PROJECT REPORT

Project Name : **A Novel Method for Handwritten Digit Recognition System**

Team ID : **PNT2022TMID44918**

Team leader : **R.Thirumal (811019104043)**

Team members : **A.Dineshwaran(811019104009)**

A.Ajithkumar(8110191404003)

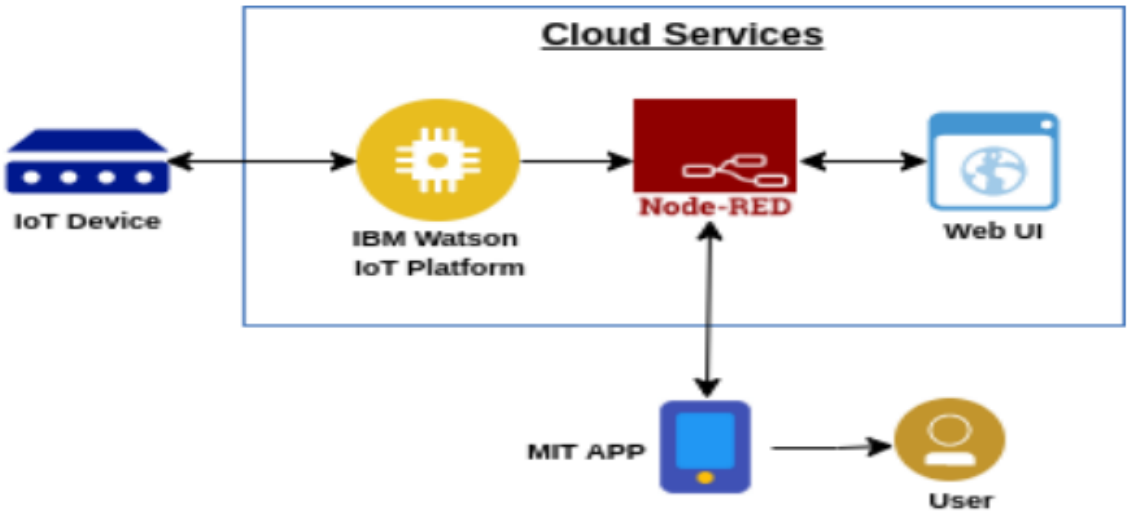
V.Krishnakumar(811019104016)

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
5. PROJECT DESIGN
5.1 Data Flow Diagrams
5.2 Solution & Technical Architecture
6. PROJECT PLANNING & SCHEDULING
6.1 Sprint Planning, Schedule & Estimation
7. CODING & SOLUTIONING (Explain the features added in the project along with code)
7.1 Feature
8. TESTING
8.1 Test Cases
8.2 User Acceptance Testing
9. Save The Model
10. Application Building
11. Advantages & Disadvantages
12. Conclusion
13. Future scope
14. Appendix
Source Code
GitHub & Project Demo Link

1. INTRODUCTION

1.1 Project Overview

- **Know fundamental concepts and techniques of the Artificial Neural Network**
- **Convolution Neural Networks Gain a broad understanding of image data**
- **Work with Sequential type of modeling**
- **Work with Keras capabilities**
- **Work with image processing techniques**
- **know how to build a web application using the Flask framework**



1.2 PURPOSE

Digit recognition systems are able to identify numbers from a variety of sources, including emails, bank checks, papers, images, etc. They can also be used in a variety of real world situations, such as online handwriting recognition on computer tablets or systems, identifying vehicle licence plates, processing bank cheque amounts, and reading numbers from forms that have been filled out by hand (such as tax forms).

2. LITERATURE SURVEY

The challenges of a smart agriculture system include the integration of these sensors and tying the sensor data to the analytics driving automation and response activities. When integrated, the use of data analytics can reduce the overall cost of agriculture and contribute to higher production from the same amount of area through precise control of water, fertilizer and light. Smart methods allow for farming on smaller and more distributed lands through remote monitoring, whether indoor or outdoor.

To successfully deploy a smart agriculture system, consider setting up a communications network that can integrate a limited number of sensors across a large area of farmland. This will require third-party network provisioning or setting up a private network consisting of access points and uplinks to a private backhaul network, which channels all the data traffic to centralized monitoring software or an analytics head-end system

REFERENCES

Handwritten Digit Recognition using Machine Learning Algorithms By S M Shamim, Mohammad Badrul Alam Miah, Angona Sarker, Masud Rana & Abdullah Al Jobair

Handwritten character recognition is one of the practically important issues in pattern recognition applications. The applications of digit recognition includes in postal mail sorting, bank check processing, form data entry, etc. The heart of the problem lies within the ability to develop an efficient algorithm that can recognize hand written digits and which is submitted by users by the way of a scanner, tablet, and other digital devices. This paper presents an approach to off-line handwritten digit recognition based on different machine learning technique. The main objective of this paper is to ensure effective and reliable approaches for recognition of handwritten digits. Several machines learning algorithm namely, Multilayer Perceptron, Support Vector Machine, Naïve Bayes, Bayes Net, Random Forest, J48 and Random Tree has been used for the recognition of digits using WEKA.

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,

Handwritten digit recognition has recently been of very interest among the researchers because of the evolution of various Machine Learning, Deep Learning and Computer Vision algorithms. In this report, We compare the results of some of the most widely used Machine Learning Algorithms like CNN- convolution neural networks and with Deep Learning algorithm like multilayer CNN using Keras with Theano and Tensor flow. MNIST is a dataset which is widely used for handwritten digit recognition. The dataset consist of 60,000 training images and 10,000 test images. The artificial neural Networks can all most mimic the human brain and are a key ingredient in image processing field. For example Convolution Neural networks with back propagation for image processing. The applications where these handwritten digit recognition can be used are Banking

sector where it can be used to maintain the security pin numbers, it can be also used for blind peoples by using sound output.

A Novel Handwritten Digit Classification System Based on Convolutional Neural Network Approach

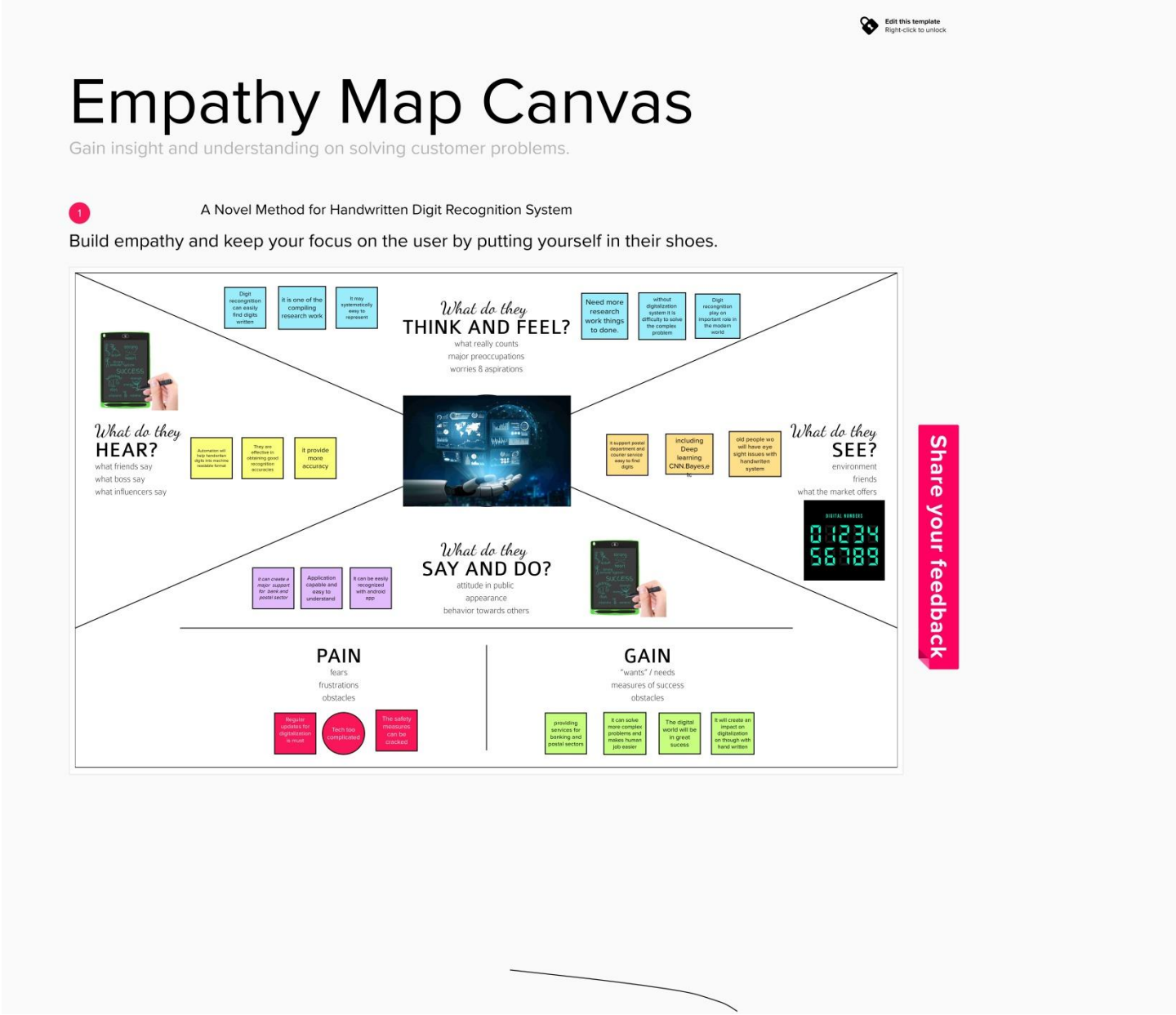
An enormous number of CNN classification algorithms have been proposed in the literature. Nevertheless, in these algorithms, appropriate filter size selection, data preparation, limitations in datasets, and noise have not been taken into consideration. As a consequence, most of the algorithms have failed to make a noticeable improvement in classification accuracy. To address the shortcomings of these algorithms, our paper presents the following contributions: Firstly, after taking the domain knowledge into consideration, the size of the effective receptive field (ERF) is calculated. Calculating the size of the ERF helps us to select a typical filter size which leads to enhancing the classification accuracy of our CNN. Secondly, unnecessary data leads to misleading results and this, in turn, negatively affects classification accuracy. To guarantee the dataset is free from any redundant or irrelevant variables to the target variable, data preparation is applied before implementing the data classification mission.

2 PROBLEM STATEMENT DEFINITION

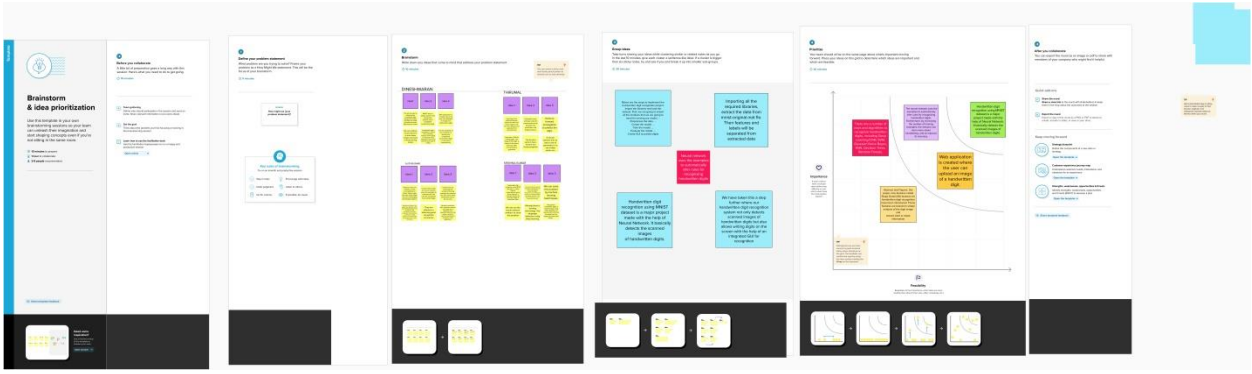
For years, the traffic department has been combating traffic law violators. These offenders endanger not only their own lives, but also the lives of other individuals. Punishing these offenders is critical to ensuring that others do not become like them. Identification of these offenders is next to impossible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

IDEATION AND PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 IDEATION & BRAINSTORMING



3.3 PROPOSED SOLUTION

PROBLEM STATEMENT:

The handwritten digit recognition is the capability of computer applications to recognize the human handwritten digits. It is a hard task for the machine because handwritten digits are not perfect and can be made with many different shapes and sizes. The handwritten digit recognition system is a way to tackle this problem which uses the image of a digit and recognizes the digit present in the image. Convolutional Neural Network model created using PyTorch library over the MNIST dataset to recognize handwritten digits.

IDEA/SOLUTION DESCRIPTION:

MNIST database contains 60,000 training images of handwritten digits¬ from zero to nine and 10,000 images for testing. We will create our CNN model. It works better for data that are¬ represented as grid structures; this is the reason why CNN works well for image classification problems.

NOVELTY/UNIQUENESS:

Handwritten digit recognition using MNIST dataset is a major project made⎫ with the help of neural network. It basically detects the scanned images of handwritten digits. We have taken this a step further where are handwritten digit recognition⎫ system not only detects the scanned images of handwritten digits but also allows writing digits on the screen with the help of an Integrated GUI for recognition.

SOCIAL IMPACT/CUSTOMER SATISFACTION:

Digital Recognition is nothing other than recognizing or identifying digits in any document. The framework of digital recognition is simply the operation of the machine to prepare or interpret digits. Handwritten Digit Recognition is the power of computers to translate handwritten digits from a variety of sources such as text messages, bank checks, papers, photos, etc. method With the use of in-depth learning methods, human efforts can be reduced in perception, learning, perception

and in too many regions. Using in-depth learning, the computer learns to perform distinctive functions in images or content anywhere accuracy, in addition to the performance of the human level. The digital recognition model uses large data sets to detect digits from different sources.

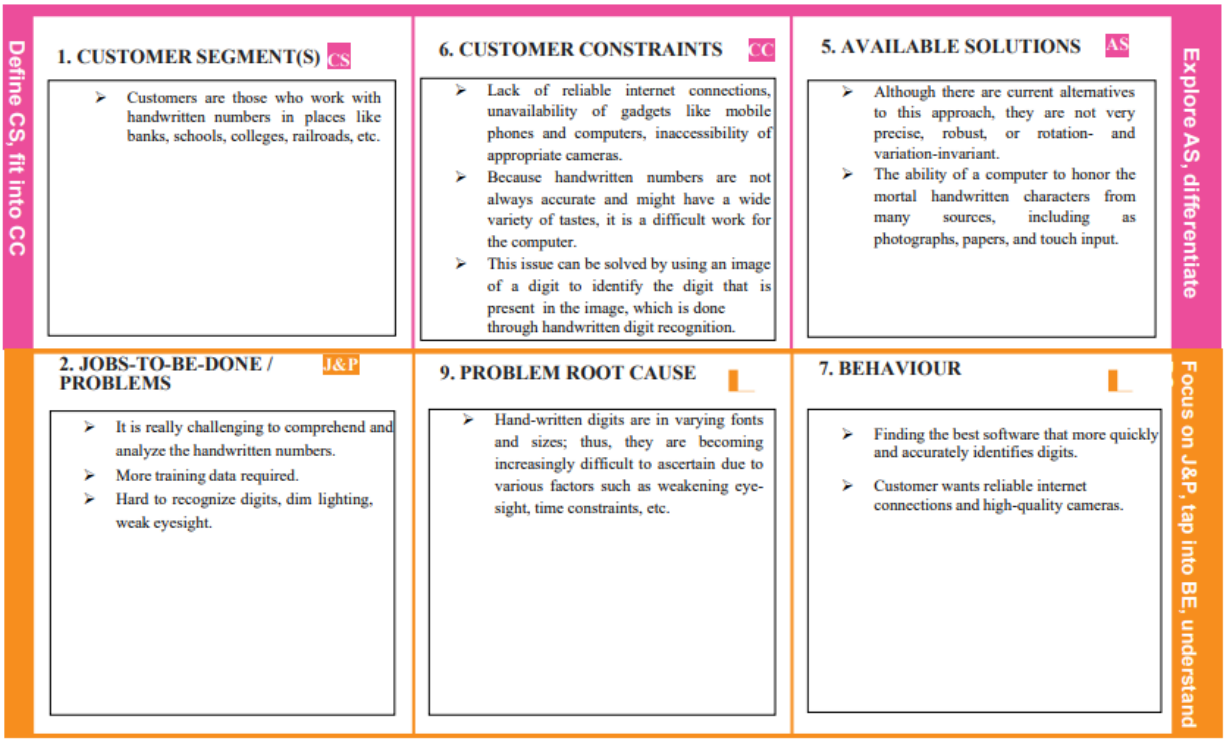
BUSINESS MODEL (FINANCIAL BENEFIT):

Handwritten digit recognition refers to a model's (machine's) capacity to detect⎫ any handwritten digits from various sources, such as photographs, papers, and touch displays, and classify them into ten specified categories 0-9. Several ways and algorithms are used to recognize handwritten digits, such as⎫ Deep Learning/CNN, SVM (Support Vector Machine), Gaussian Naive Bayes, KNN (K-Nearest Neighbour), Decision Trees, Random Forests, etc. We used the CNN (Convolutional Neural network) algorithm to recognize⎫ handwritten digits in this project.

SCALABILITY OF SOLUTION:

The variations of accuracies for handwritten digit were observed for 15 epochs⎫ by varying the hidden layers using CNN model and MNIST digit dataset. The maximum accuracy in the performance was found 99.64% and the total⎫ lowest test loss is 0.0239 approximately.

3.4 PROBLEM SOLUTION FIT



Identify strong TR & EM

3. TRIGGERS

- Obtain the data quickly and accurately.
- The exchange of information is made simple and is one of the simplest ways to speak with a computer and grasp the language.

4. EMOTIONS: BEFORE / AFTER

BEFORE: Uncertain, Reserved, and Perplexed.

AFTER: Assured, Upright, and Rational.

10. YOUR SOLUTION

- The solution aims to reliably recognize hand-written digits using Convolutional Neural Network (CNN) algorithm.Therefore,reducing costs for the company and increasing worker productivity.

8. CHANNELS OF BEHAVIOUR

8.1 ONLINE

- The processing and uploading of the photographs both require a steady internet connection.

8.2 OFFLINE

- Purchase contemporary electronics and confirm their functionality.

Identify strong TR & EM

REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

FR.NO	FUNCTIONAL REQUIREMENTS	SUB REQUIREMENTS
FR-1	Model Creation	Get access the MNIST dataset
		Analyze the dataset
		Define a CNN model
		Train and Test the Model
FR-2	Application Development	Create a website to let the user recognize handwritten digits.
		Create a home page to upload images
		Create a result page to display the results
		Host the website to let the users use it from anywhere
FR-3	Input Image Upload	Let users upload images of various formats.
		Let users upload images of various size
		Prevent users from uploading unsupported image formats

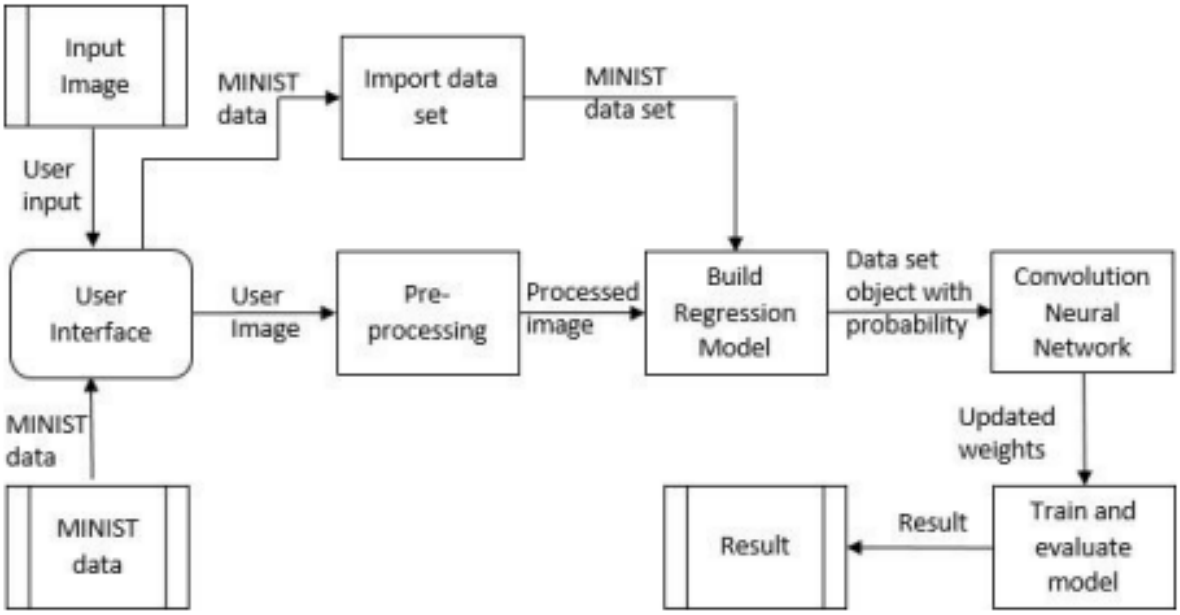
		Pre-Process the image to use it on the model
		Create a database to store all the input images
FR-4	Display Results	Display the result from the model
		Display input image
		Display accuracy the result
		Display other possible predictions with their respective accuracy

4.2 NON FUNCTIONAL REQUIREMENTS

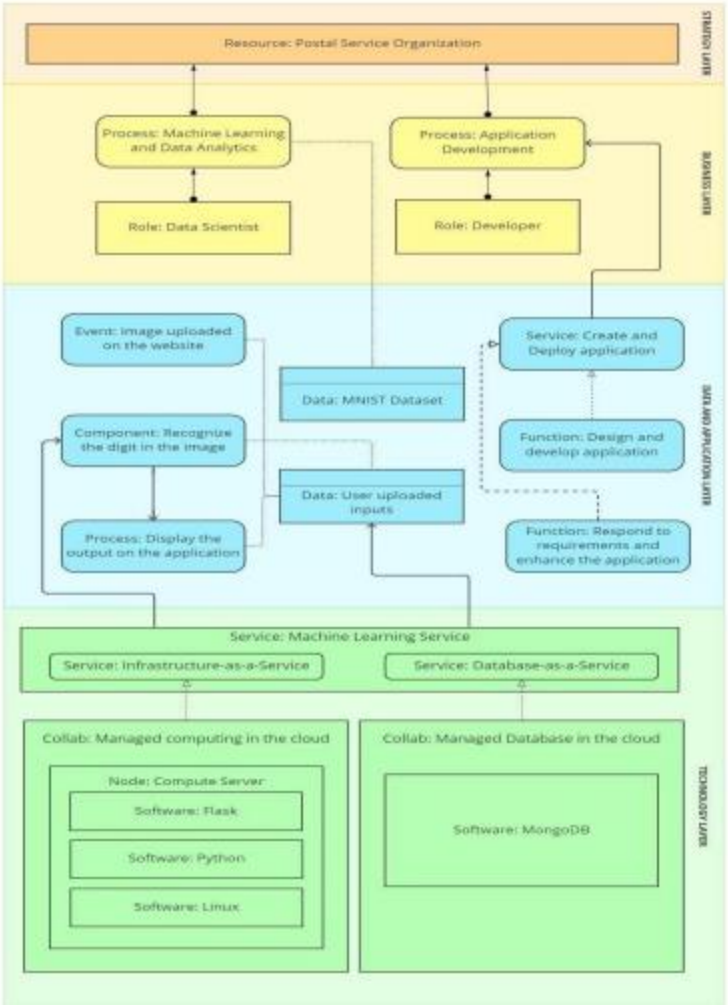
NFR	NON-FUNCTIONAL REQUIREMENTS	DESCRIPTION
NFR-1	Usability	The application must be usable in all devices
NFR-2	Security	The application must protect user uploaded image
NFR-3	Reliability	The application must give an accurate result as much as possible
NFR-4	Performance	The application must be fast and quick to load up
NFR-5	Availability	The application must be available to use all the time
NFR-6	Scalability	The application must scale along with the user base

PROJECT DESIGN

5.1 DATA FLOW DIAGRAM



5.2 SOLUTION & TECHNICAL ARCHITECTURE



PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning, Schedule & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	R.Thirumal
Sprint-1	Login	USN-2	As a user, I can log into the application by entering email & password	1	High	A.Dineshwaran
Sprint-2	Upload Image of digital document	USN-3	As a user, I can able to input the images of digital documents to the application	2	Medium	A.Ajith kumar
Sprint-2	Prediction	USN-4	As a user, I can predict the word	1	Medium	R.Thirumal
Sprint-3	Upload Image of Handwritten document	USN-5	As a user, I can able to input the images of the handwritten documents or images to the application	2	High	A.Dineshwaran
Sprint-3	Recognize text	USN-6	As a user, I can able to choose the font of the text to be displayed	1	Medium	A.Ajith kumar
Sprint-4	Recognize digit	USN-7	As a user I can able to get the recognized digit as output from the images of digital documents or images	1	Medium	K.Krishna kumar
Sprint-4	Recognize digit	USN-8	As a user I can able to get the recognized digit as output from the images of handwritten documents or images	2	High	K.Krishna kumar

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

CODING AND SOLUTIONING

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
#Provide your IBM Watson Device Credentials
organization = "hde0t6"
deviceType = "galaxy"
deviceId = "98765"
authMethod = "token"
authToken = "987654321"
# Initialize GPIO
def myCommandCallback(cmd):
 print("Command received: %s" % cmd.data['command'])
 status=cmd.data['command']
 if status=="lighton":
 print ("led is on")
 elif status == "lightoff":
 print ("led is off")
 else :
 print ("please send proper command")
try:
deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-method": authM
ethod, "auth-token": authToken}
```

```
deviceCli = ibmiotf.device.Client(deviceOptions)
#.....
except Exception as e:
print("Caught exception connecting device: %s" % str(e))
sys.exit()
# Connect and send a datapoint "hello" with value "world" into the cloud as an event of type
 "greeting" 10 times
deviceCli.connect()
while True:
 #Get Sensor Data from DHT11

 temp=random.randint(90,110)
 Humid=random.randint(60,100)

 data = { 'temp' : temp, 'Humid': Humid }
 #print data
 def myOnPublishCallback():
 print ("Published Temperature = %s C" % temp, "Humidity = %s %%" % Humid, "to
IBM Watson")
 success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on_publish=
myOnPublishCallback)
 if not success:
 print("Not connected to IoTF")
 time.sleep(10)

 deviceCli.commandCallback = myCommandCallback
# Disconnect the device and application from the cloud
deviceCli.disconnect()
```

TESTING

8.1 TEST CASES

A Novel Method for Handwritten Digit Recognition System

Test the Model

In [29]:
```
prediction = model.predict(X_test[:4])
print(prediction)
```

```
1/1 [==============================] - 0s 64ms/step
[[4.77358049e-11 1.26020884e-14 2.23637656e-07 2.59297366e-07
  1.53105145e-18 1.41474479e-13 2.73819453e-19 9.99999523e-01
  5.75746352e-12 1.40723442e-08]
 [3.92702641e-05 3.63764530e-09 9.99928832e-01 1.10518204e-06
  3.28396650e-11 1.87219923e-13 3.02575540e-06 4.75269130e-12
  2.79003762e-05 1.17118581e-09]
 [3.37602168e-11 9.99982953e-01 7.10459869e-09 3.63090309e-13
  1.67968246e-05 6.36366426e-09 4.59948364e-11 2.65287614e-09
  2.72516672e-07 1.53049936e-12]
 [9.99999762e-01 1.02759820e-17 6.89465485e-10 4.13503087e-14
  3.53135576e-12 2.56500203e-11 6.89072754e-09 4.50628203e-14
  8.74276596e-10 1.82247064e-07]]
```

In [22]:
```
print(numpy.argmax(prediction, axis=1))
print(Y_test[:4])
```

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

8.2 USER ACCEPTANCE TESTING

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

Save The Model

Your model is to be saved for future purposes. This saved model can also be integrated with an android application or web application in order to predict something.

Saving the model

```
# Save the model
model.save('models/mnistCNN.h5')
```

The model is saved with .h5 extension as follows: An H5 file is a data file saved in the Hierarchical Data Format (HDF). It contains multidimensional arrays of scientific data.

Test With Saved Model

Now open another jupyter file and write the below code

Taking images as input and checking results

```
# Importing the Keras libraries and packages
from tensorflow.keras.models import load_model
model = load_model(r'C:/Users/DELL/Hand written recognition System/models/mnistCNN.h5')
from PIL import Image#used for manipulating image uploaded by the user.
import numpy as np#used for numerical analysis
for index in range(4):
    img = Image.open('data/' + str(index) + '.png').convert("L")# convert image to monochrom
    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
    # Predicting the Test set results
    y_pred = model.predict(im2arr) #predicting the results
    print(y_pred)
```

```
[[1. 0. 0. 0. 0. 0. 0. 0. 0. 0.]]
[[0. 1. 0. 0. 0. 0. 0. 0. 0. 0.]]
[[0. 0. 1. 0. 0. 0. 0. 0. 0. 0.]]
[[0. 0. 0. 1. 0. 0. 0. 0. 0. 0.]]
```

Firstly, we are loading the model which was built. Then we are applying for a loop for the first four images and converting the image to the required format. Then we are resizing the input image, converting the image as per the CNN model and we are reshaping it according to the requirement. At last, we are predicting the result.

Application Building

In this section, we will be building a web application that is integrated into the model we built. A UI is provided for the uses where he has uploaded an image. The uploaded image is given to the saved model and prediction is showcased on the UI. This section has the following tasks

- Building HTML Pages
- Building server-side script

ADVANTAGES & DISADVANTAGES

ADVANTAGES

- ✓ Reduces manual work
- ✓ More accurate than average human
- ✓ Capable of handling a lot of data
- ✓ Can be used anywhere from any device

DISADVANTAGES

- Cannot handle complex data
- All the data must be in digital format
- Requires a high performance server for faster predictions
- Prone to occasional errors

Conclusion

This project demonstrated a web application that uses machine learning to recognise handwritten numbers. Flask, HTML, CSS, JavaScript, and a few other technologies were used to create this project. The model predicts the handwritten digit using a CNN network. During testing, the model achieved a 99.61% recognition rate. The proposed project is scalable and can easily handle a huge number of users.

Since it is a web application, it is compatible with any device that can run a browser. This project is extremely useful in real-world scenarios such as recognizing number plates of vehicles, processing bank cheque amounts, numeric entries in forms filled up by hand (tax forms) and so on. There is so much room for improvement, which can be implemented in subsequent versions

FUTURE SCOPE

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

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

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

Appendix

Source Code

```
<html>

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

 <meta name="viewport" content="width=device-width">
 <!-- GoogleFont -->
 <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&displa
y=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',"");
       });
     });

</script>

<body>

 <h1 class="welcome">IBM PROJECT
 <div id="team_id">TEAM ID : PNT2022TMID44918</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 analyzed by the model and the detected result is returned on to UI</p>

</section>

<section id="content">

<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()">

<button type="submit" class="btn btn-dark" id="predict_button">Predict</button>
<button type="button" class="btn btn-dark" id="clear_button"> Clear </button>

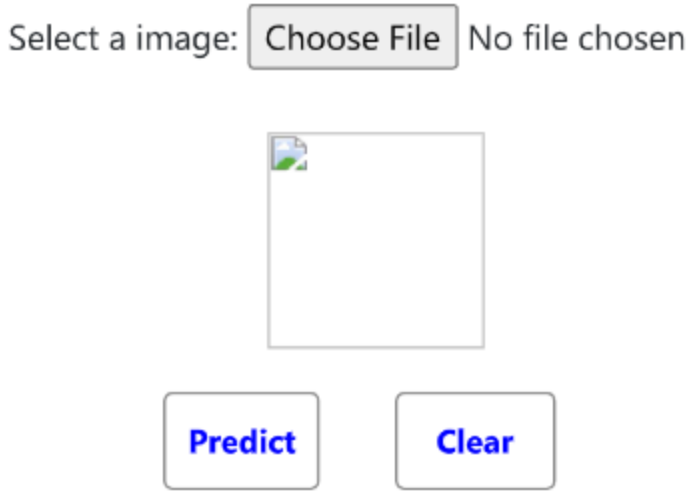
</form>

</section>

</body>

</html>

Ouotput



GITHUB Link :

https://github.com/IBM-EPBL/IBM-Project-6357-1658827145

DEMO VIDEO :

https://drive.google.com/file/d/1YlDMNMeOaKQaYhYLEmihDf4V3LXVktmT/view?usp=sharing