

19CSP14 - PROFESSIONAL READINESS FOR INNOVATION, EMPLOYABILITY AND ENTREPRENEURSHIP

Novel Method for Handwritten Digit Recognition System

Team ID : PNT2022TMID22106

Team Leader : Deepaabishek P K

Team member 1: Ashwin Krishna M

Team member 2: Madhavan V

Team member 3: Jegadish M

1 . INTRODUCTION

Project Overview

Machine learning and deep learning play an important role in computer technology and artificial intelligence. Deep learning and machine learning can be used to reduce human effort in perception, learning, prediction, and many other areas.

Handwritten digit recognition is the ability of computer systems to recognize handwritten digits from various sources such as images, documents, and so on. This project aims to help users reduce the manual work of digit recognition using machine learning.

Purpose

Digit recognition systems can recognize digits from a variety of sources such as emails, bank checks, documents, and photographs. Recognize digits in a variety of real-world scenarios, such as online handwriting recognition on computer tablets and systems, vehicle license plate recognition, and bank check processing. For example, filling out handwritten forms (tax forms).

2. LITERATURE SURVEY

1) A novel method for Handwritten Digit Recognition with Neural Networks (MALOTHU NAGU*1, N VIJAY SHANKAR#2, K. ANNAPURNA)

- ♦ Character recognition plays an important role in the modern world. It can solve more complex problems and makes humans' job easier. An example is handwritten character recognition.
- ♦ Two techniques researched in this paper are Pattern Recognition and Artificial Neural Network (ANN). Both techniques are defined and different methods for each technique is also discussed.
- ♦ Neural Network is used to train and identify written digits. After training and testing, the accuracy rate reached 99%. This accuracy rate is very high.

2) An effective result-feedback neural algorithm for handwritten character recognition (Hao Y., Shi Y., Zhang D., Zhu X. 2001)

- ♦ It is designed as an effective neural network by adding confidence back-propagation and input modification, thus both pre-processing and recognition operations are closely integrated [10].
- ♦ The convergence of the algorithm is proved and many experiments show that the error rate in such a result-feedback neural network (RFNN) can be greatly reduced as well as the robust to environmental noise.

3) Handwritten numerical recognition based on multiple algorithms (Kimura, F. and Shiridhar, M. (1991))

- ♦ In this paper, the authors developed two algorithms for application to recognition of unconstrained isolated handwritten numerals.
- ♦ While both algorithms yielded very low error rates, the authors combined the two algorithms in different ways to study the best polling strategy and realized significant improvement in performance.

4) Recognition of isolated and simply handwritten numerals (M. Shridhar and A. Badreldin -1986)

- ♦ In this paper the authors describe the results of their investigation into the development of a recognition algorithm for identifying
- ♦ Using a structural classification scheme, the recognition algorithm is derived as a tree classifier.
- ♦ In an extensive test experiment, an accuracy of 99% was realized with isolated numerals. When connected numerals were also included a recognition accuracy of 93% was obtained

References

- 1) **A NOVEL METHOD FOR HAND WRITTEN DIGIT RECOGNITION USING DEEP LEARNING** Rohini.M1 ,
Dr.D.Surendran2 1 Assistant Professor,Sri Krishna College of Engineering and Technology, 2 Professor, Sri Krishna College of Engineering and Technology 1 rohinim@skcet.ac.in, 2 surendran@skcet.ac.in
- 2) **AN EFFECTIVE RESULT-FEEDBACK NEURAL ALGORITHM FOR HANDWRITTEN CHARACTER RECOGNITION** Authors:Xiaoyan Zhu,Yu Hao,Yifan Shi,David Zhang Neural, Parallel & Scientific Computations Volume 9, Issue 2
- 3) **HANDWRITTEN NUMERICAL RECOGNITION BASED ON MULTIPLE ALGORITHMS**
Authors: F.Kimuraa* M.Shridhara The University of Michigan-Dearborn, Dearborn, Michigan, U.S.A.
- 4) **RECOGNITION OF ISOLATED AND SIMPLY CONNECTED HANDWRITTEN NUMERALS**
Authors: M.ShridharA.Badreldin Department of Electrical Engineering, University of Windsor, Windsor, Ontario, Canada N9B 3

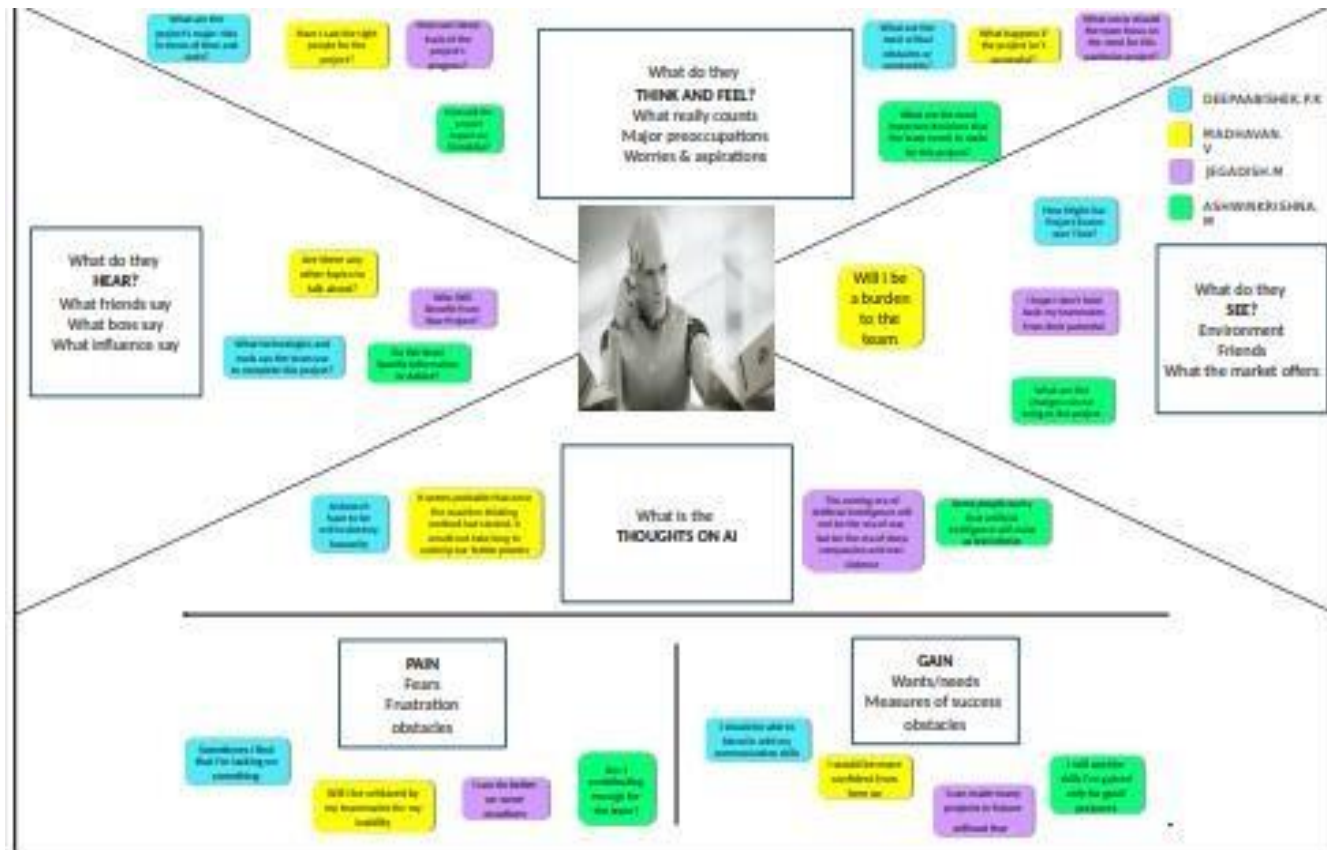
Problem Statement Definition

Handwritten digit recognition is the ability of a computer to recognize digits handwritten by humans. Handwritten digits are imperfect and can be produced in many different flavors, making it a difficult task for machines. The solution to this problem is handwritten digit recognition, which takes images of digits and recognizes the digits in the image. To build a user interface application for scanning a hand written notes to convert digital text in quickly.

3. IDEATION & PROPOSED SOLUTION

Empathy Map Canvas

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes. It is a useful tool to help teams better understand their users. Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.



Ideation and Brainstorming

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem-solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions. Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

Brainstorm, Idea Listing and Grouping

Deepaabishek

Recognize scanned images of handwritten digits	Feature Extraction from Processed Images
Using CNNs to Predict Handwritten Values in Real Time	The CNN has a hidden layer that recognizes the digits

Madhavan

Importing libraries and downloading datasets	Training predictive model with large dataset
models take pictures and categorize them under a certain category	Extracting features from a processed image

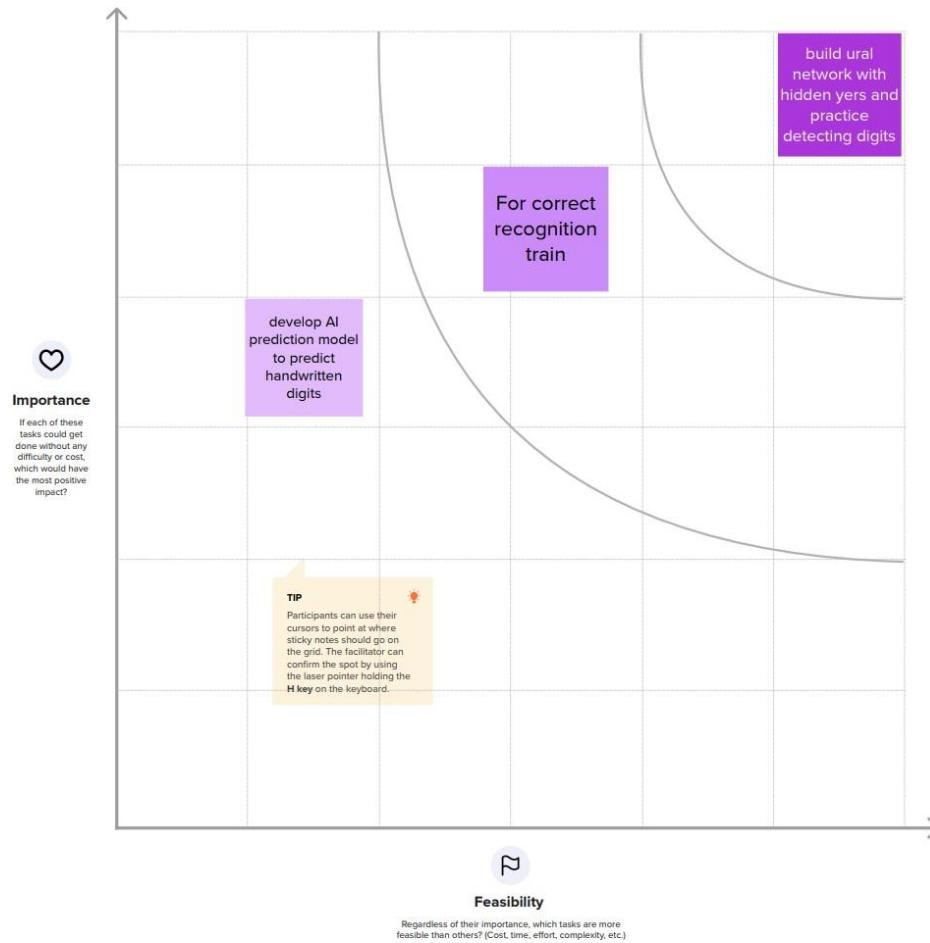
Ashwin Krishna

Using CNN to predict correct handwritten numbers	Image classification and reconnaissance
Large dataset import	numbers can be classified from words, letters and characters

Jegadish

Model Rating	create conditions for people to work in the banking system
Extract object from a processed image	pictures and models classify them under a certain category

Idea Prioritization



3.4 Proposed Solution

Project team shall fill in the following information in the proposed solution template

S.N o.	Parameter	Description
1	Problem Statement (Problem to be solved)	<ul style="list-style-type: none">• This problem is aimed at developing a new handwriting recognition system using ML.• A handwritten digit recognition system is one way to address the problem of taking an image of digits and recognizing the digits present in the image
2.	Idea / Solution description	<ul style="list-style-type: none">• Developing an AI prediction model that predicts handwritten digits and builds a neural network with hidden layers to train digit recognition.

3.	Novelty / Uniqueness	<ul style="list-style-type: none"> The system generates not only digit classifications, but also rich descriptions of instantiation parameters that can provide information such as typefaces.
4.	Social Impact / Customer Satisfaction	<ul style="list-style-type: none"> Handwritten numbers can be recognized easily without strenuous effort. This reduces time and improves everyone's productivity.
5	Business Model (Revenue Model)	<ul style="list-style-type: none"> It is used in vehicle number detection, banks to read checks, post offices to sort mail, and many other tasks
6	Scalability of the Solution	<ul style="list-style-type: none"> To achieve higher performance in the field of character recognition and pattern recognition, thanks to its excellent feature extraction and acts as the best classifier

3.4 Problem Solution fit

Define CS / fit into CC	1. CUSTOMER SEGMENT(S) CS <i>One who wants to extract digits from handwritten text images</i>	6. CUSTOMER CONSTRAINTS CC <i>A blurry image will not give accurate results.</i>	5. AVAILABLE SOLUTIONS <i>Traditional handwriting recognition systems have relied on manual features and a large amount of prior knowledge.</i>	Explore AS, differentiate
Focus on J&P, tap into BE, understand RC	2. JOBS-TO-BE-DONE / PROBLEMS J&P <i>People may find it difficult to read other people's posts. handwritten digits don't always have the same size, width, orientation as they are different in each person's handwriting, so the general problem will be when classifying the digits.</i>	9. PROBLEM ROOT CAUSE RC <i>The issue is that there's a wide range of handwriting - good and bad. This makes it tricky for programmers to provide enough examples of how every character might look.</i>	7. BEHAVIOUR BE <i>Customers must try with clear image and neat handwriting to get accuracy in digits</i>	Focus on J&P, tap into BE, understand RC
Identify strong TR & EM	3. TRIGGERS TR <i>When there is need for recognition of handwritten digits</i> 4. EMOTIONS: BEFORE / AFTER EM <i>frustration, exhausted > curious, satisfied</i>	10. YOUR SOLUTION <i>It uses Artificial Neural Network to recognize them. Neural Network is used to train and identify written digits. After training and testing, the accuracy rate reached 99%. This accuracy rate is very high.</i>	8. CHANNELS of BEHAVIOUR CH 8.1 ONLINE <i>Extract online channels from behaviour block</i> 8.2 OFFLINE <i>Extract offline channels from different handwriting styles</i>	Extract online & offline CH of BE

REQUIREMENT ANALYSIS

Functional requirements

Following are the functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
FR-1	User Registration	Registration through Form Registration through Gmail Registration through Linked IN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Upload image	Image upload via files Image upload via folders Image upload via drive Image upload via web Image upload via scan/camera
FR-4	Spelling support	Identifies handwriting of different styles and fonts Spelling check
FR-5	Translation	Handwritten digits from the image are extracted. Conversion of handwritten digits into machine readable form
FR-6	log out	Log out / sign out.

Non-Functional requirements

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

NFR NO	Non-Functional Requirement	Description
NFR-1	Usability	The proposed system gives good results for images that contain handwritten text written in different styles, different size and alignment with varying background
NFR-2	Security	Only authorized people can access the system data and modify the database
NFR-3	Reliability	The Database is frequently updated with handwriting of different styles and size and will roll back when any update fails.
NFR-4	Performance	The proposed system is advantageous as it uses fewer features to train the neural network, which results in faster convergence.
NFR-5	Availability	The system functionality and services are available for use with all operation
NFR-6	Scalability	The website traffic limit must be scalable enough to support 2 lakhs users at a time.

PROJECT DESIGN

Data Flow Diagrams

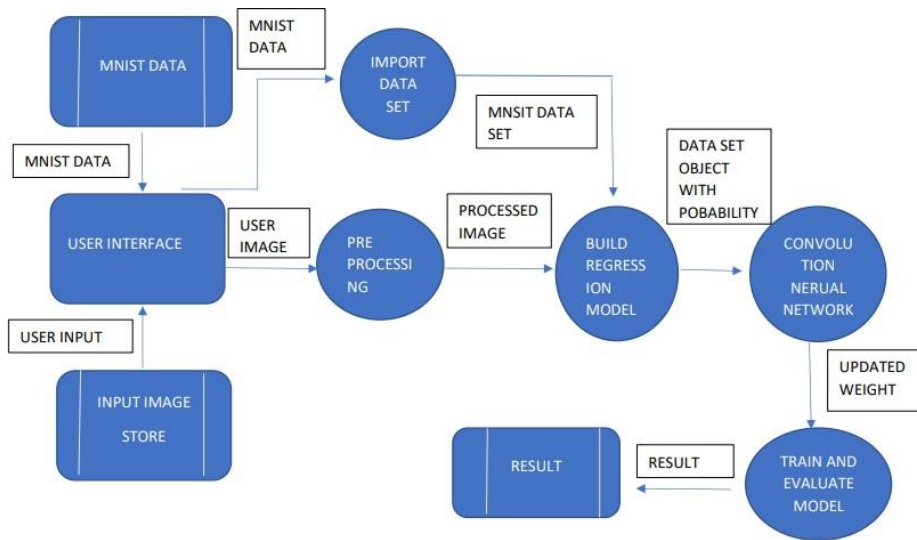
A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

DFD Level-0

Level DFD-0 includes two external entities, the user interface and output, as well as a process that represents the CNN for digit recognition. The output is obtained after processing.

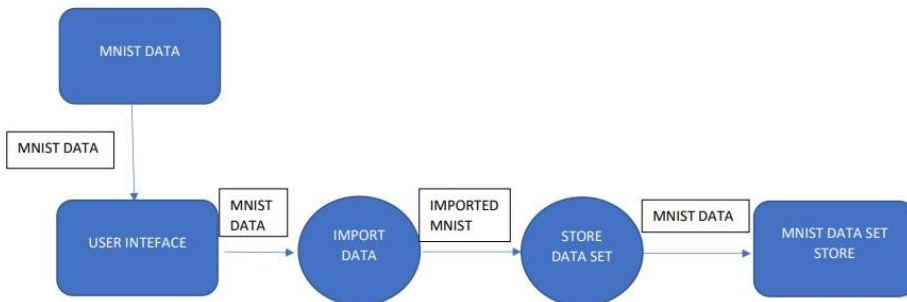
DFD Level-1

Level DFD-1 consists of 2 external entities, a GUI and an output, along with five process blocks and 2 MNIST datastores and input image stores, representing CNN's internal workings for the System. number identification. Block the process of importing MNIST data from the library. The processing block imports the image, processes it, and sends it to the block where the regression model. It sends probabilistic objects to the CNN where the weights are updated and some classes are constructed. Block training and model evaluation to produce outputs.



DFD Level-2

The DFD Level-2 for import data (figure 4) consists of two external data and one entity UI along with three process blocks, representing the three functionalities of the CNN for Digit Recognition System. It imports data from MNIST data store and stores on the system.



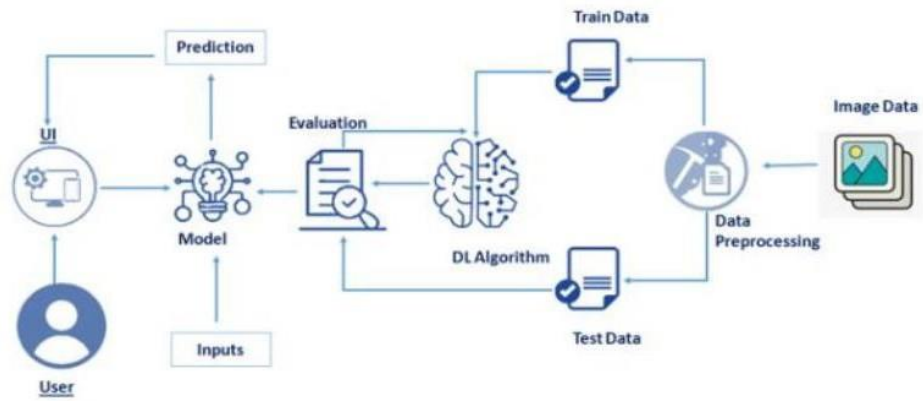
User Stories

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)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-2
		USN-3	As a user, I can register for the application through gmail or facebook	I can register & access the dashboard with Facebook Login	Medium	Sprint-2
	Login	USN-4	As a user, I can log into the application by entering email & password	I can login to the application	High	Sprint-1
	Dashboard	USN-5	Go to dashboard and refer the content about our project	I can read instructions also and the home page is user-friendly.	Low	Sprint-1
	Upload Image	USN-6	As a user, I can able to input the images of digital documents to the application	As a user, I can able to input the images of digital documents to the application	High	Sprint-3
	Predict	USN-7	As a user I can able to get the recognised digit as output from the images of digital documents or images	I can access the recognized digits from digital document or images	High	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.	Medium	Sprint-4
Customer (Web user)	Login	USN-9	As a user, I can use the application by entering my email, password.	I can access my account	Medium	Sprint-4
Customer Care Executive	Dashboard	USN-10	upload the image	Recognize and get the output	High	Sprint-1
Administrator	Security	USN-11	updated the features	checking the security	Medium	Sprint-1

Solution & Technical Architecture



User Stories Table-1:**Components & Technologies:**

S.No	Component	Description	Technology
1.	User Interface	User interacts with the application using the web application.	HTML, CSS, JavaScript / Angular Js / React Js etc.
2.	Application Logic	Login to access the application	Java / Python
5.	Database	Data Type, Configurations etc.	MySQL, NoSQL, etc.
6.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc.
7.	File Storage	Storage of user files of handwritten image	IBM Block Storage or Other Storage Service or Local Filesystem
10.	Machine Learning Model	Machine learning model is used to identify the handwritten image uploaded by users	Object Recognition Model, etc.
11.	Infrastructure (Server / Cloud)	Application Deployment on Local System / AI Local Server Configuration AI Server Configuration	Local, Cloud Foundry, Kubernetes, etc.

Table 2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Machine learning frameworks is used to train a predictive model	PyTorch, Open-cv
2.	Security Implementations	The system will automatically be able to authenticate all users with their unique username and password	Password based login, Authorization
3.	Scalable Architecture	The website traffic limit must be scalable enough to support 2 lakhs users at a time	3-tier
4.	Availability	The system functionality and services are available for use with all operations.	distributed servers
5.	Performance	The app can respond to requests within 5 seconds. It uses fewer features to train the neural network, resulting in faster convergence.	number of requests per sec

5 PROJECT PLANNING AND SCHEDULING

Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data Collection	USN-1	As a user, I can collect the dataset from various resources with different handwritings.	10	Low	Jegadish M Deepaabishek P K
Sprint-1	Data Preprocessing	USN-2	As a user, I can load the dataset, handling the missing data, scaling and split data into train and test.	10	Medium	Jegadish M Deepaabishek P K
Sprint-2	Model Building	USN-3	As a user, I will get an application with ML model which provides high accuracy of recognized handwritten digit.	5	High	Ashwin Krishna M Madhavan V Deepaabishek P K

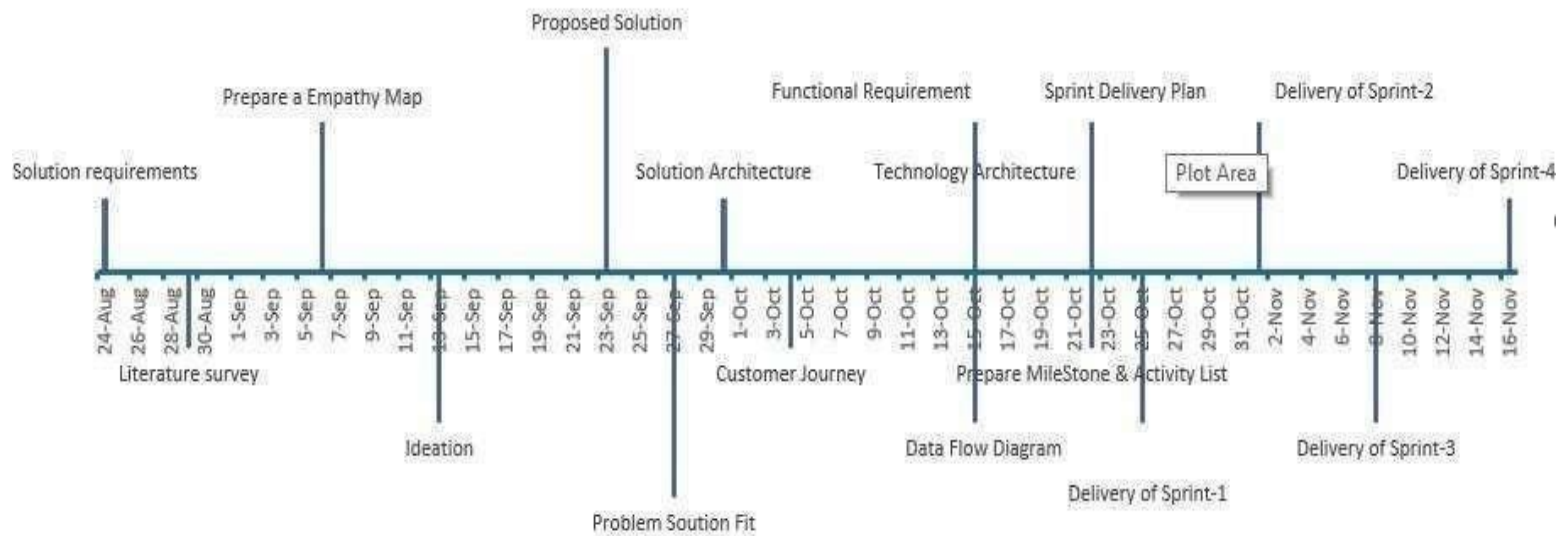
Sprint-2	Add CNN layers	USN-4	Creating the model and adding the input, hidden, and output layers to it.	5	High	Ashwin Krishna M Madhavan V Deepaabishek P K
Sprint-2	Compiling the model	USN-5	With both the training data defined and model defined, it's time to configure the learning process.	2	Medium	Ashwin Krishna M Madhavan V

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-2	Train & test the model	USN-6	As a user, let us train our model with our image dataset.	6	Medium	Ashwin Krishna M Madhavan V
Sprint-2	Save the model	USN-7	As a user, the model is saved & integrated with an android application or web application in order to predict something.	2	Low	Madhavan V
Sprint-3	Building UI Application	USN-8	As a user, I will upload the handwritten digit image to the application by clicking a upload button.	5	High	Ashwin Krishna M Deepaabishek P K
Sprint-3	Building UI Application	USN-9	As a user, I can know the details of the fundamental usage of the application.	5	Low	Deepaabishek P K
Sprint-3	Building UI Application	USN-10	As a user, I can see the predicted / recognized digits in the application.	5	Medium	Jegadish M Deepaabishek P K
Sprint-4	Train the model on IBM	USN-11	As a user, I train the model on IBM and integrate flask/Django with scoring end point.	10	High	Deepaabishek P K Ashwin Krishna M Madhavan V
Sprint-4	Cloud Deployment	USN-12	As a user, I can access the web application and make the use of the product from anywhere.	10	High	Jegadish M Deepaabishek P K

6.2 Sprint Delivery Schedule

A milestone schedule, or milestone chart, is a timeline that uses milestones to divide a project schedule into major phases. A milestone chart is a way to visualize the most important steps of our project. Each milestone the team achieves brings us closer to completing the project. As a result, milestones provide a sense of accomplishment and show the team how the work they're doing contributes to the overarching project objective.

Milestone Timeline Chart



Project Tracker, Velocity & Burndown Chart:

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)
Sprint-1	20	2.5 Days	11 Nov 2022	13 Nov 2022	20
Sprint-2	20	2.5 Days	13 Nov 2022	15 Nov 2022	20
Sprint-3	20	2.5 Days	15 Nov 2022	17 Nov 2022	20
Sprint-4	20	2.5 Days	17 Nov 2022	19 Nov 2022	20

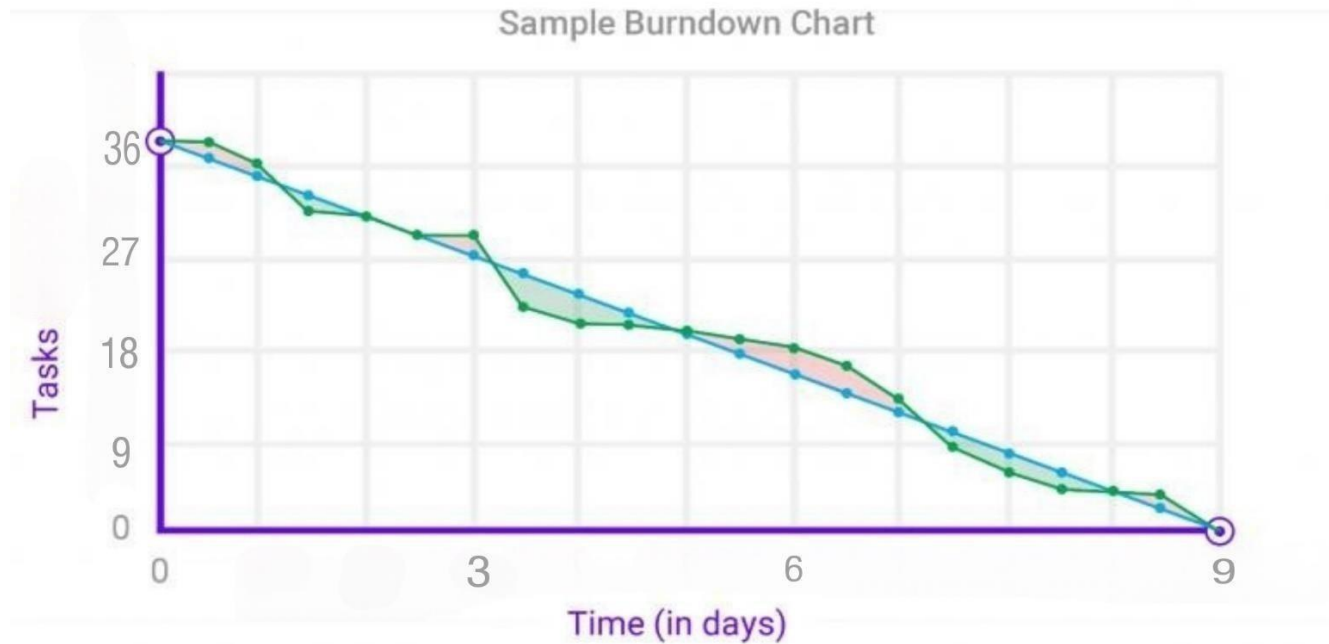
Velocity:

Imagine we've got a 9-day dash duration, and the speed of the group is 20 (factors consistent with dash). Let's calculate the group's common velocity (AV) consistent with generation unit (tale factors consistent with day)

$$\text{Average Velocity} = 9 / 2.5 = 3.6$$

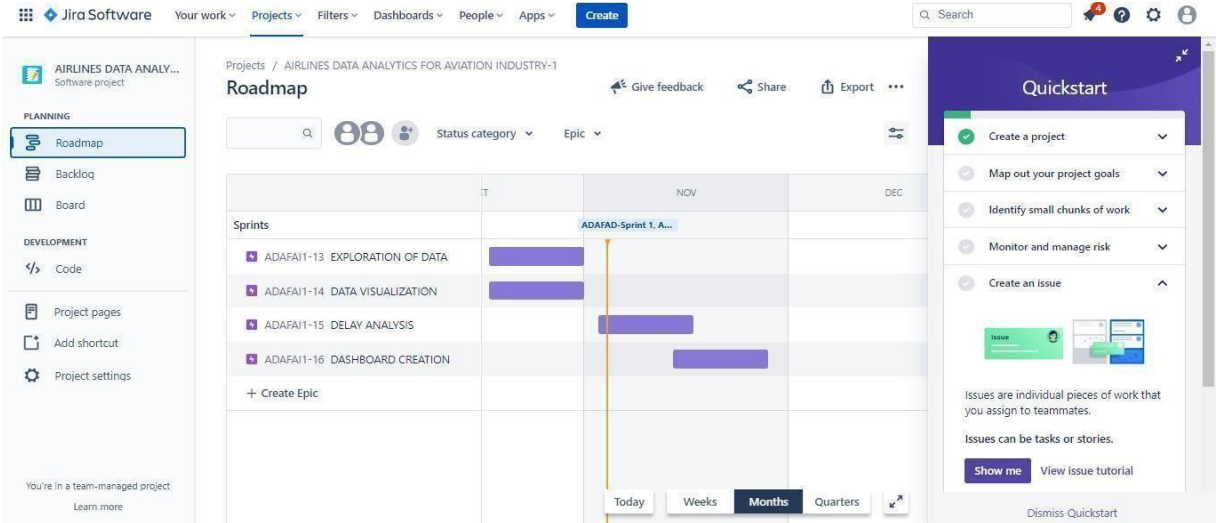
Burndown Chart:

The log chart is a graphical representation of the remaining work to be done as a function of time. It is often used in agile software development methodologies like Scrum. However, log charts can be applied to any project with measurable progress over time.



6. REPORT FROM JIRA

Project RoadMap



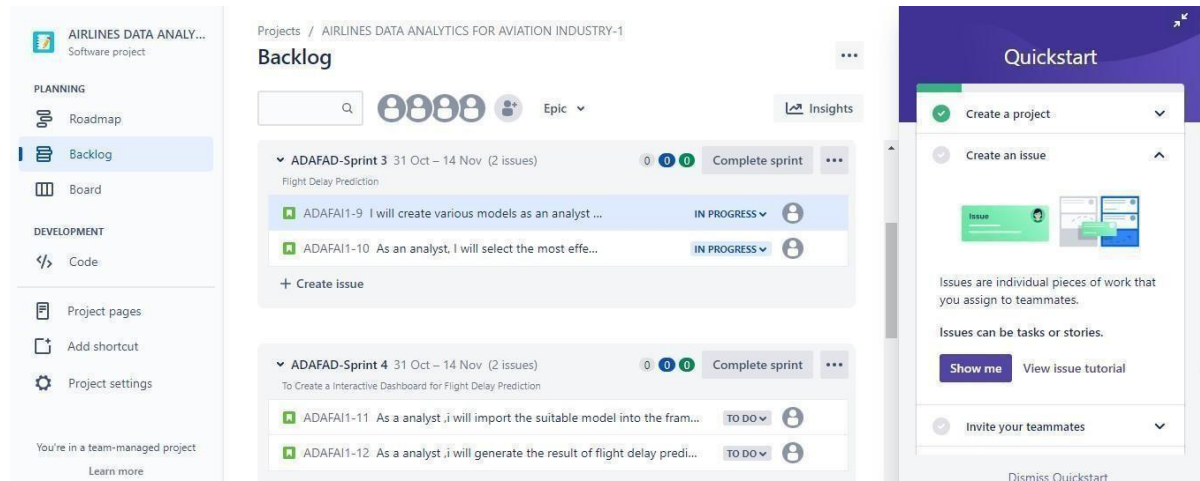
Project Backlog

The image displays three screenshots of the Jira Project Backlog interface, illustrating different views and settings for a software project.

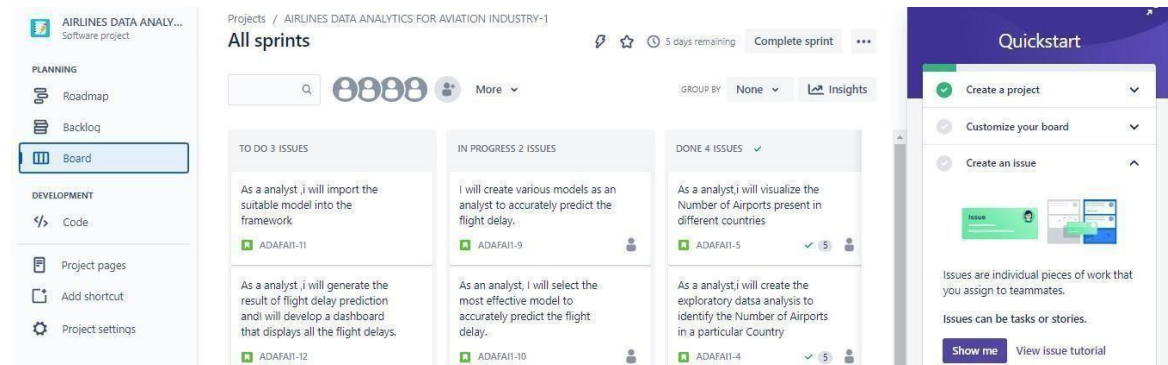
Top Screenshot: Shows the "Backlog" view for a project named "Software project". The left sidebar lists navigation options: PLANNING (Roadmap, Backlog, Board), DEVELOPMENT (Code), and Project pages. The main area displays the "Backlog" for "ADAFAD-Sprint 2" (31 Oct - 14 Nov, 2 issues). The sprint progress bar shows 0/0/10. Two issues are listed: "ADAFAD-5 As a analyst,i will visualize the Number of Airport..." (DONE) and "ADAFAD-4 As a analyst,i will create the explorato..." (DONE). A "Quickstart" panel on the right offers options to "Create a project" and "Create an issue", with a note: "Issues are individual pieces of work that you assign to teammates."

Middle Screenshot: Similar to the top screenshot, but the "ADAFAD-Sprint 2" is expanded, showing the "To do Exploratory data analysis and Visualization for Flight dataset" description. The "ADAFAD-5" issue is now marked "DONE" and the "ADAFAD-4" issue is marked "DONE".

Bottom Screenshot: Shows the "Code" view in the left sidebar. The main area displays the "Backlog" for "ADAFAD-Sprint 1" (31 Oct - 14 Nov, 3 issues). The sprint progress bar shows 0/0/0. Three issues are listed: "ADAFAD-19 As a user, I can view the data" (DONE), "ADAFAD-21 As a analyst,I can explore the data and prepro..." (DONE), and "ADAFAD-20 As a analyst ,i will create the Data module" (TO DO). A "Quickstart" panel on the right offers options to "Create a project", "Create an issue", and "Invite your teammates", with a note: "Issues are individual pieces of work that you assign to teammates. Issues can be tasks or stories." and a "Show me" button for the "View issue tutorial".



Board



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

Feature 1

The user can enter the Elevation feet of the flight to predict whether the delay has occurred or not.

CODE:

```
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)

```

Feature 2

If a delay occurred, the delay is predicted using the Elevation_ft parameter given in the dataset which provides the delay, in minutes.

CODE:

```

<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">&nbsp; Clear &nbsp;</button>
    </div>
  </form>
</div>
</section>

```

```

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

```

Feature 3

This feature can be used to clear the image if we uploaded a wrong image or if we need to change the image. The clear button clears both the image value and the preview of the image in script tag.

```

<script>

```

```

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

```

```

</script>

```

8. TESTING

8.1. Test Cases

				Date	03-Nov-22				
				Team ID	PNT2022TMID17847				
				Project Name	Project - Flight Aviation Industry				
				Maximum Marks	4 marks				
Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status
Main Page	UI	Home Page	User can explore the Web App .		Visit the web page URL and click GO		Elevation details entries should be displayed.	Working as expected	Pass
Entering parameter_TC_001	Functional	Home Page	Verify the UI elements in the main page.		1.Click on the CHECK button displayed on the bottom of the application to check the delay.		Application should show below UI elements: a.Elevation feet Entry Area b.Checking the delay by CHECK button.	Working as expected	Pass
Navigation to Resultpage_TC_002	Funtional	Home Page	Results will displayed with the analysed delay.		Delay analysis is done if occurs.	Elevation_ft ID: 200 Delay predicted: 10 mins	Application should show correct delay time in minutes.	Working as expected	Pass
Return to Homepage_TC_001	Functional	Second page	To check the delay for another elevation feet .		1.Click on the CHECK button displayed on the bottom of the application to check the delay.	Elevation_ft ID: 2391 Delay predicted: 100 mins	User should be navigated from the loginpage to the dashboard. The Dashboard displays the User Name.	Working as expected	Pass

8.2 User Acceptance Testing:

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the Airlines Data Analytics for Aviation Industry project at the time of the release to User Acceptance Testing (UAT).

2. Defect Analysis

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

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	2	3	20
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	24	14	13	26	77

3. Test Case Analysis

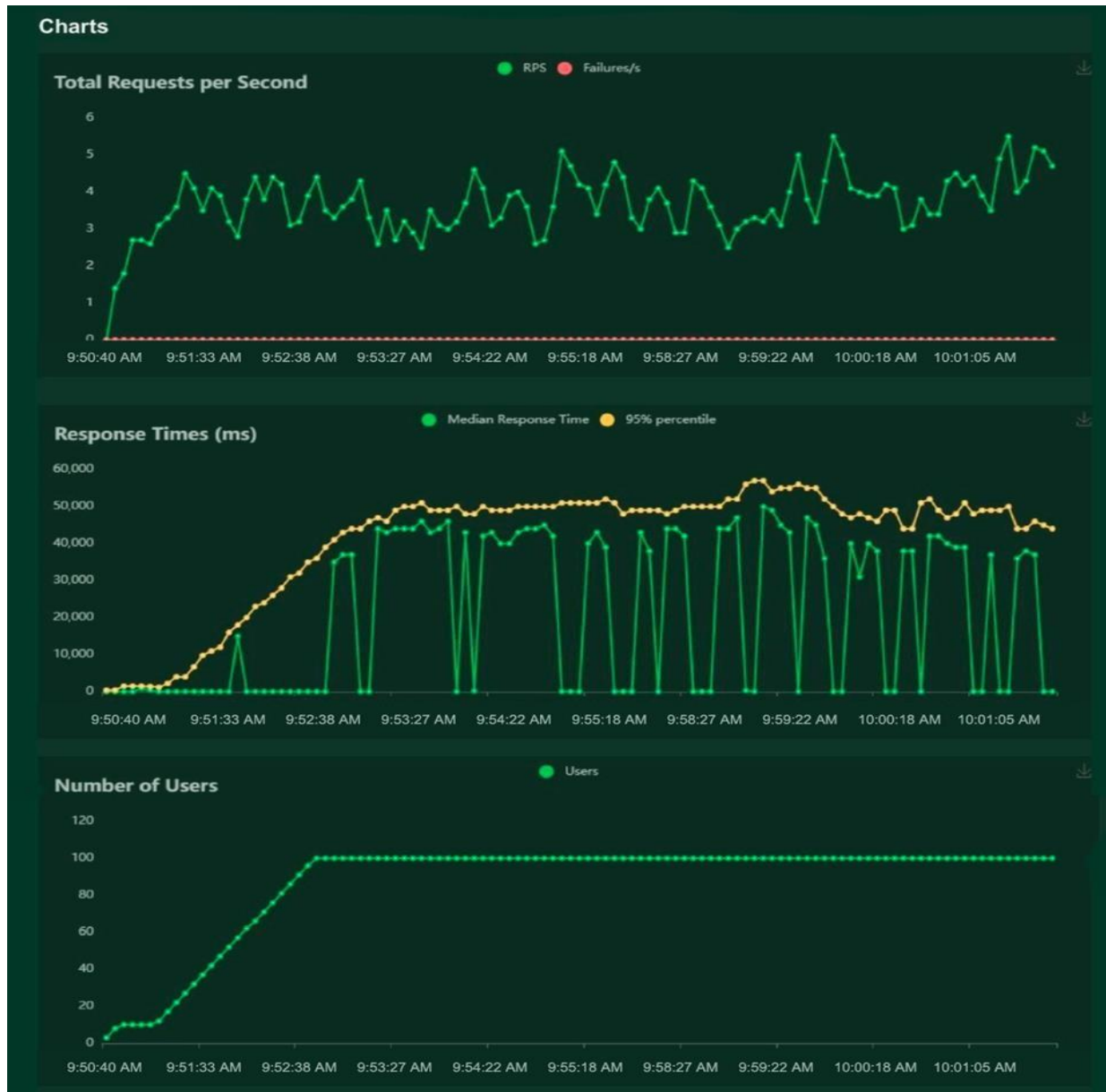
This report shows the number of test cases that have passed, failed, and untested

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	30	0	0	30
Security	2	0	0	2

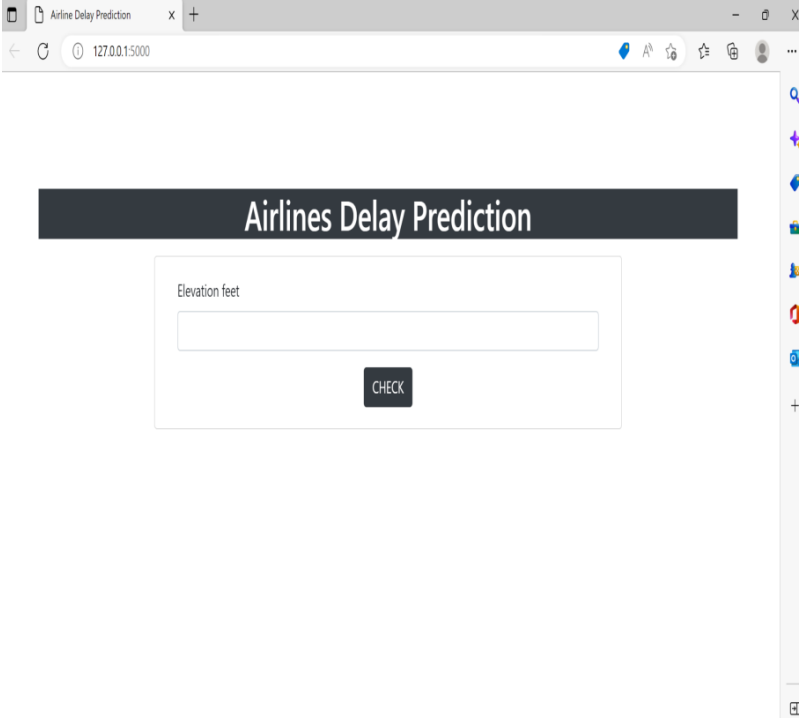
8. RESULTS

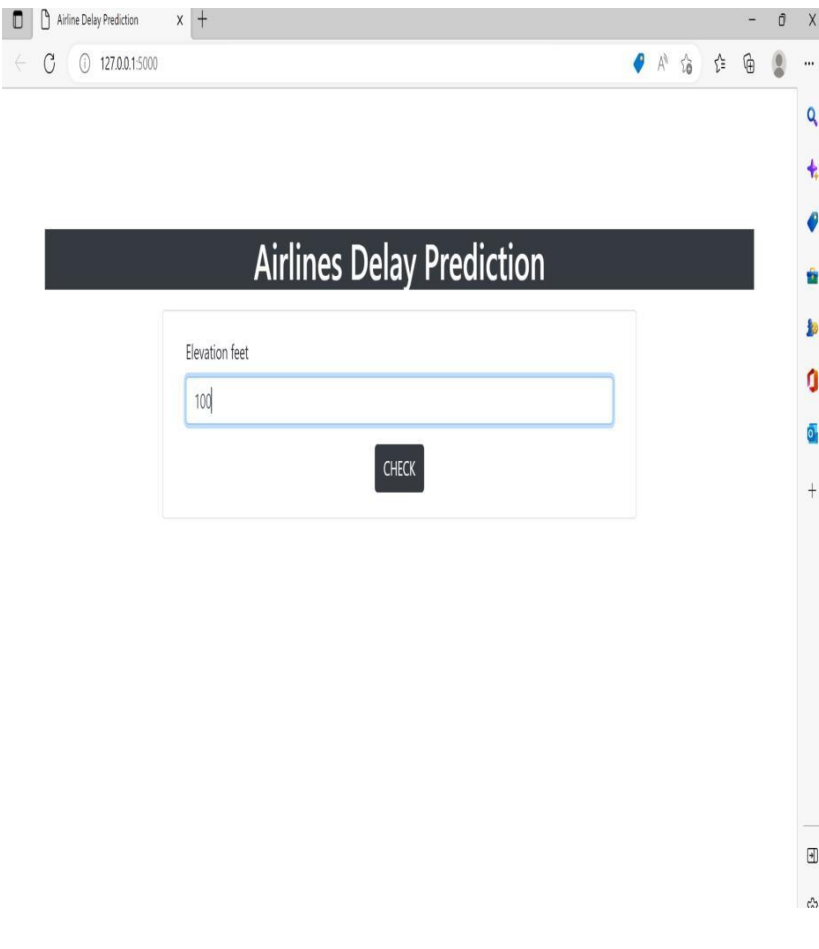
Performance metrics:

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



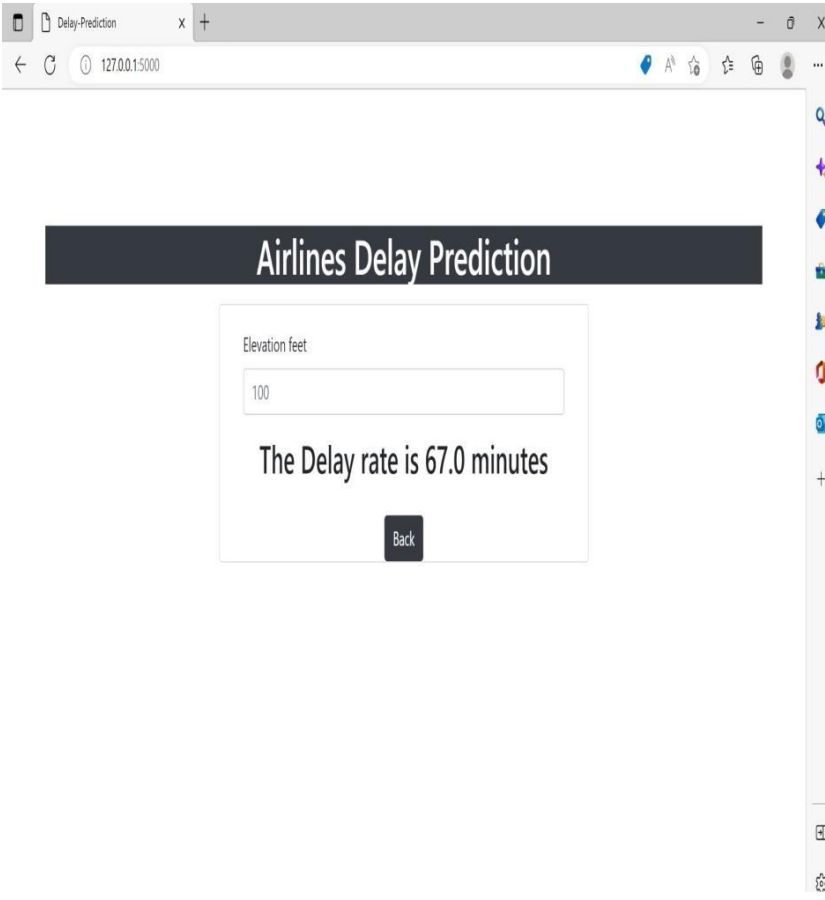
The project team shall fill in the following information in the model performance testing template.

S.No	Parameter	Screenshot
1	Dashboard design	 A screenshot of a web browser displaying a dashboard titled "Airlines Delay Prediction". The dashboard features a dark header bar with the title in white. Below the header, there is a light gray box containing a text input field labeled "Elevation feet" and a dark gray button labeled "CHECK". The browser's address bar shows the URL "127.0.0.1:5000". The browser's tab bar shows a single tab titled "Airlines Delay Prediction". The browser's sidebar on the right shows various icons for search, home, and other functions.

2	Data entry	
3	Data responsiveness	The Delay rate is 67.0 minutes

4

Descriptive report of delay analysis



The screenshot shows a web browser window with a single tab titled "Delay-Prediction". The address bar displays "127.0.0.1:5000". The webpage has a dark header bar with the text "Airlines Delay Prediction". Below the header, there is a light gray box containing a label "Elevation feet" above a text input field with the value "100". Underneath the input field, the text "The Delay rate is 67.0 minutes" is displayed. At the bottom right of this box is a dark button labeled "Back".

9. ADVANTAGES & DISADVANTAGES

Advantages

- Reduced manual work
- More accurate than the average human
- Can handle large amounts of data
- Available anywhere from any device

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

10. CONCLUSION

This project demonstrated a web application that uses machine learning to recognize handwritten digits. Flask, HTML, CSS, JavaScript, and other technologies were used to create this project. This model uses a CNN network to predict handwritten digits. During testing, the model achieved a recognition rate of 99.61%. The proposed project is scalable and can easily handle large numbers of users. Since it is a web application, it can be used on any device that can run a browser. This project is very useful in real-world scenarios such as recognizing vehicle license plates, processing bank check amounts, and entering numbers in handwriting forms (tax forms). There is a great deal of room for improvement to be implemented in subsequent versions.

11. FUTURE SCOPE

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

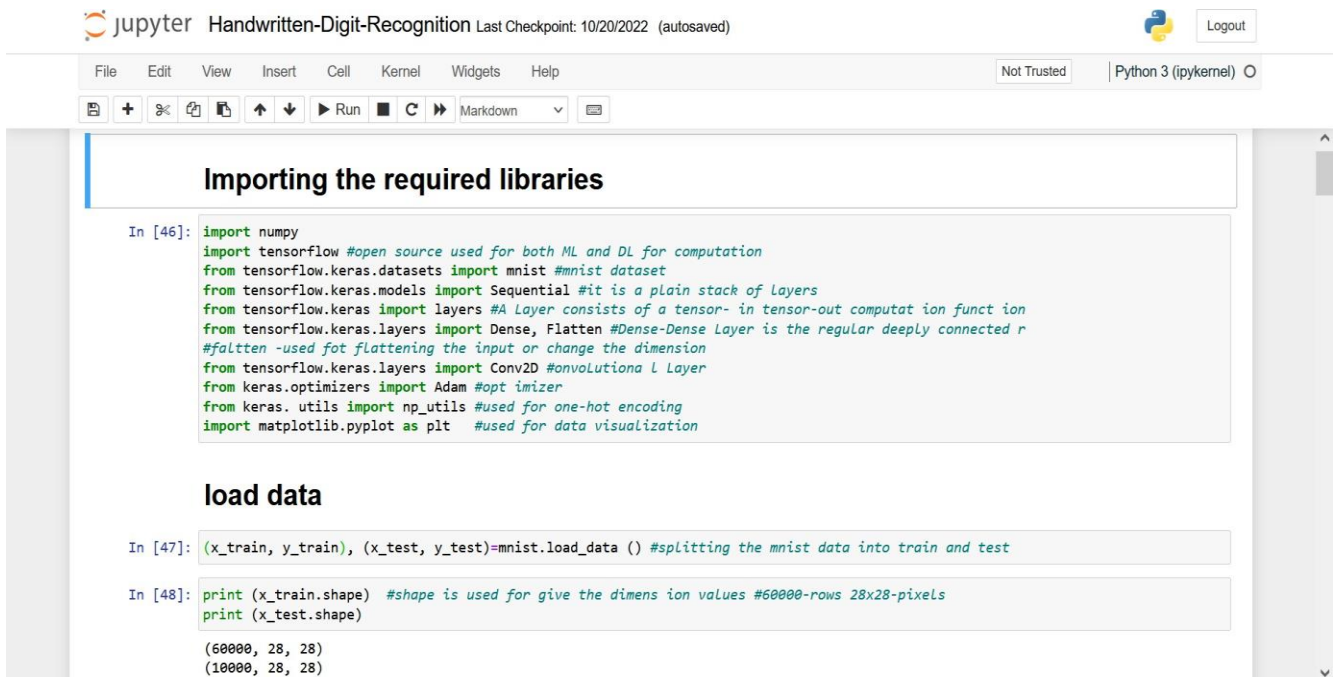
- Add support for digitizing multiple images and saving the results.
- Added support for recognizing multiple digits

- Enhancing the model to recognize digits from complex images
- Add support for different languages to help users around the world

This project has endless possibilities and can always be improved and made better. Implementing this concept in the real world will benefit multiple industries, reduce the workload for many workers, and improve overall work efficiency.

12. APPENDIX

MODEL CREATION:



The screenshot shows a Jupyter Notebook titled "Handwritten-Digit-Recognition" with a last checkpoint of "10/20/2022 (autosaved)". The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations, running, and markdown. The notebook content is divided into two sections: "Importing the required libraries" and "load data".

Importing the required libraries

```
In [46]: import numpy
import tensorflow #open source used for both ML and DL for computation
from tensorflow.keras.datasets import mnist #mnist dataset
from tensorflow.keras.models import Sequential #it is a plain stack of Layers
from tensorflow.keras import layers #A Layer consists of a tensor- in tensor-out computat ion funct ion
from tensorflow.keras.layers import Dense, Flatten #Dense-Dense Layer is the regular deeply connected r
#faltten -used fot flattening the input or change the dimension
from tensorflow.keras.layers import Conv2D #onvoLutiona l Layer
from keras.optimizers import Adam #opt imizer
from keras. utils import np_utils #used for one-hot encoding
import matplotlib.pyplot as plt #used for data visualization
```

load data

```
In [47]: (x_train, y_train), (x_test, y_test)=mnist.load_data () #splitting the mnist data into train and test

In [48]: print (x_train.shape) #shape is used for give the dimens ion values #60000-rows 28x28-pixels
print (x_test.shape)

(60000, 28, 28)
(10000, 28, 28)
```

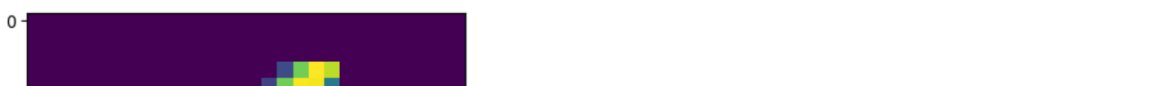


```
In [49]: x_train[0]
```

[illegible]

```
In [50]: plt.imshow(x_train[6000])    #ploting the index=image
```

```
Out[50]: <matplotlib.image.AxesImage at 0x222c71e0250>
```



Reshaping Dataset

```
In [52]: #Reshaping to format which CNN expects (batch, height, width, channels)
x_train=x_train.reshape (60000, 28, 28, 1).astype('float32')
x_test=x_test.reshape (10000, 28, 28, 1).astype ('float32')
```

Applying One Hot Encoding

```
In [53]: number_of_classes = 10 #storing the no of classes in a variable
```

```
In [54]: y_train = np_utils.to_categorical(y_train, number_of_classes) #converts the output in binary format
         y_test = np_utils.to_categorical(y_test, number_of_classes)
```

Add CNN Layers

```
In [55]: #create model
         model=Sequential ()
```

```
In [56]: #adding model Layer
model.add(Conv2D(64, (3, 3), input_shape=(28, 28, 1), activation='relu'))
```

```
In [57]: #flatten the dimension of the image
model.add(Flatten())
```

```
In [58]: #output layer with 10 neurons
model.add(Dense(number_of_classes,activation = 'softmax'))
```

Compiling the model

```
In [59]: #Compile model
model.compile(loss= 'categorical_crossentropy', optimizer="Adam", metrics=['accuracy'])
```

```
In [62]: x_train = np.asarray(x_train)
         y_train = np.asarray(y_train)
```

Train the model

```
In [63]: #fit the model
model.fit(x_train, y_train, validation_data=(x_test, y_test), epochs=5, batch_size=32)

Epoch 1/5
1875/1875 [=====] - 88s 47ms/step - loss: 0.2299 - accuracy: 0.9506 - val_loss: 0.0899 - val_accuracy: 0.9730
Epoch 2/5
1875/1875 [=====] - 81s 43ms/step - loss: 0.0682 - accuracy: 0.9796 - val_loss: 0.0765 - val_accuracy: 0.9780
Epoch 3/5
1875/1875 [=====] - 80s 43ms/step - loss: 0.0493 - accuracy: 0.9838 - val_loss: 0.0888 - val_accuracy: 0.9755
Epoch 4/5
1875/1875 [=====] - 80s 43ms/step - loss: 0.0357 - accuracy: 0.9887 - val_loss: 0.0835 - val_accuracy: 0.9791
Epoch 5/5
1875/1875 [=====] - 89s 48ms/step - loss: 0.0288 - accuracy: 0.9907 - val_loss: 0.1145 - val_accuracy: 0.9708

Out[63]: <keras.callbacks.History at 0x222d90d9db0>
```

Observing the metrics

```
In [64]: # Final evaluation of the model
metrics = model.evaluate(x_test, y_test, verbose=0)
print("Metrics (Test loss &Test Accuracy) : ")
print(metrics)

Metrics (Test loss &Test Accuracy) :
[0.1144733875989914, 0.97079998254776]
```

Test The Model

```
In [65]: prediction=model.predict(x_test[6000:6001])
print(prediction)

1/1 [=====] - 0s 84ms/step
[[1.4689527e-14 1.8748821e-17 2.3109615e-13 9.5624716e-07 3.1683821e-04
 2.4785629e-07 6.1842380e-18 6.2185841e-06 8.8746788e-07 9.9967492e-01]]
```

Save The model

```
In [70]: # Save the model
model.save('models/mnistCNN.h5')
```

CNNPREDICTION:

```
In [2]: from tensorflow.keras.models import load_model
        from keras.preprocessing import image
        from PIL import Image
        import numpy as np

In [3]: model = load_model("mnistCNN.h5")

In [4]: img = Image.open("C:/Users/Dell/PycharmProjects/A-novel-method-for-digit-recognition-system/data/1.png").convert("L") # convert i
img = img.resize( (28,28) ) # resizing of input image
< >

In [5]: img
Out[5]: 1

In [6]: im2arr = np.array(img) #converting to image
im2arr = im2arr.reshape(1, 28, 28, 1) #reshaping according to our requirement

In [7]: pred = model.predict(im2arr)
        print(pred)

1/1 [=====] - 0s 182ms/step
[[2.5381066e-09 4.9758598e-01 6.5878254e-07 3.7901787e-06 5.3061078e-05
 2.5423644e-06 2.0804979e-10 9.8954014e-02 1.2672696e-03 4.0213263e-01]]
```

TRAIN THE MODEL ON IBM:

```
In [1]: from ibm_watson_machine_learning import APIClient
        credentials = {
            "url": "https://jp-tok.ml.cloud.ibm.com",
            "apikey": "BHyalu2c7JN6n9cnvAVULvSKRYFVLMQ_m51toZ9Yk0nS"
        }
        client = APIClient(credentials)

In [2]: def guid_from_space_name(client,deploy):
        space = client.spaces.get_details()
        return (next(item for item in space['resources'] if item['entity']['name']==deploy)['metadata']['id'])

In [3]: space_uid = guid_from_space_name(client,'digitrecognition')
        print("Space UID = " + space_uid)

Space UID = aa24227a-9f01-493f-90e6-1b6132057fc6

In [4]: client.set.default_space(space_uid)
Out[4]: 'SUCCESS'

In [6]: client.repository.download("97d463b1-45ee-47f7-b8af-aed338794ce1", 'DigitRecog_IBM_model.tar.gz')

Successfully saved model content to file: 'DigitRecog_IBM_model.tar.gz'

Out[6]: 'C:\\Users\\Dell\\PycharmProjects\\A-novel-method-for-digit-recognition-system/DigitRecog_IBM_model.tar.gz'
```

HOME PAGE(HTML) – index.html

```
<html>

<head>
  <title>Digit Recognition WebApp</title>

  <meta name="viewport" content="width=device-width">

  <!-- Google Font -->
  <link href="https://fonts.googleapis.com/css2?family=Prompt:wght@600&display=swap"
rel="stylesheet">
  <link href="https://fonts.googleapis.com/css2?family=Varela+Round&display=swap"
rel="stylesheet">
  <link
href="https://fonts.googleapis.com/css2?family=Source+Code+Pro:wght@500&display=swap"
rel="stylesheet">
  <link
href="https://fonts.googleapis.com/css?family=Calistoga|Josefin+Sans:400,700|Pacifico&display=s
wap" 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')}} ">

  <!--font awesome -->
  <script src="https://kit.fontawesome.com/b3aed9cb07.js" crossorigin="anonymous"></script>

  <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-
q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
crossorigin="anonymous"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"
integrity="sha384-
UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1"
crossorigin="anonymous"></script>
  <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"
```

```
integrity="sha384-
JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM"
crossorigin="anonymous"></script>
```

```
<script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@latest"></script>
```

```
</head>
```

```
<script>
```

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

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

```
</script>
```

```
<body>
```

```
<h1 class="welcome">IBM PROJECT
```

```
<div id="team_id">TEAM ID: PNT2022TMID27424</div>
```

```
</h1>
```

```
<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 analysed by the model and the detected result is returned on to UI

```
</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">&nbsp; Clear
&nbsp;</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
```

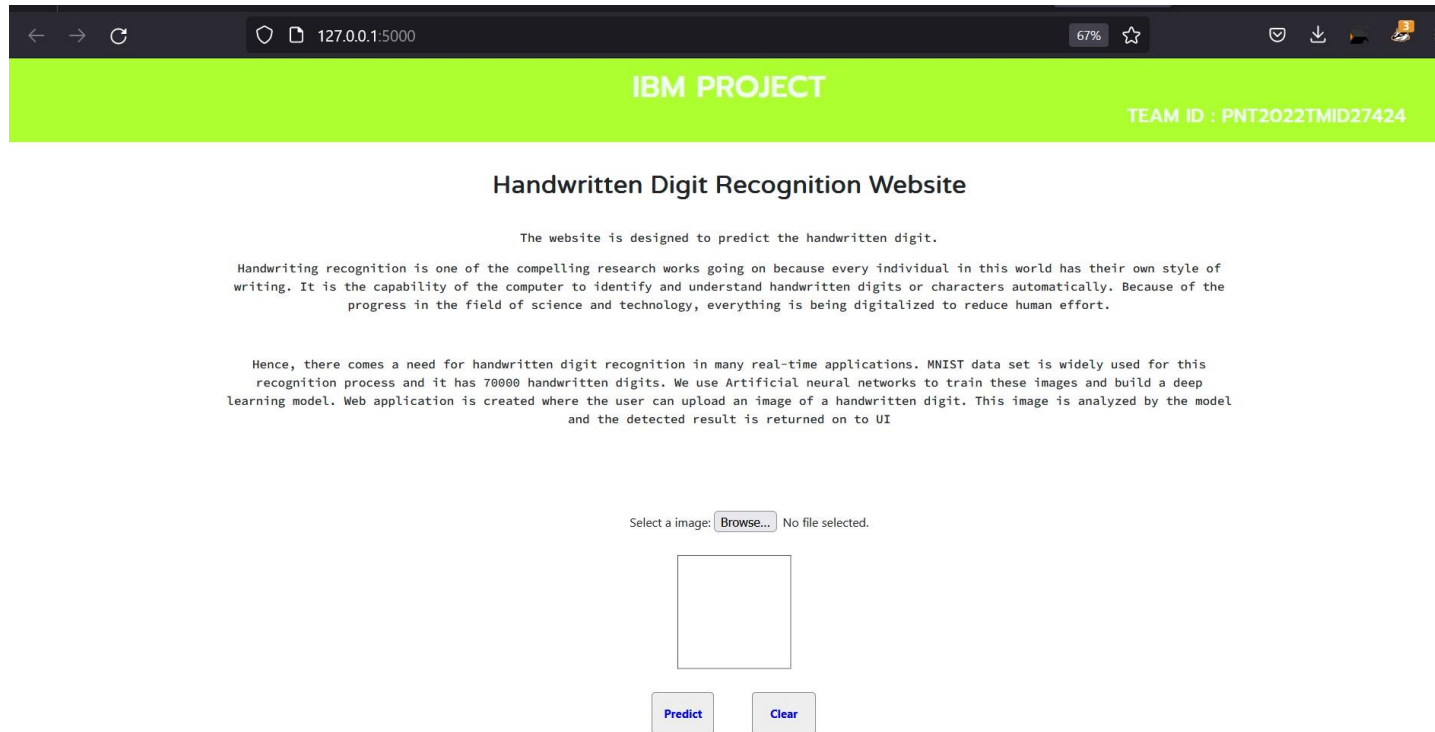
```
pred = model.predict(im2arr)
```

```
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:



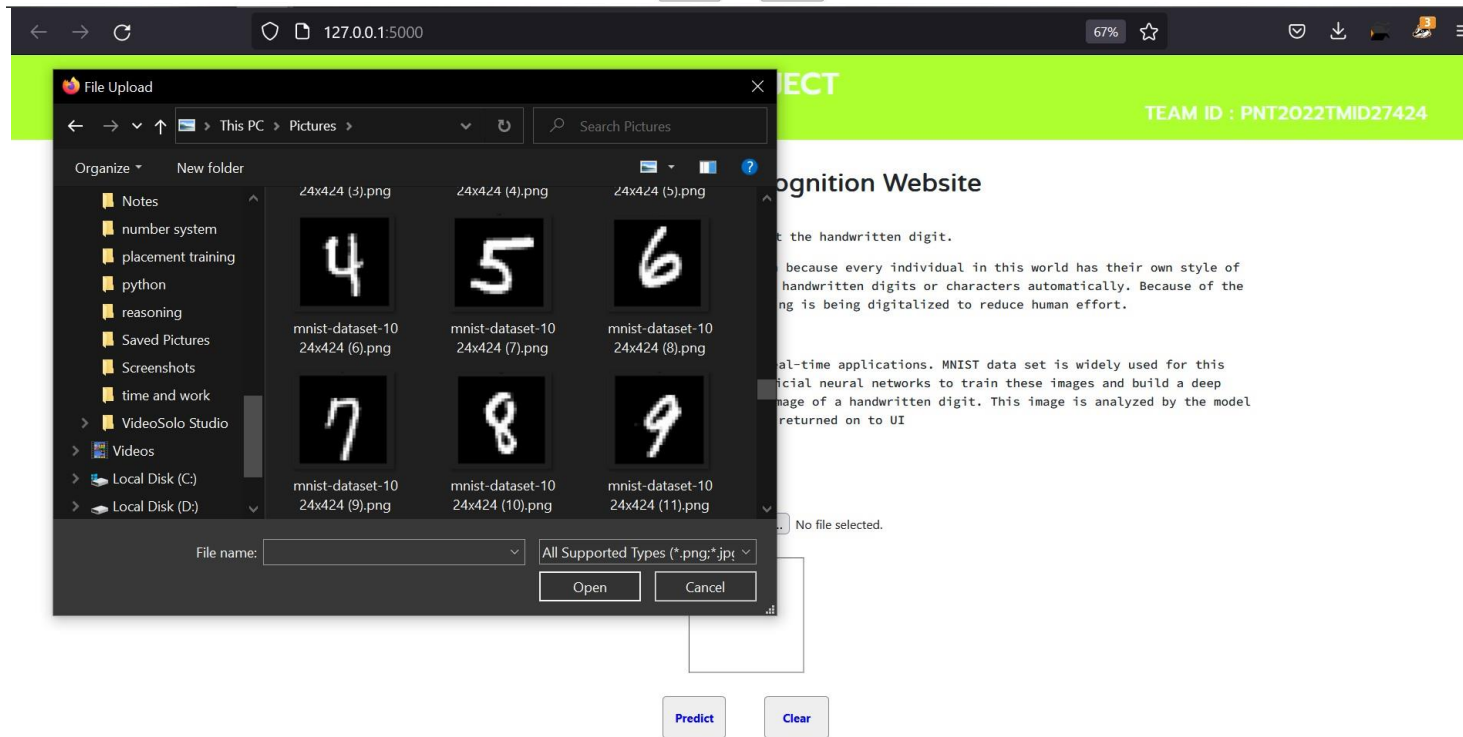
Handwritten Digit Recognition Website

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 an image: mnist-dataset-1024x424 (6).png



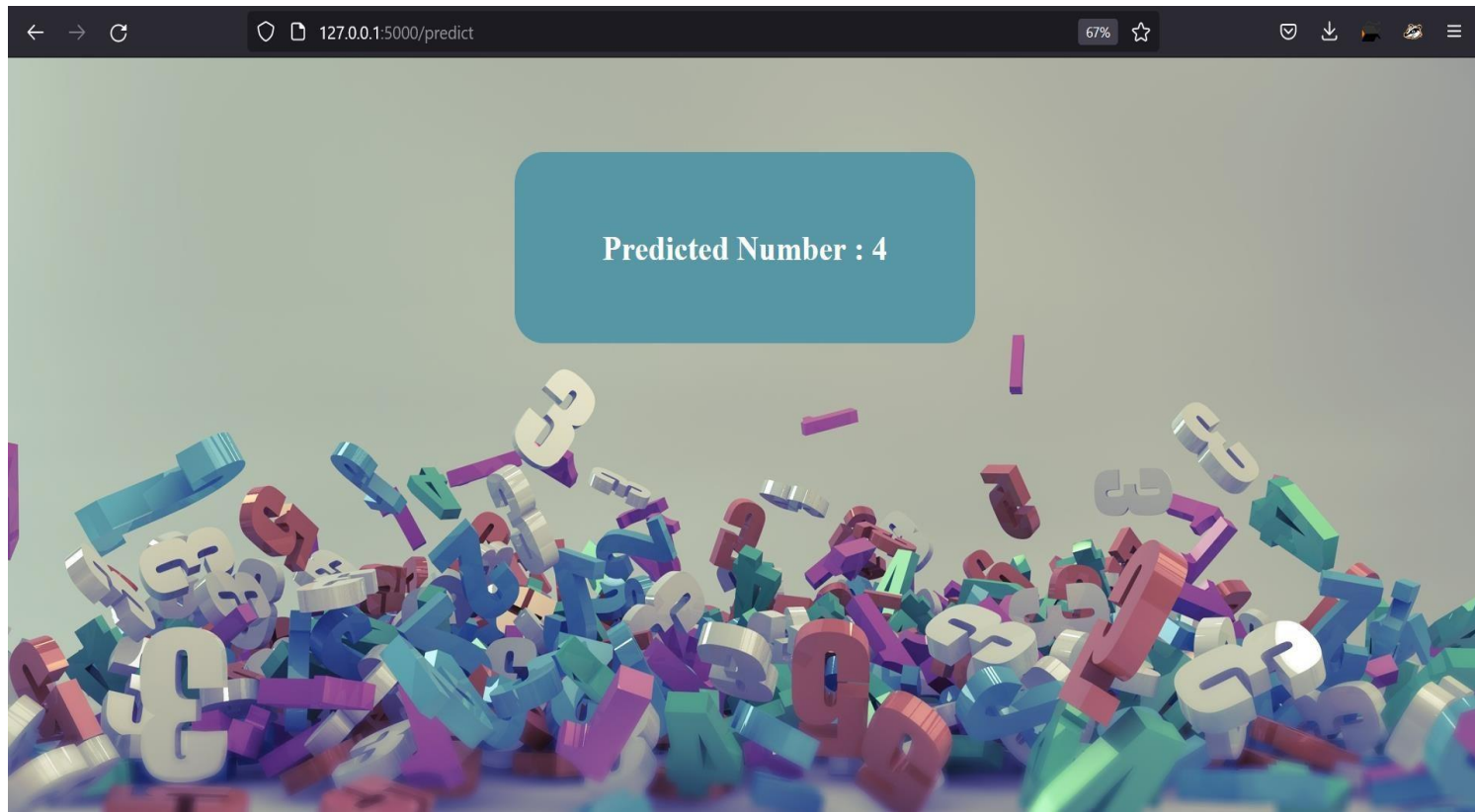
Handwritten Digit Recognition Website

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

No file selected.



GITHUB LINK:

<https://github.com/IBM-EPBL/IBM-Project-20414-1659718946>