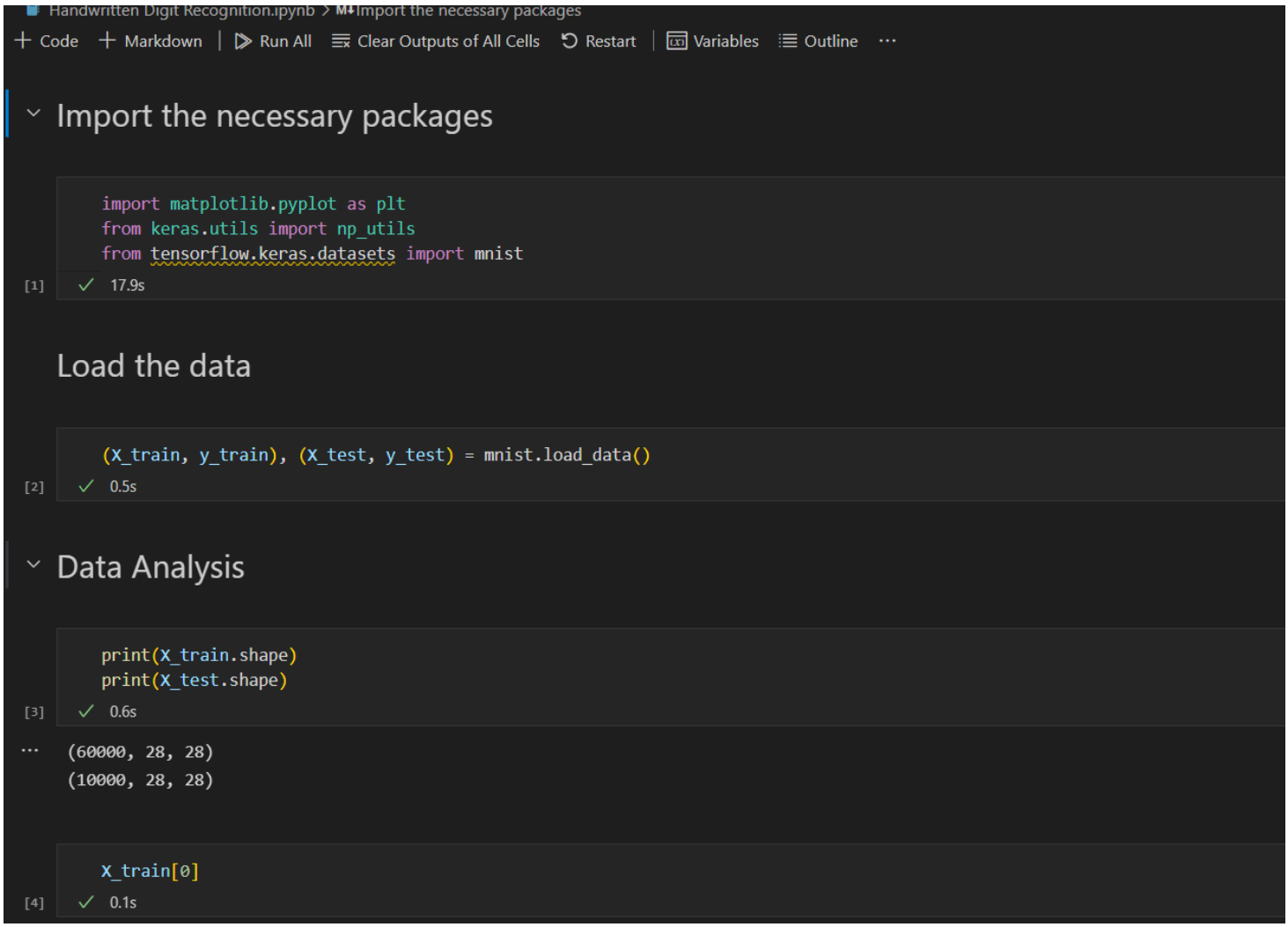
- Add support to detect from digits multiple images and save the results
- Add support to detect multiple digits
- Improve model to detect digits from complex images
- Add support to different languages to help users from all over the world

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.

APPENDIX

SOURCE CODE

MODEL CREATION:



```
Output exceeds the size limit. Open the full output data in a text editor
array([[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   3,
         18,  18,  18, 126, 136, 175,  26, 166, 255, 247, 127,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,  30,  36,  94, 154, 170,
        253, 253, 253, 253, 253, 225, 172, 253, 242, 195,  64,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,  49, 238, 253, 253, 253, 253,
        253, 253, 253, 253, 251,  93,  82,  82,  56,  39,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,  18, 219, 253, 253, 253, 253,
...
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
```

```
y_train[0]
```

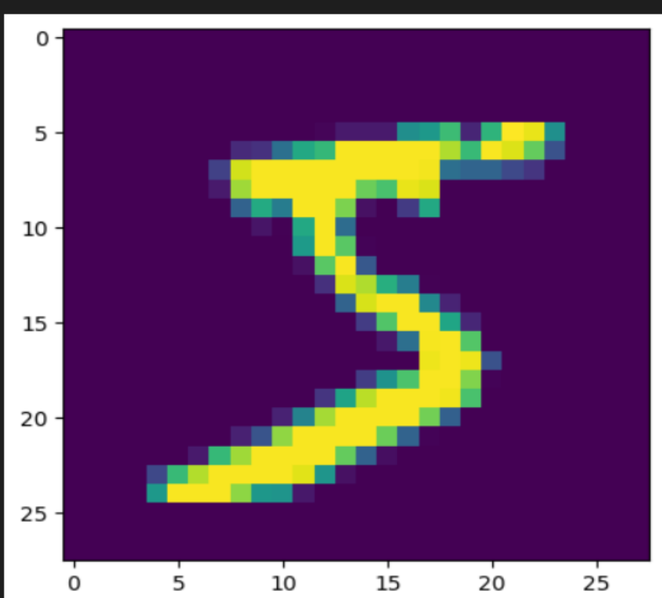
[5] ✓ 0.8s

```
5
```

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

[6] ✓ 0.4s

```
<matplotlib.image.AxesImage at 0x16bfeab66e0>
```



Create 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"))
```

[10] ✓ 0.3s

compile: Any

compile: Any

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

[11] ✓ 0.6s

Train the model

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

[12] ✓ 9m 6.1s

```
Epoch 1/5
1875/1875 [==============================] - 114s 60ms/step - loss: 0.2773 - accuracy: 0.9506 - val_loss: 0.1015 - val_accuracy: 0.9698
Epoch 2/5
1875/1875 [==============================] - 107s 57ms/step - loss: 0.0697 - accuracy: 0.9787 - val_loss: 0.0694 - val_accuracy: 0.9797
Epoch 3/5
1875/1875 [==============================] - 108s 58ms/step - loss: 0.0497 - accuracy: 0.9843 - val_loss: 0.1044 - val_accuracy: 0.9763
Epoch 4/5
1875/1875 [==============================] - 110s 59ms/step - loss: 0.0381 - accuracy: 0.9877 - val_loss: 0.0844 - val_accuracy: 0.9791
Epoch 5/5
1875/1875 [==============================] - 107s 57ms/step - loss: 0.0297 - accuracy: 0.9903 - val_loss: 0.1028 - val_accuracy: 0.9768
```

Test the model

+ Code + Markdown

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

[13] ✓ 2.9s

```
Metrics (Test Loss & Test Accuracy):
[0.1028306856751442, 0.9768000245094299]
```

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

[14] ✓ 0.2s

```
1/1 [==============================] - 0s 138ms/step
[[1.0741620e-11 2.2714581e-17 5.8157077e-08 1.3988182e-07 3.9624286e-16
  1.9408322e-14 4.9455917e-20 9.9999988e-01 2.2219671e-08 2.5562844e-11]
 [2.9521760e-11 2.1299244e-11 1.0000000e+00 8.4716071e-12 7.1529199e-17
  1.3787961e-19 3.1256714e-10 1.1063413e-16 2.0238765e-12 5.0704795e-16]
 [4.1316980e-06 9.8946989e-01 6.2848478e-05 1.1830205e-09 9.7428793e-03
  9.2782386e-07 2.3566345e-08 3.2235835e-08 7.1932178e-04 5.4061683e-10]
 [9.9999893e-01 7.0810903e-13 1.0412999e-06 9.8047904e-13 1.5475894e-10
  2.5959249e-10 6.4390622e-09 1.6189073e-11 1.2669723e-08 1.2285047e-08]]
```

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

[15] ✓ 0.4s

```
[7 2 1 0]
```

TRAIN THE MODEL ON IBM:

Cloud deploy

```
!pip install -U ibm-watson-machine-learning
```

[23]

```
Requirement already satisfied: ibm-watson-machine-learning in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (1.0.257)
Requirement already satisfied: certifi in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (2022.9.24)
Requirement already satisfied: tabulate in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (0.8.9)
Requirement already satisfied: urllib3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (1.26.7)
Requirement already satisfied: ibm-cos-sdk==2.11.* in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning)
(2.11.0)
Requirement already satisfied: pandas<1.5.0,>=0.24.2 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning)
(1.3.4)
Requirement already satisfied: importlib-metadata in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (4.8.2
Requirement already satisfied: lomond in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (0.3.3)
Requirement already satisfied: requests in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (2.26.0)
Requirement already satisfied: packaging in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (21.3)
Requirement already satisfied: ibm-cos-sdk-s3transfer==2.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk==2.11.*-
>ibm-watson-machine-learning) (2.11.0)
Requirement already satisfied: ibm-cos-sdk-core==2.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk==2.11.*->ibm-
watson-machine-learning) (2.11.0)
Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk==2.11.*->ibm-watson
machine-learning) (0.10.0)
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk-core==2.11.0-
>ibm-cos-sdk==2.11.*->ibm-watson-machine-learning) (2.8.2)
Requirement already satisfied: pytz>=2017.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas<1.5.0,>=0.24.2->ibm-watson-machine
learning) (2021.3)
Requirement already satisfied: numpy>=1.17.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas<1.5.0,>=0.24.2->ibm-watson-
```

```
from ibm_watson_machine_learning import APIClient
credentials ={
    "url":"https://us-south.ml.cloud.ibm.com",
    "apikey": "3rOdoqX0g-y3SFPJyngfww-p9LcUhoyNBIiY34tCAYsm"
}
client = APIClient(credentials)
```

[52]

```
client.spaces.get_details()
```

[53]

```
Output exceeds the size limit. Open the full output data in a text editor
{'resources': [{'entity': {'compute': [{'crn': 'crn:v1:bluemix:public:pm-20:us-south:a/47abf318d4794b229c16e8f0270a723f:e3130ac1-3cdb-414a-b0
7e40fa9a643c::',
      'guid': 'e3130ac1-3cdb-414a-b06e-7e40fa9a643c',
      'name': 'Watson Machine Learning-fq',
      'type': 'machine_learning'}],
    'description': '',
    'name': 'Handwritten recognition',
    'scope': {'bss_account_id': '47abf318d4794b229c16e8f0270a723f'},
    'stage': {'production': False},
    'status': {'state': 'active'},
    'storage': {'properties': {'bucket_name': 'b60679a5-6e93-4e92-b0da-972ac47a5a78',
      'bucket_region': 'us-south',
      'credentials': {'admin': {'access_key_id': 'f01c8a8c6efa4e48b2480641d3ba34af',
        'api_key': 'RGGtm1Sp0uA6Hh85ewdi39qy0CZIuoSZrJp9gjc8MWLE',
        'secret_access_key': '085c88b71ae3609ffefff76804e33bdd38023b69513917be',
        'service_id': 'ServiceId-99dcddd1-5770-4e6a-9ab8-827785e02cd1'},
       'editor': {'access_key_id': '20470cfb96324028a4329f6b75a8fb72',
        'api_key': 'ONncxeikVftmgIQLNXxGQorMI5avuU_2h0IIbmBxuNCj',
        'resource_key_crn': 'crn:v1:bluemix:public:cloud-object-storage:global:a/47abf318d4794b229c16e8f0270a723f:bad41abf-8dcb-4699-bed3-
```

```
[55]
wml_clients=APIClient(wml_cred)
wml_clients.spaces.list()
```

```
Note: 'limit' is not provided. Only first 50 records will be displayed if the number of records exceed 50
------------------------------------  ----------------------  ------------------------
ID                                    NAME                    CREATED
f6ca7717-a4b6-4bcc-81c9-b83c200a3135  Handwritten recognition  2022-11-19T10:16:16.154Z
------------------------------------  ----------------------  ------------------------
```

```
[56]
space_id="f6ca7717-a4b6-4bcc-81c9-b83c200a3135"
```

```
[57]
client.set.default_space(space_id)
```

```
'SUCCESS'
```

```
[58]
client.software_specifications.list(limit=100)
```

```
Output exceeds the size limit. Open the full output data in a text editor
------------------------------  ------------------------------------  ----
NAME                            ASSET_ID                              TYPE
default_py3.6                   0062b8c9-8b7d-44a0-a9b9-46c416adcbd9  base
kernel-spark3.2-scala2.12       020d69ce-7ac1-5e68-ac1a-31189867356a  base
pytorch-onnx_1.3-py3.7-edt      069ea134-3346-5748-b513-49120e15d288  base
scikit-learn_0.20-py3.6         09c5a1d0-9c1e-4473-a344-eb7b665ff687  base
spark-mllib_3.0-scala_2.12      09f4cff0-90a7-5899-b9ed-1ef348aebdee  base
```

```
[60]
software_space_uid = client.software_specifications.get_uid_by_name('tensorflow_rt22.1-py3.9')
software_space_uid
```

```
'acd9c798-6974-5d2f-a657-ce06e986df4d'
```

```
[70]
cd models
```

```
[Errno 2] No such file or directory: 'models'
/home/wsuser/work/models
```

```
[72]
ls
```

```
mnistCNN.h5
```

```
[79]
model_details = client.repository.store_model(model='hdr_deployment.tgz',meta_props={
    client.repository.ModelMetaNames.NAME:"Digit Recognition System",
    client.repository.ModelMetaNames.TYPE:"tensorflow_2.7",
    client.repository.ModelMetaNames.SOFTWARE_SPEC_UID:software_space_uid
})
```

```
[80]
model_details
```

```
{'entity': {'hybrid pipeline software specs': [],
```

```
[80]
model_details
```

```
{'entity': {'hybrid_pipeline_software_specs': [],
  'software_spec': {'id': 'acd9c798-6974-5d2f-a657-ce06e986df4d',
   'name': 'tensorflow_rt22.1-py3.9'},
  'type': 'tensorflow_2.7'},
 'metadata': {'created_at': '2022-11-19T11:33:53.501Z',
  'id': '0feb60bc-30a8-40c0-a08c-12ef666da690',
  'modified_at': '2022-11-19T11:33:55.709Z',
  'name': 'Digit Recognition System',
  'owner': 'IBMid-666002MT1F',
  'resource_key': '3b2e7ffe-30c3-4f84-a39a-587f6268e912',
  'space_id': 'f6ca7717-a4b6-4bcc-81c9-b83c200a3135'},
 'system': {'warnings': []}}
```

```
[81]
model_id = client.repository.get_model_id(model_details)
model_id
```

```
'0feb60bc-30a8-40c0-a08c-12ef666da690'
```

```
[83]
client.repository.download(model_id,'DigitRecoginizer_IBM_model.tar.gz')
```

```
Successfully saved model content to file: 'DigitRecoginizer_IBM_model.tar.gz'

'/home/wsuser/work/models/DigitRecoginizer_IBM_model.tar.gz'
```

HOME PAGE(HTML) – index.html

```
<html>

<head>
  <title>Digit Recognition Webpage</title>

  <meta name="viewport" content="width=device-width">
  <!-- G
  <link                                                    lay=swap" rel="stylesheet">
  <link href="https://fonts.googleapis.com/css2?family=Varela+Round&display=swap" rel="stylesheet">
  <link href="https://fonts.googleapis.com/css2?family=Source+Code+Pro:wght@500&display=swap" rel="stylesheet">
  <link href="https://fonts.googleapis.com/css?family=Calistoga|Josefin+Sans:400,700|Pacifico&display=swap" rel="stylesheet">
  <!-- bootstrap -->
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
  integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">
  <link rel="stylesheet" type= "text/css" href= "{{ url_for('static',filename='css/style.css') }}">
  <!-- fontawesome -->
  <script src="https://kit.fontawesome.com/b3aed9cb07.js" crossorigin="anonymous"></script>

  <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH
  +8abtTE1Pi6jizo" crossorigin="anonymous"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"
  integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1" crossorigin="anonymous"></script>
  <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"
  integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM" crossorigin="anonymous"></script>
  <script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@latest"></script>

</head>

<script>
  function preview() {
    frame.src=URL.createObjectURL(event.target.files[0]);
}

    $(document).ready(function() {
          $('#clear_button').on('click', function() {
              $('#image').val('');
              $('#frame').attr('src',"");
            });
```

This attribute specifies the URL of the linked resource. A URL can be absolute or relative.

```
</script>

<body>


  <section id="title">
    <h4 class="heading">Handwritten Digit Recognition Website</h4>
    <br><br>
      <p>
        The website is designed to predict the handwritten digit.
      </p>
    <p>
        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.</p>

      <br>
    <p> 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</p>

  </section>

  <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" src="" width="100px" height="100px"/>
          <div class="buttons_div">
            <button type="submit" class="btn btn-dark" id="predict_button">Predict</button>
            <button type="button" class="btn btn-dark" id="clear_button">  Clear  </button>
          </div>
        </form>
```

```
  </section>

  <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" src="" width="100px" height="100px"/>
          <div class="buttons_div">
            <button type="submit" class="btn btn-dark" id="predict_button">Predict</button>
            <button type="button" class="btn btn-dark" id="clear_button">  Clear  </button>
          </div>
        </form>
        </div>
  </section>

</body>

</html>
```

HOME PAGE(CSS) – style.css

```
#clear_button{
  margin-left: 15px;
  font-weight: bold;
  color: blue;
}

#confidence{
  font-family: 'Josefin Sans', sans-serif;
  margin-top: 7.5%;
}

#content{
  margin: 0 auto;
  padding: 2% 15%;
  padding-bottom: 0;
}

.welcome{
   text-align: center;
   position: relative;
   color: honeydew;
   background-color: greenyellow;
   padding-top: 1%;
   padding-bottom: 1%;
   font-weight: bold;
   font-family: 'Prompt', sans-serif;
}

#team_id{
   text-align: right;
   font-size: 25px;
   padding-right: 3%;
}

#predict_button{
  margin-right: 15px;
  color: blue;
  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: 'Varela Round', sans-serif;
  font-weight: 700;
  font-size: 2rem;
  display: inline;
}

.leftside{
  text-align: center;
  margin: 0 auto;
  margin-top: 2%;
  /* padding-left: 10%; */
}

#frame{
  margin-right: 10%;
}

.predicted_answer{
  text-align: center;
  margin: 0 auto;
  padding: 3% 5%;
  padding-top: 0;
  /* padding-left: 10%; */
}

p{
  font-family: 'Source Code Pro', monospace, sans-serif;
  margin-top: 1%;
}

@media (min-width: 720px) {
  .leftside{
    padding-left: 10%;
  }
}
```

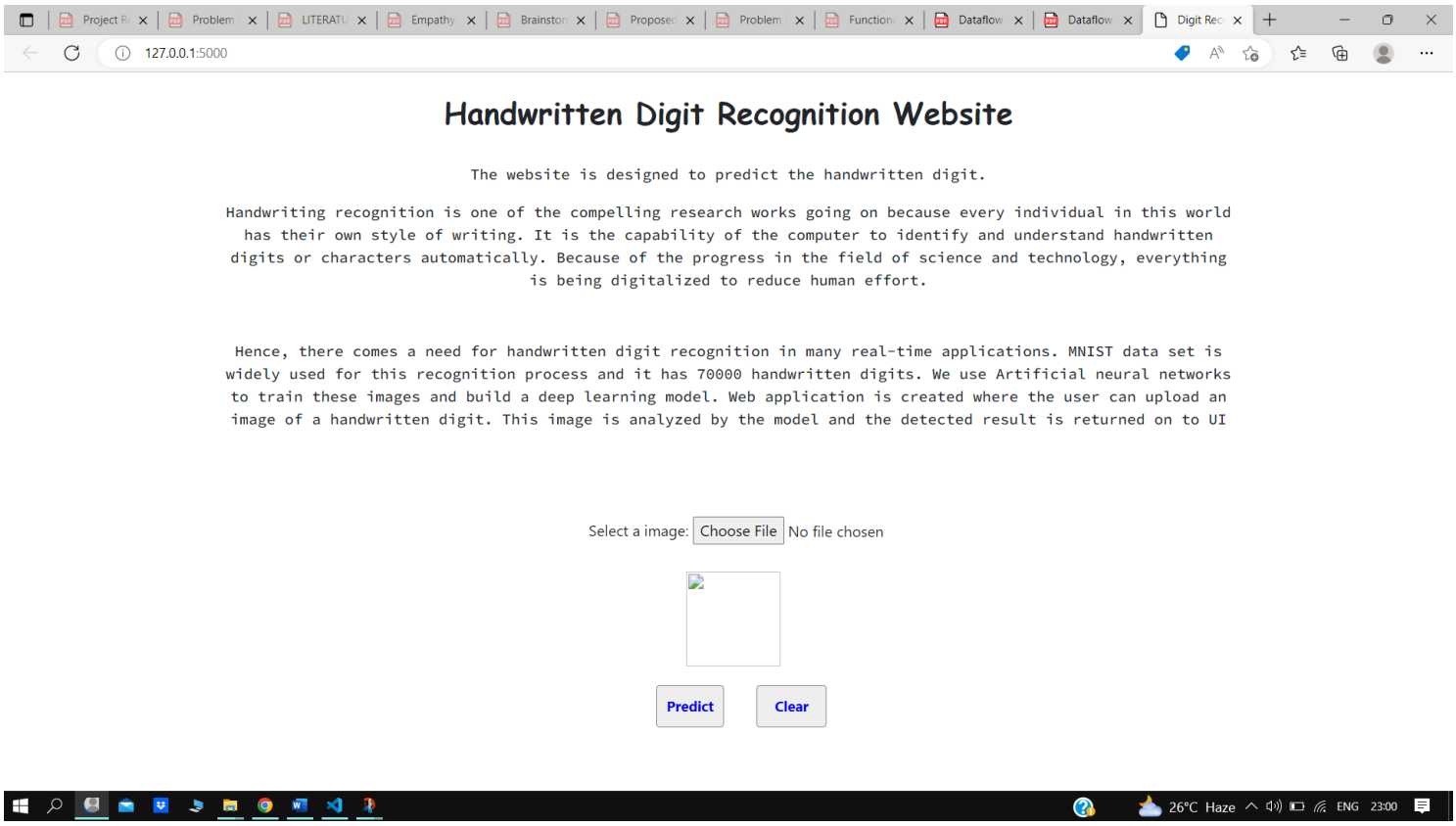
PREDICT PAGE (HTML) – predict.html

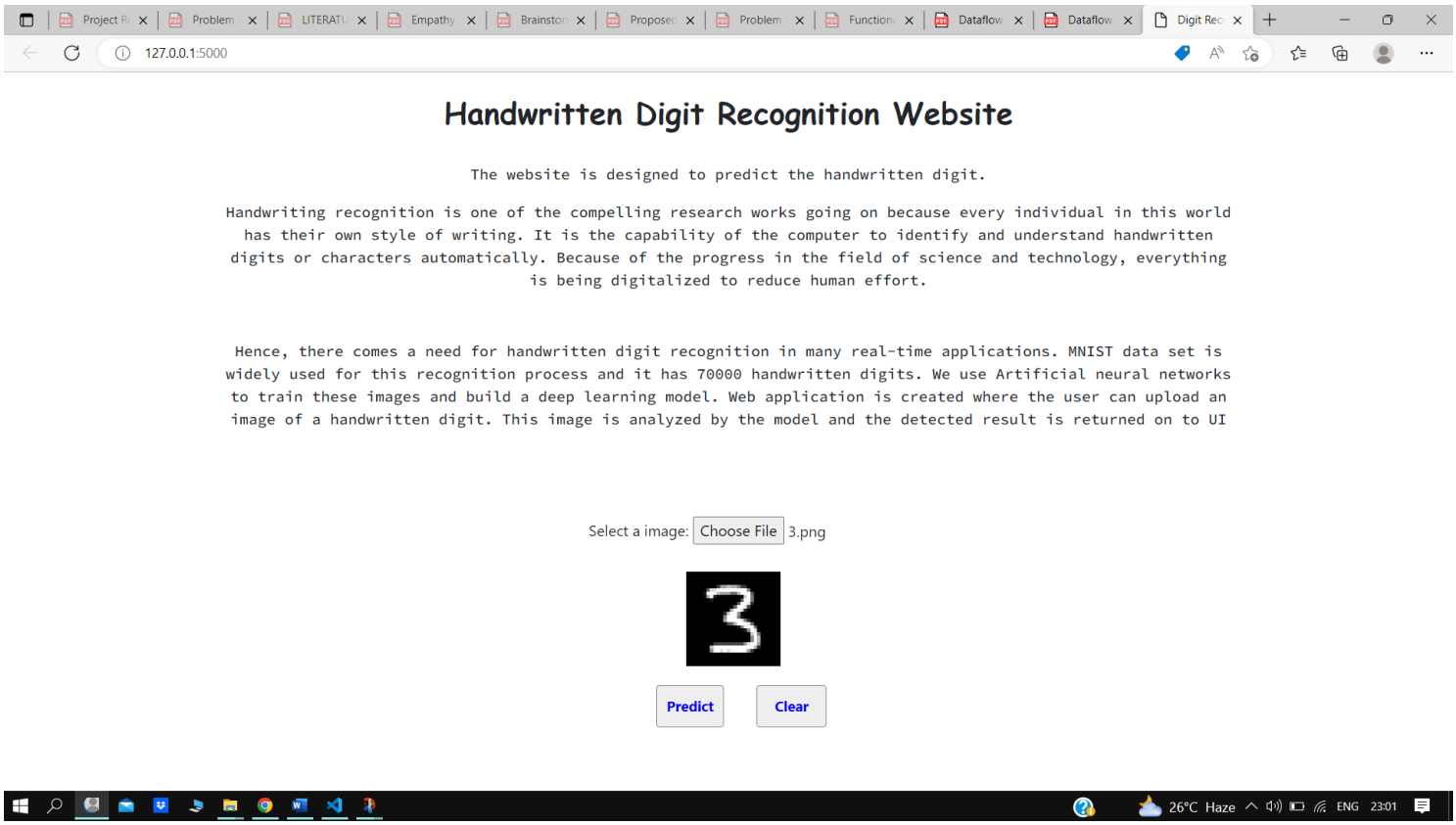
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Prediction</title>
</head>

<style>
    body{
     background-image: url('static/images/index6.jpg');
     background-repeat: no-repeat;
     background-size: cover;
    }

    #rectangle{
     width:400px;
     height:150px;
     background-color: #5796a5;
     border-radius: 25px;
     position: absolute;
     top:25%;
     left:50%;
     transform: translate(-50%,-50%);
    }

    #ans{
     text-align: center;
     font-size: 40px;
     margin: 0 auto;
     padding: 3% 5%;
     padding-top: 15%;
     color: white;
    }

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

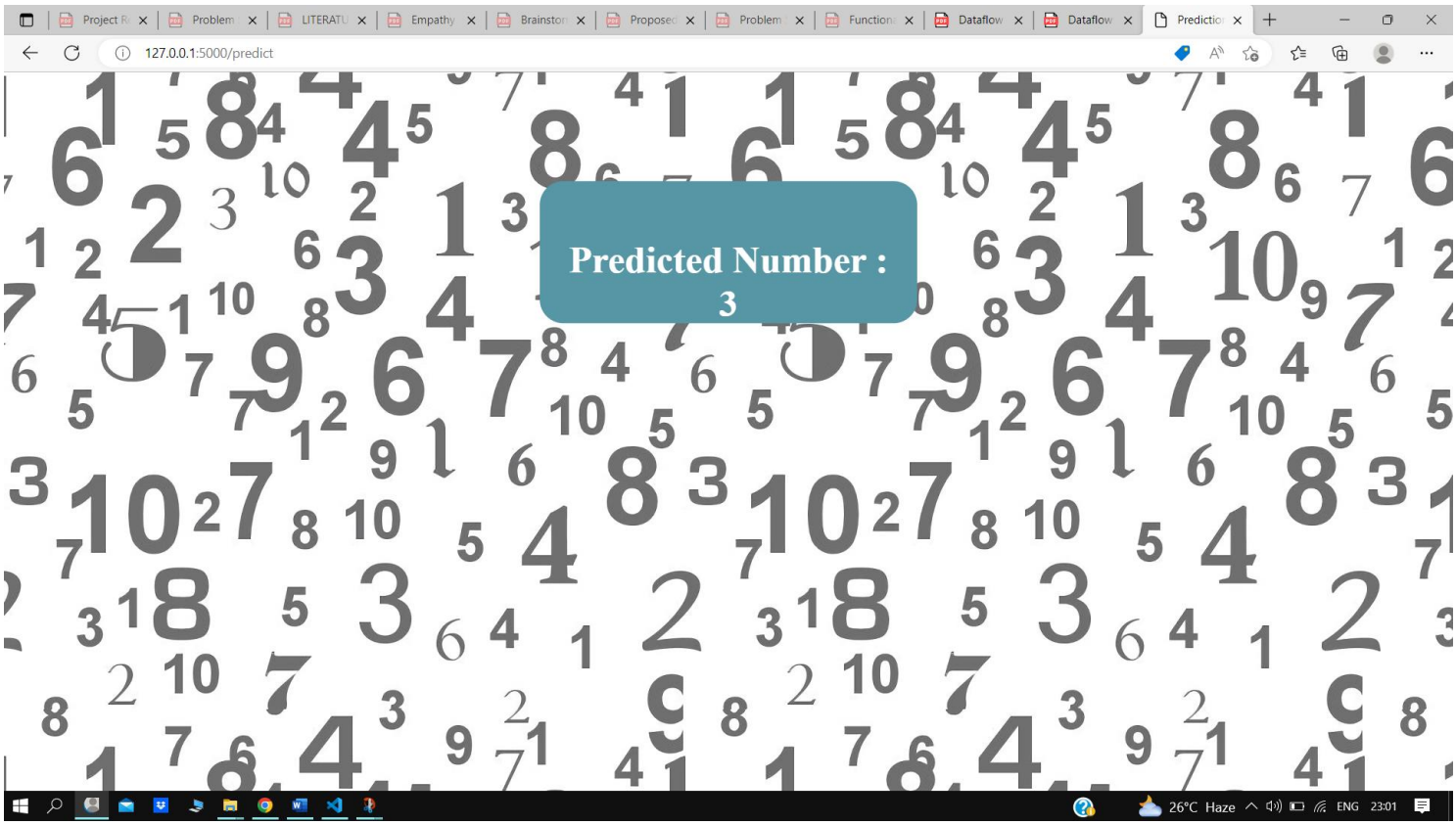
FLASK APP - app.py

```
import numpy as np
import os
from PIL import Image
from flask import Flask, request, render_template
from werkzeug.utils import secure_filename
from keras.models import load_model

UPLOAD_FOLDER = 'C:/Users/Dell/PycharmProjects/A-novel-method-for-digit-recognition-
system/flask_app/uploads'


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

model = load_model("mnistCNN.h5")


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


@app.route('/predict', methods=['GET', 'POST'])
def upload():
    if request.method == "POST":
        f = request.files["image"]
        filepath = secure_filename(f.filename)
        f.save(os.path.join(app.config['UPLOAD_FOLDER'], filepath))

        upload_img = os.path.join(UPLOAD_FOLDER, filepath)
        img = Image.open(upload_img).convert("L")  # convert image to monochrome
        img = img.resize((28, 28))  # resizing of input image

        im2arr = np.array(img)  # converting to image
        im2arr = im2arr.reshape(1, 28, 28, 1)  # reshaping according to our requirement

        pred = model.predict(im2arr)

        num = np.argmax(pred, axis=1)  # printing our Labels

        return render_template('predict.html', num=str(num[0]))


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

SCREENSHOTS:




127.0.0.1:5000
Handwritten Digit Recognition Website
The website is designed to predict the handwritten digit.
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
Select a image:
Choose File
3.png
Predict
Clear
26°C Haze
ENG
23:01



GITHUB LINK:

[Click to git repository]