**PROJECT REPORT**

A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

**Submitted By**

**PNT2022TMID15430**

|  |  |  |
| --- | --- | --- |
| S.Chinmaye | - | 111519104155 |
| S.Deepika | - | 111519104144 |
| S.Lasya | - | 111519104121 |
| S.Santosh | - | 111519104127 |

**TABLE OF CONTENTS**

1. **INTRODUCTION** 1.1Project Overview 1.2Purpose
2. **LITERATURE SURVEY** 2.1Existing problem 2.2References

2.3Problem Statement Definition

1. **IDEATION & PROPOSED SOLUTION**
   1. Empathy Map Canvas 3.2Ideation & Brainstorming 3.3Proposed Solution 3.4Problem Solution fit
2. **REQUIREMENT ANALYSIS** 4.1Functional requirement 4.2Non-Functional requirements
3. **PROJECT DESIGN**
   1. Data Flow Diagrams
   2. Solution & Technical Architecture 5.3User Stories
4. **PROJECT PLANNING & SCHEDULING**
   1. Sprint Planning & Estimation 6.2Sprint Delivery Schedule 6.3Reports from JIRA
5. **CODING & SOLUTIONING (Explain the features added in the project along with code)**
6. **TESTING**
   1. Test Cases
   2. User Acceptance Testing
7. **RESULTS**
   1. Performance Metrics
8. **ADVANTAGES & DISADVANTAGES**
9. **CONCLUSION**
10. **FUTURE SCOPE**
11. **APPENDIX**

Source Code

GitHub & Project Demo Link

1. **INTRODUCTION**
   1. **Project Overview**

Machine learning and deep learning play an important role in computer technology and Artificial Intelligence. With the use of Deep Learning and Machine learning, human effort can be reduced in recognizing, learning, predictions and in many more areas.

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

* 1. **Purpose**

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

1. **LITERATURE SURVEY 2.1Existing problem**

* The different architectures of CNN, hybrid CNN,CNN - RNN and CNNHMM models, and domain - specific recognition system, are not thoroughly inquired and evolutionary algorithms are not clearly explored for optimizing CNN learning parameters ,the number of layers, learning rate and kernel sizes of convolutional filters.
* The fluctuation of accuracies for handwritten digits was observed for 15 epochs by varying the hidden layers. There is no clear explanation given for observing variation in the overall classification accuracy by varying the number of hidden layers and batch size.
  1. **References**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **S.NO** | **Author Name** | **Paper Title** | **Journal/ Conference title** | **Page No/ Volume**  **No** | **Year of Publicati on** | **Description** |
|  | Savita | Improved | IEEE Sensors |  | 2020 | In this |
| Ahlawat | Handwritten | Journal |  | paper, with |
| , Amit | Digit |  |  | the aim of |
| Choudh | Recognition |  |  | improving |
| ary, | Using |  |  | the |
| Anand | Convolutiona |  |  | performance |
| Nayyar, | l Neural |  |  | of |
| Saurabh | Networks |  |  | handwritten |
| Singh | (CNN) |  |  | digit |
| and |  |  |  | recognition, |
| Byungu |  |  |  | they |
| n Yoon. |  |  |  | valuated |
|  |  |  |  | variants of a |
|  |  |  |  | convolution |
|  |  |  |  | al neural |
|  |  |  |  | network to |
|  |  |  |  | avoid |
|  |  |  |  | complex |
|  |  |  |  | preprocessin |
|  |  |  |  | g, costly |
|  |  |  |  | feature |
|  |  |  |  | extraction |
|  |  |  |  | and a |
|  |  |  |  | complex |
|  |  |  |  | ensemble |
|  |  |  |  | (classifier |
|  |  |  |  | combination |
|  |  |  |  | ) approach |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  | of a traditional recognition  system. |
|  | Vijayala | Handwritten | International | Volume | 2019 | In this |
| xmi R | Digit | Journal of | -4 Issue- |  | paper, the |
| Rudras | Recognition | Innovative | 6 |  | most widely |
| wamima | using CNN | Science and |  |  | used |
| th, |  | Research |  |  | Machine |
| Bhavani |  | Technology |  |  | learning |
| shankar |  |  |  |  | algorithms, |
| and |  |  |  |  | KNN, SVM, |
| Channas |  |  |  |  | RFC and |
| andra. |  |  |  |  | CNN have |
|  |  |  |  |  | been trained |
|  |  |  |  |  | and tested |
|  |  |  |  |  | on the same |
|  |  |  |  |  | data in order |
|  |  |  |  |  | acquire the |
|  |  |  |  |  | comparison |
|  |  |  |  |  | between the |
|  |  |  |  |  | classifiers |
|  | Fathma | Recognition | 5th |  | 2019 | In this |
| Siddiqu | of | International |  | paper, they |
| e, | Handwritten | Conference |  | observed the |
| Shadma | Digit using | on Advances |  | variation of |
| n Sakib | Convolutiona | in Electrical |  | accuracies |
| and Md. | l Neural | Engineering |  | of CNN to |
| Abu | Network in | (ICAEE) |  | classify |
| Bakr | Python with |  |  | handwritten |
| Siddiqu | Tensorflow |  |  | digits for 15 |
| e. | and |  |  | epochs |
|  | Comparison |  |  | using |
|  | of |  |  | various |
|  | Performance |  |  | numbers of |
|  | for Various |  |  | hidden |
|  | Hidden |  |  | layers and |
|  | Layers |  |  | epochs and |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  | to make the comparison between the accuracies. For this performance evaluation of CNN, they performed the experiment using Modified National Institute of Standards and Technology( MN IST)  dataset. |
|  | Akanks ha Gupta, Ravindr a Pratap Narwari a and Madhav Singh | Review on Deep Learning Handwritten Digit Recognition using Convolutiona l Neural Network | International Journal of Recent Technology and Engineering (IJRTE) | Volume  -9 Issue-  5 | 2021 | In this paper, Object Character Recognition (OCR) is  used on printed or documented letters to convert them into text. The database has training image database of 60,000  images and |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  | testing image database of 10,000  images. The KNN  algorithm describes categorical value by making use of majority of votes of K - nearest neighbors, the K value used to  differ here. |
|  | Md. Anwar Hossain and Md. Mohon Ali | Recognition of Handwritten Digit using Convolutiona l Neural Network (CNN) | Global Journal of Computer Science and Technology: D Neural & Artificial Intelligence | Volume 19  Issue2 | 2019 | The goal of this work will be to create a model that will be able to identify and determine the handwritten digit from its image with better accuracy using using the concepts of Convolution al Neural Network  and MNIST |

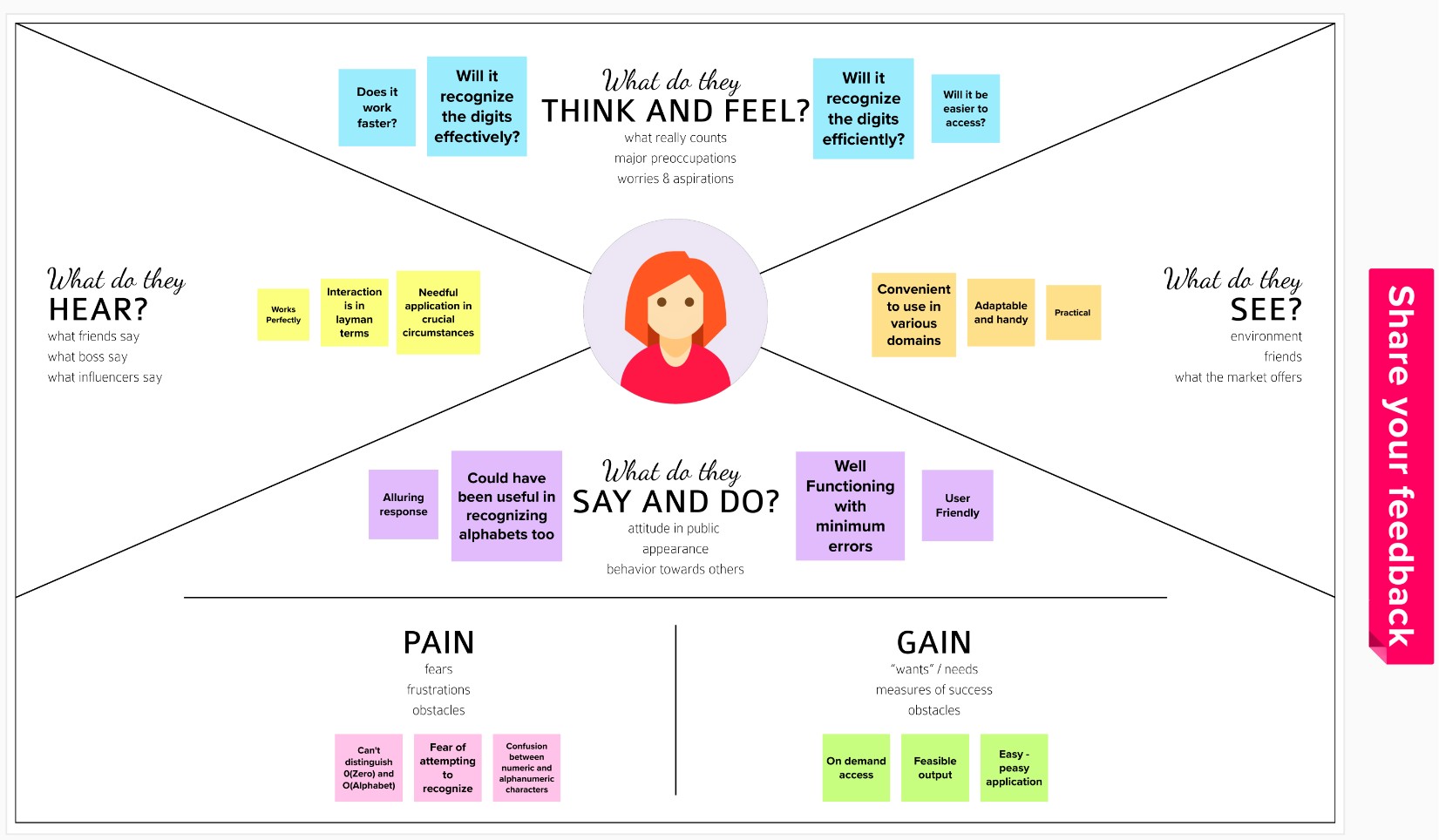
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  | dataset. Later it can be extended for character recognition and real- time person’s handwriting. The results can be made more accurate with more convolution layers and more number of hidden  neurons. |

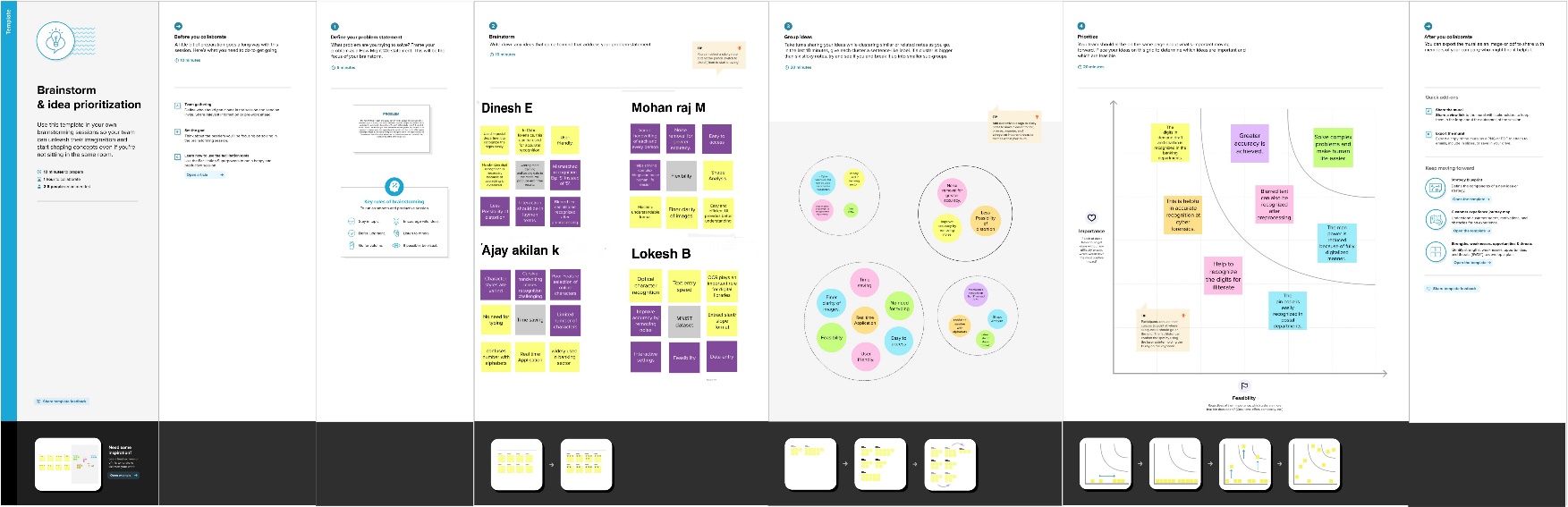
* 1. **Problem Statement Definition**

Handwriting recognition is one of the compelling research works going on because every individual in this world has their own style of writing. It is the capability of the computer to identify and understand handwritten digits or characters automatically.

Because of the progress in the field of science and technology, everything is being digitalized to reduce human effort.

Hence, there comes a need for handwritten digit recognition in many real time applications. MNIST data set is widely used for this recognition process and it has 70000 handwritten digits. We use Artificial neural networks to train these images and build a deep learning model. Web application is created where the user can upload an image of a handwritten digit. this image is analyzed by the model and the detected result is returned on to UI(User Interface).

1. **IDEATION & PROPOSED SOLUTION**
   1. **Empathy Map Canvas**
   2. **Ideation & Brainstorming**

****

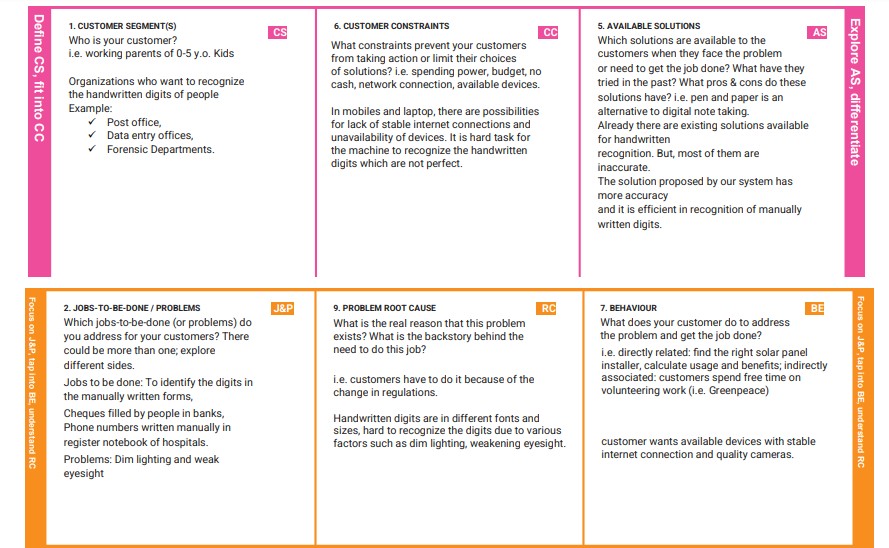
* 1. **Proposed Solution**

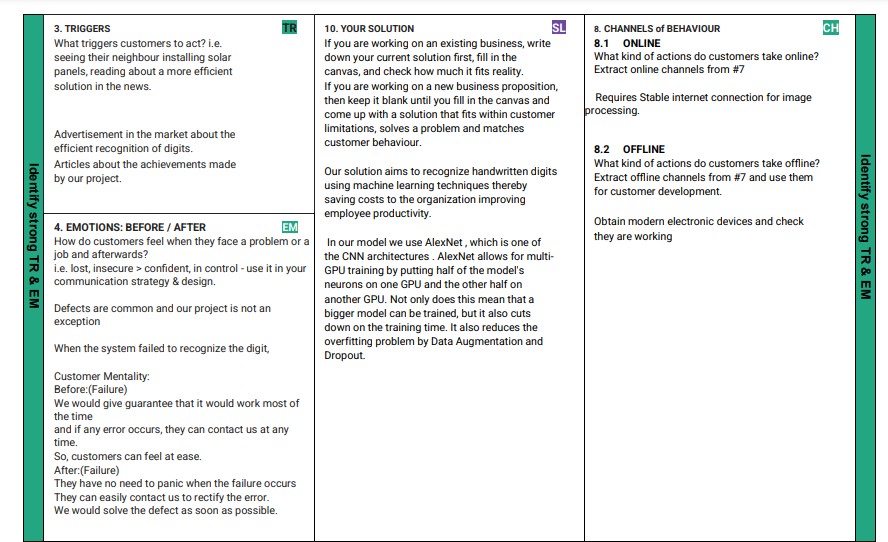
|  |  |  |
| --- | --- | --- |
| **S.No.** | **Parameter** | **Description** |
| 1 | Problem Statement (Problem to be solved) | Handwriting recognition is one of the compelling research works going on because every individual in this world has their own style of writing. It is the capability of the computer to identify and understand handwritten digits or characters automatically. Because of the progress in the field of science and technology, everything is being digitized to reduce human effort. Hence, there comes a need for handwritten digit recognition in many real-time applications. The user interacts with the UI (User Interface) to upload the image as input. The uploaded image is analyzed by the model which is integrated.  Once the model analyses the uploaded image, the prediction  is showcased on the UI. |
| 2 | Idea / Solution description | Convolutional Neural Networks (CNN) has become one of the most appealing approaches and has been an ultimate factor in a variety of recent success and challenging machine learning applications. In our model we use AlexNet , which is one of the CNN architectures .  AlexNet allows for multi-GPU training by putting half of the model's neurons on one GPU  and the other half on another |

|  |  |  |
| --- | --- | --- |
|  |  | GPU. Not only does this mean that a bigger model can be trained, but it also cuts down on the training time. It also reduces the overfitting problem by Data  Augmentation and Dropout. |
| 3 | Novelty / Uniqueness | Handwritten Digit Recognition is the capability of a computer to fete the mortal handwritten integers from different sources like images, papers,touch defenses, etc. And classify them into 10 predefined classes(0- 9).This is the existing method along with this we add some features to make our project  unique among them. |
| 4 | Social Impact / Customer Satisfaction | Even the unclear or blurred digits can be recognized after the removal of noise and data preprocessing .One such application is a handwritten digit recognition system that can be used in postal mail sorting, bank check processing,  form data entry, etc., |
| 5 | Business Model (Revenue Model) | Handwritten digit recognition is necessary because everything is digitalized. The benefits of handwritten digit recognizer is high. In the banking sector, it is very efficient. It is used to recognize the figures written on cheques.So, Varied handwriting of each and every person in the cheque can be identified.  Handwritten addresses are difficult to sort by machine, not |

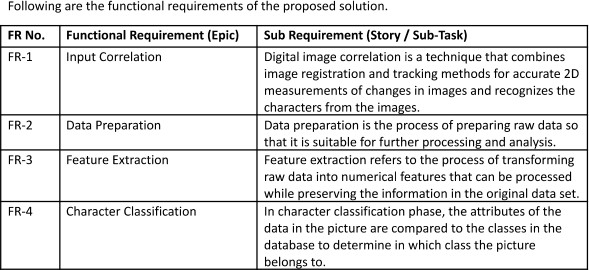
|  |  |  |
| --- | --- | --- |
|  |  | necessarily because of sloppy handwriting, but because people write all over the envelope.We have hard time segmenting handwritten addresses into their components, such as ZIP code or street address, because very few people print addresses neatly in a prescribed format.  So, this problem can be solved using Handwritten digit  recognition system. |
| 6 | Scalability of the Solution | In our model, AlexNet significantly outperformed as it is trained on a GTX 580 GPU with only 3 GB of memory which couldn’t fit the entire network. So the network was split across 2 GPUs, with half of the neurons(feature maps) on each GPU. So, a greater accuracy can be attained by allowing multi-GPU training by putting half of the model's neurons on one GPU and the  other half on another GPU. |

* 1. **Problem Solution fit**

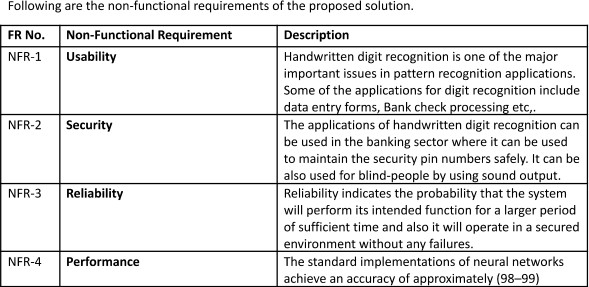


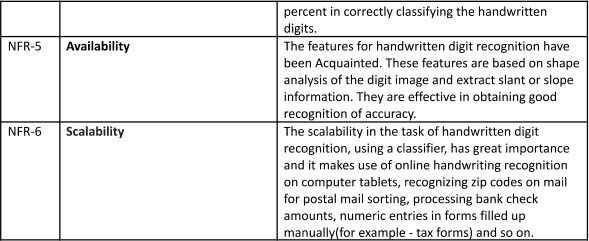


1. **REQUIREMENT ANALYSIS**
   1. **Functional requirement**

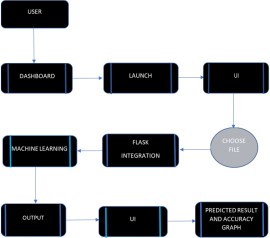


* 1. **Non-Functional requirements**

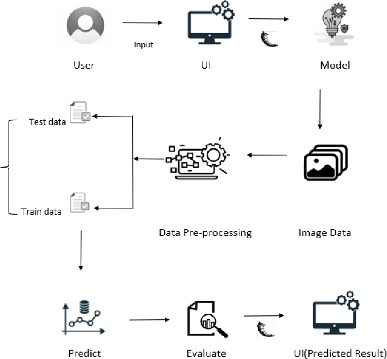




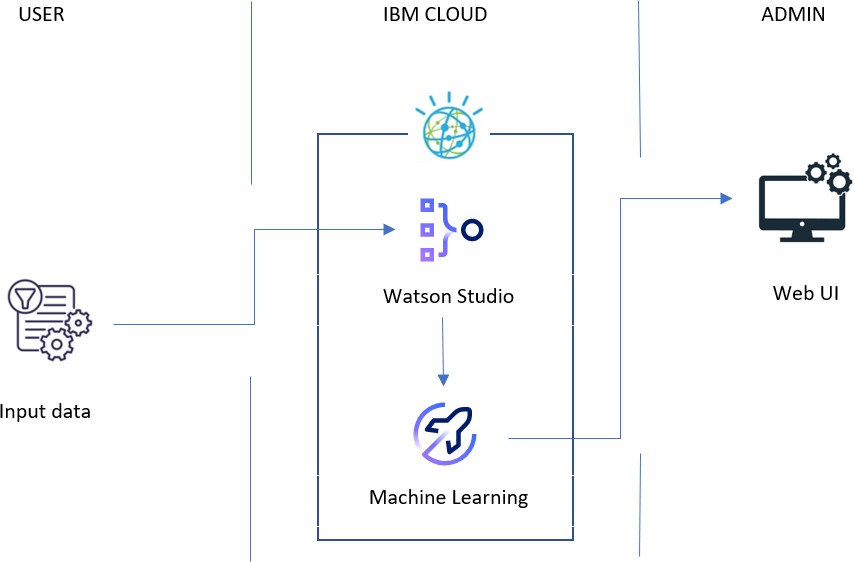
1. **PROJECT DESIGN 5.1Data Flow Diagrams**

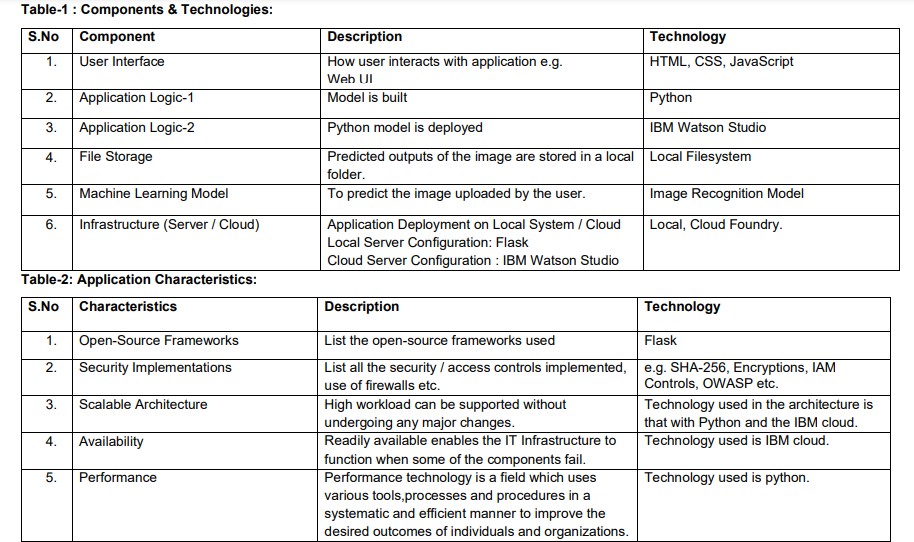


* 1. **Solution & Technical Architecture Solution Architecture**

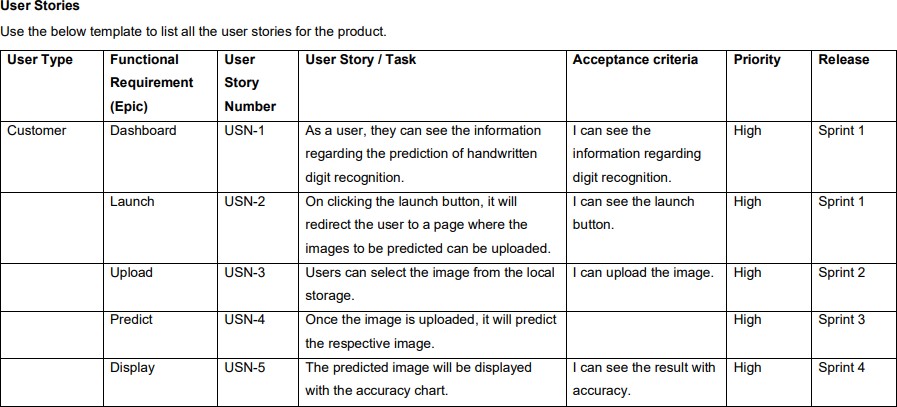


**Technology Architecture**

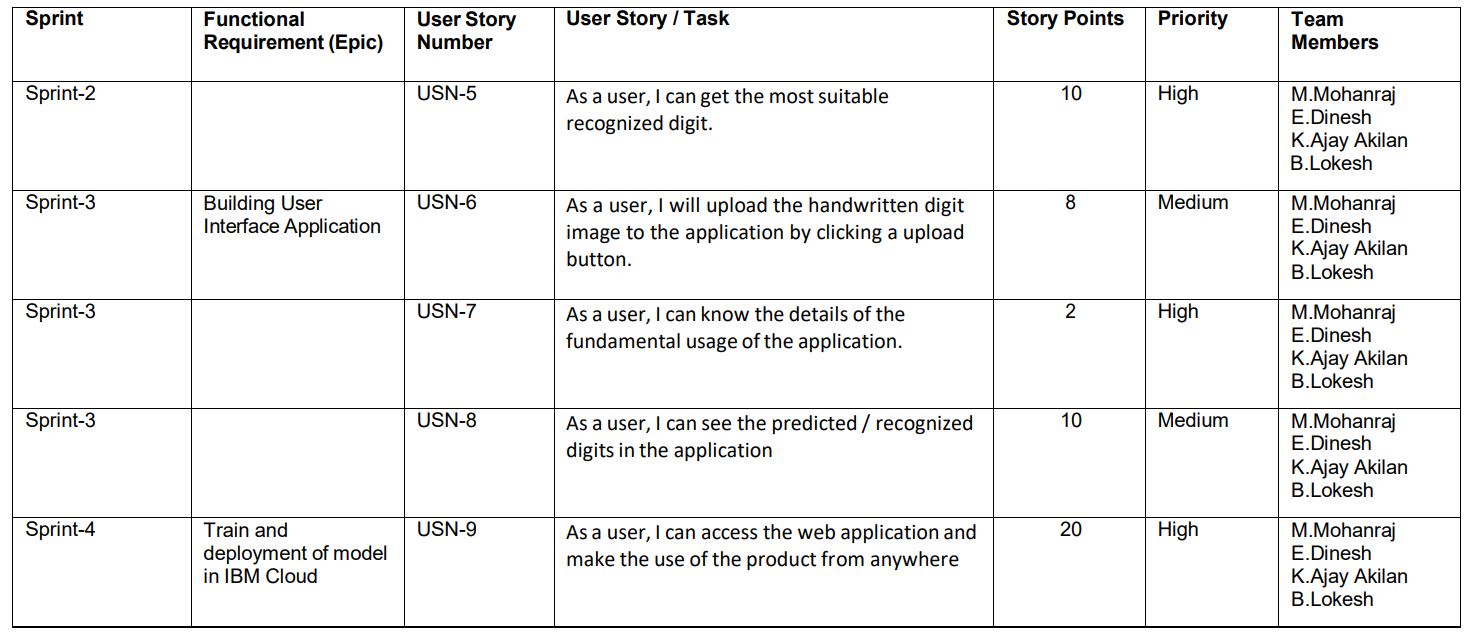
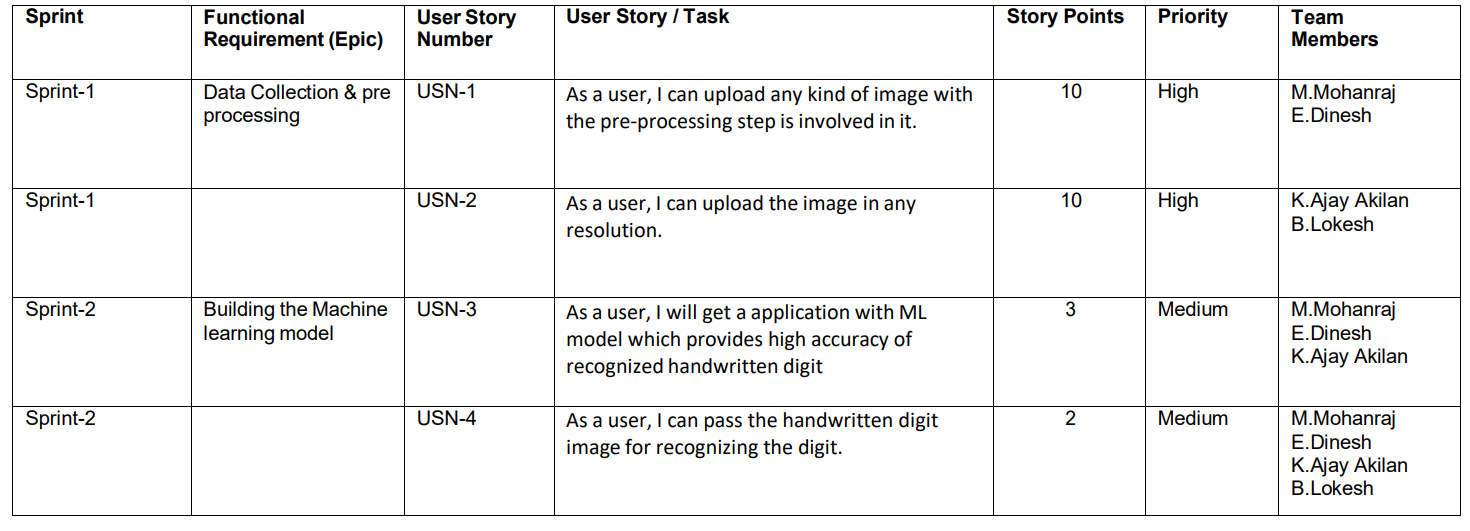




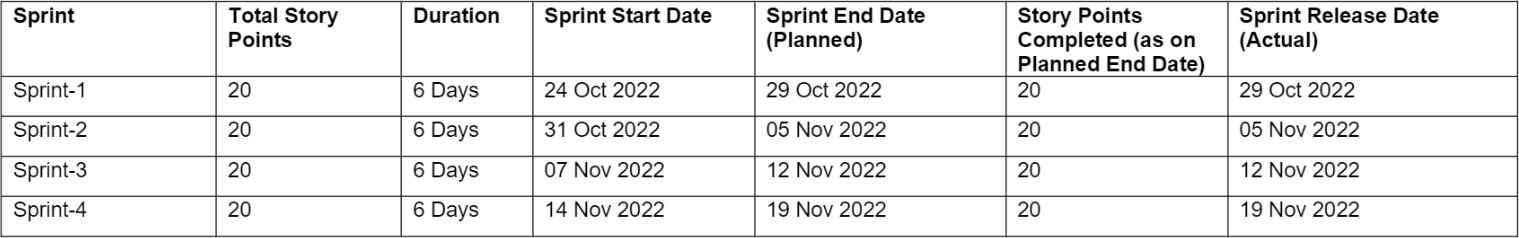
* 1. **User Stories**



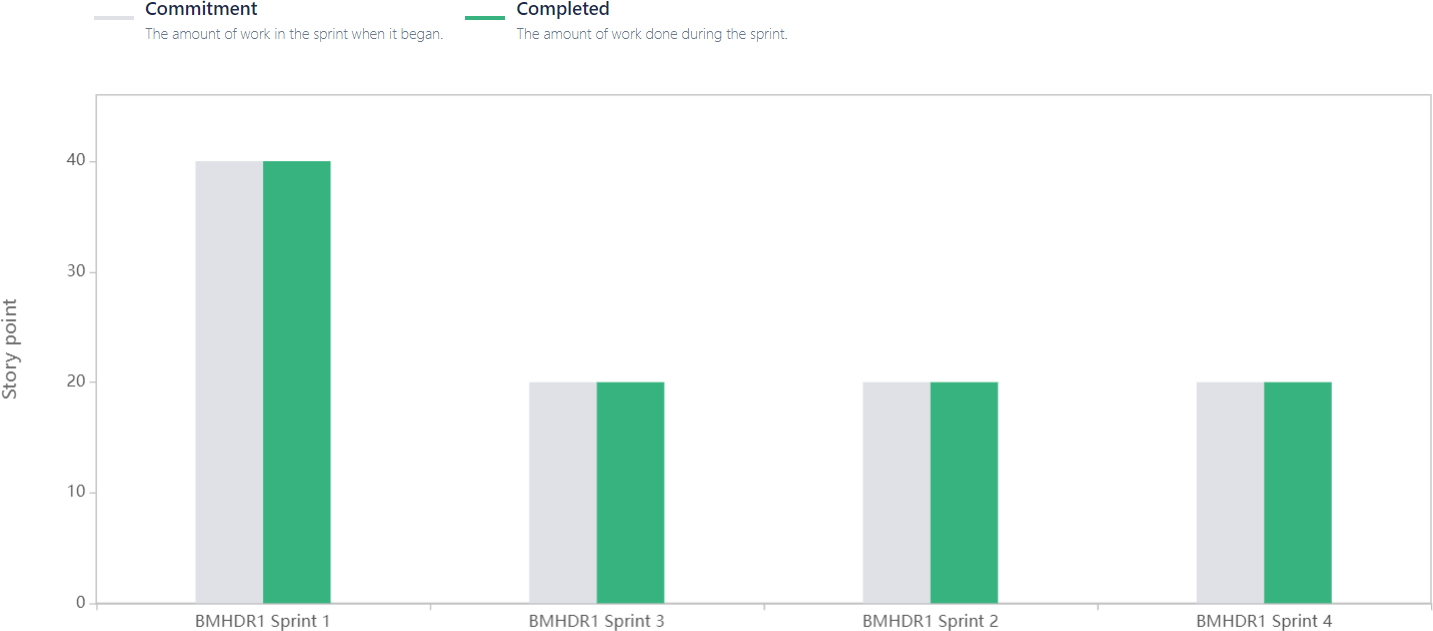
1. **PROJECT PLANNING & SCHEDULING 6.1Sprint Planning & Estimation**

****

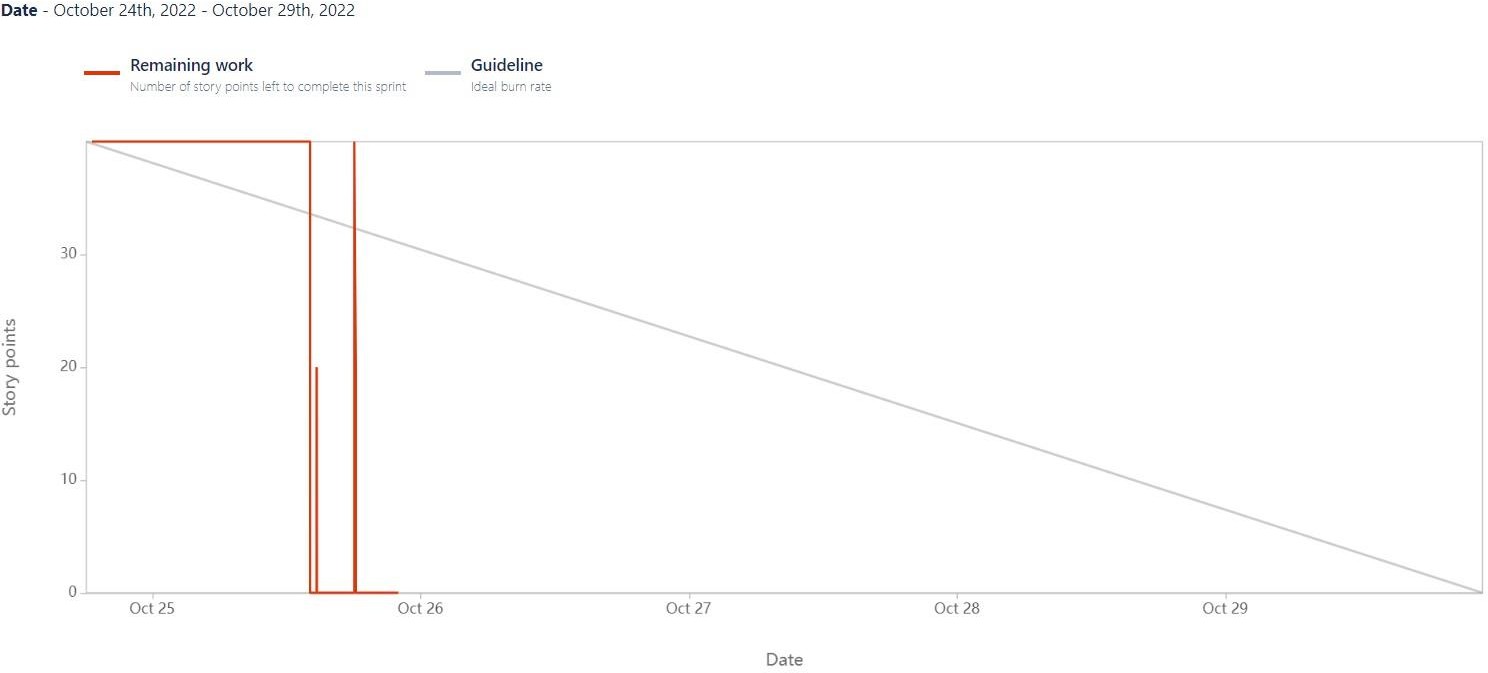
* 1. **Sprint Delivery Schedule**



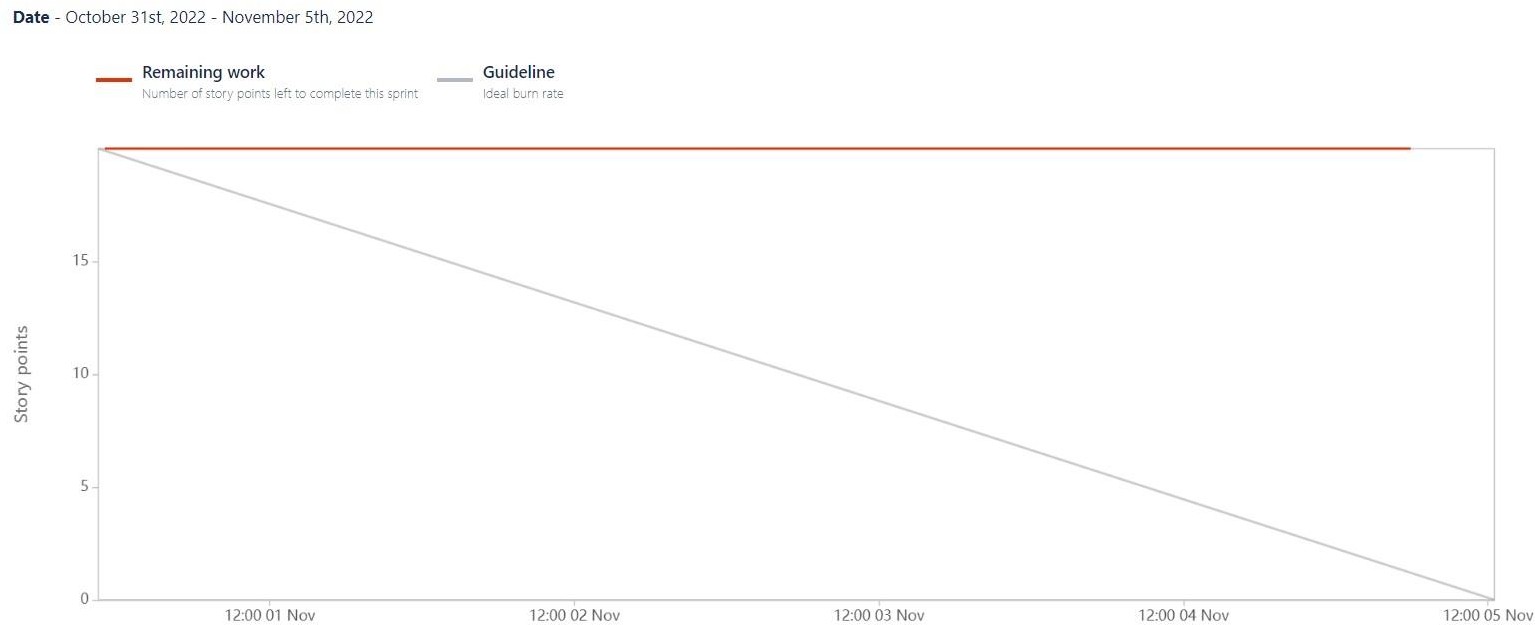
* 1. **Reports from JIRA Velocity Report**



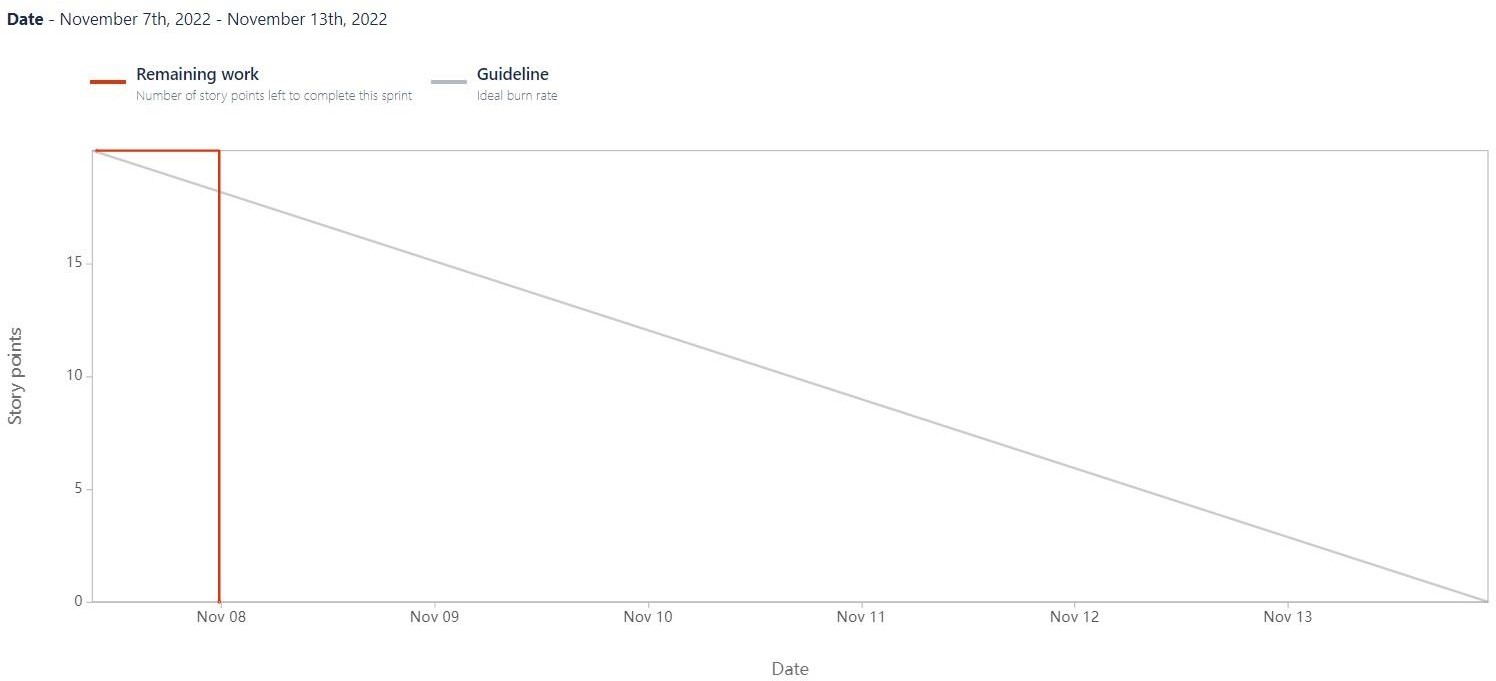
**` Sprint 1**



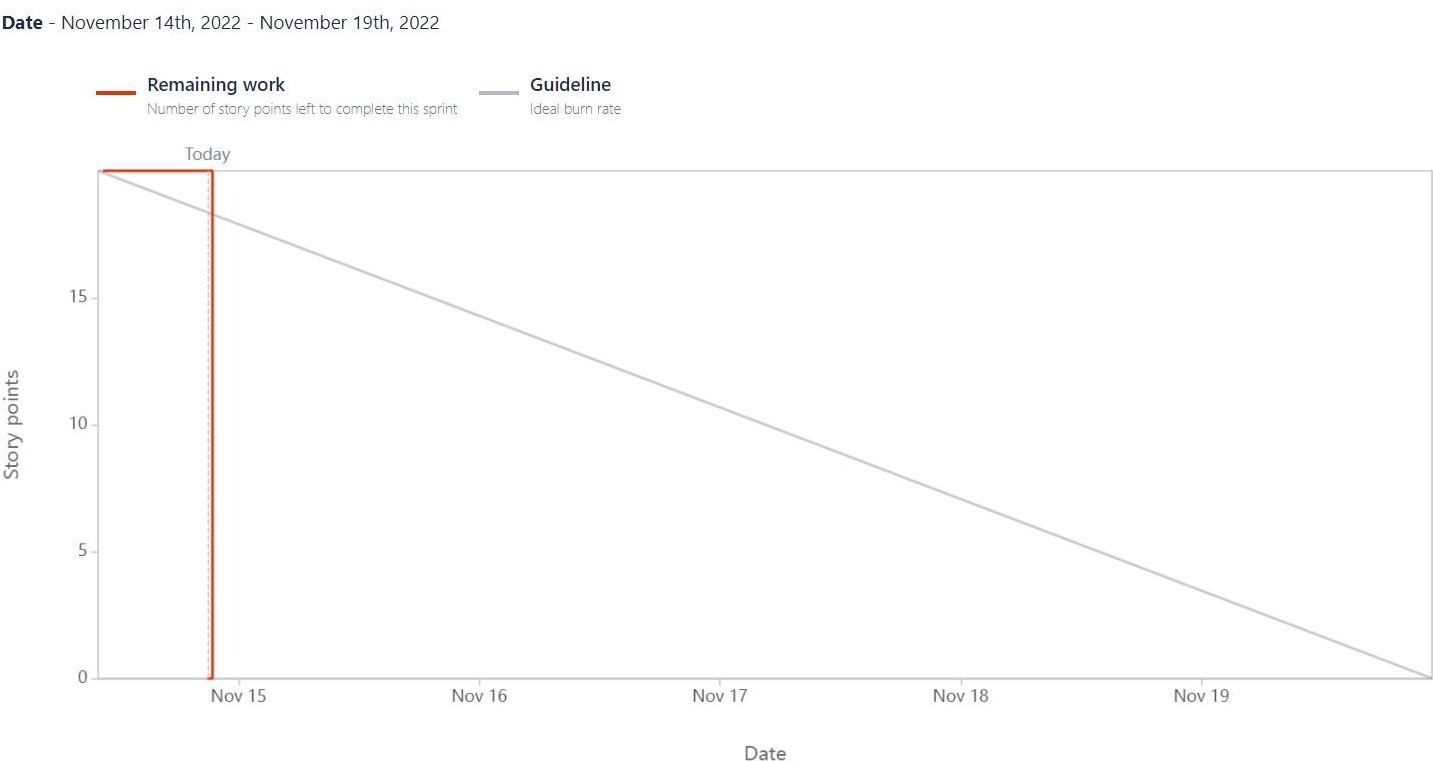
**Sprint 2**



**Sprint 3**



**Sprint 4**



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

import numpy as np import os

from PIL import Image

from flask import Flask, request, render\_template, url\_for from werkzeug.utils import secure\_filename, redirect

#from gevent.pywsgi import WSGIServer from keras.models import load\_model from keras.preprocessing import image from flask import send\_from\_directory

UPLOAD\_FOLDER = 'D:/ibm/data'

app = Flask( name ) app.config['UPLOAD\_FOLDER'] = UPLOAD\_FOLDER

model = load\_model("./DigitRecog\_IBM\_model/mnistCNN.h5")

@app.route('/') def index():

return render\_template('index.html')

@app.route('/predict', methods=['GET', 'POST']) def upload():

if request.method == "POST": f = request.files["image"]

filepath = secure\_filename(f.filename) f.save(os.path.join(app.config['UPLOAD\_FOLDER'], filepath))

upload\_img = os.path.join(UPLOAD\_FOLDER, filepath)

img = Image.open(upload\_img).convert("L") # convert image to monochrome img = img.resize((28, 28)) # resizing of input image

im2arr = np.array(img) # converting to image

im2arr = im2arr.reshape(1, 28, 28, 1) # reshaping according to our requirement pred = model.predict(im2arr)

num = np.argmax(pred, axis=1) # printing our Labels return render\_template('predict.html', num=str(num[0]))

if name == ' main ': app.run(debug=True, threaded=False)

1. **TESTING**
   1. **Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test case ID** | **Feature Type** | **Component** | **Test Scenario** | **Expected Result** | **Actual Result** | **Status** |
| Homepage\_TC\_OO1 | Functional | Home Page | Verify user is able to see the Homepage when clicked  on the link | Home Page should be displayed. | Working as expected | Pass |
| Homepage\_TC\_OO2 | UI | Home Page | Verify the UI elements in Homepage | Application should show below UI elements: a.choose file button b.predict button  c.clear button | Working as expected | Pass |
| Homepage\_TC\_OO3 | Functional | Home Page | Verify user is able to choose file from the local system and click on predict | Choose file popup screen must be displayed and user should be able to click on  predict button | Working as expected | Pass |
| Homepage\_TC\_OO4 | Functional | Home page | Verify user able to select invalid file format | Application won't allow to attach formats other than ".png, .jiff,  .pjp, .jpeg,  .jpg, .pjpeg" | Working as expected | Pass |
| Predict\_TC\_OO5 | Functional | Predict page | Verify user is able to navigate to the predict to and view the  predicted result | User must be navigated to the predict page and must view the  predicted result | Working as expected | Pass |

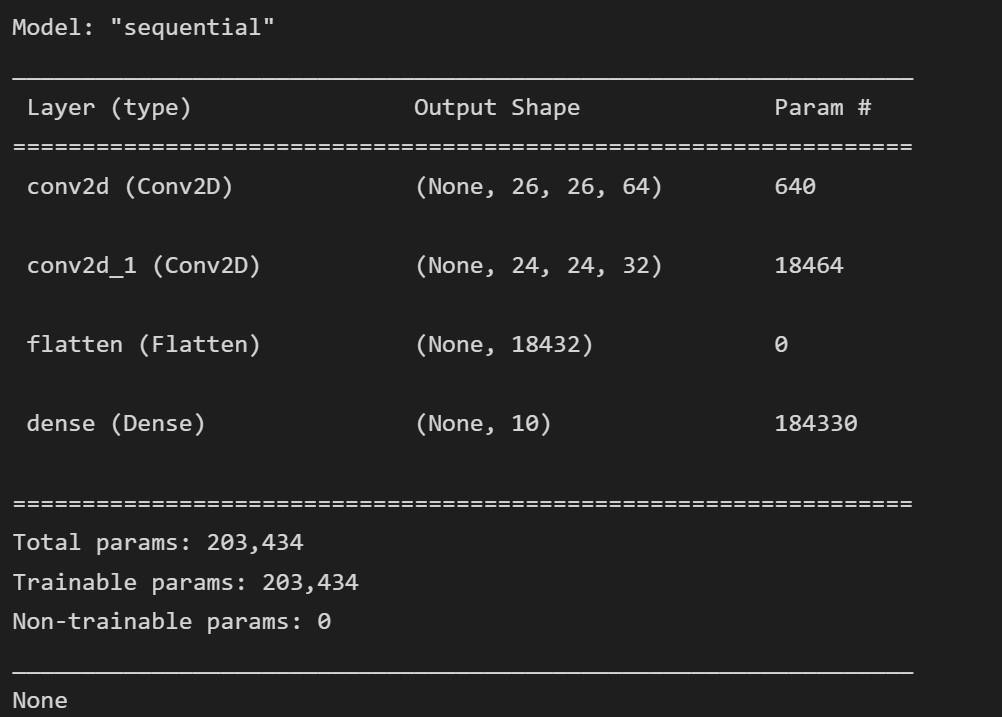
* 1. **User Acceptance Testing Defect Analysis**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Resolution** | **Severity 1** | **Severity 2** | **Severity 3** | **Severity 4** | **Subtotal** |
| By Design | 0 | 0 | 0 | 0 | 0 |
| Duplicate | 0 | 0 | 0 | 0 | 0 |
| External | 0 | 0 | 0 | 0 | 0 |
| Fixed | 0 | 0 | 0 | 0 | 0 |
| Not Reproduced | 0 | 0 | 0 | 0 | 0 |
| Skipped | 0 | 0 | 0 | 0 | 0 |
| Won't Fix | 0 | 0 | 0 | 0 | 0 |
| Totals | 0 | 0 | 0 | 0 | 0 |

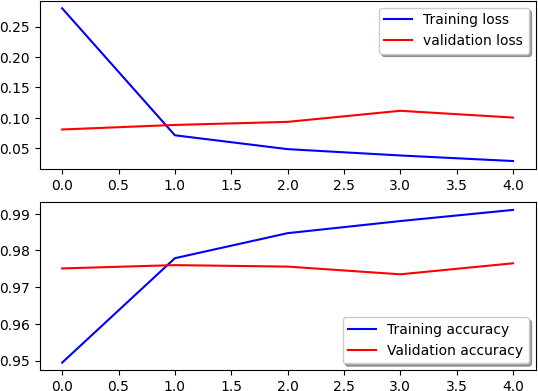
**Test Case Analysis**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Section** | **Total Cases** | **Not Tested** | **Fail** | **Pass** |
| Client Application | 5 | 0 | 0 | 5 |
| Security | 5 | 0 | 0 | 5 |
| Final Report Output | 5 | 0 | 0 | 5 |
| Performance | 5 | 0 | 0 | 5 |

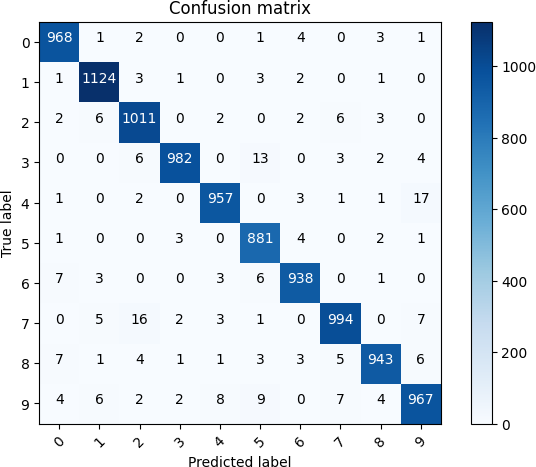
1. **RESULTS**
   1. **Performance Metrics Model Summary**:



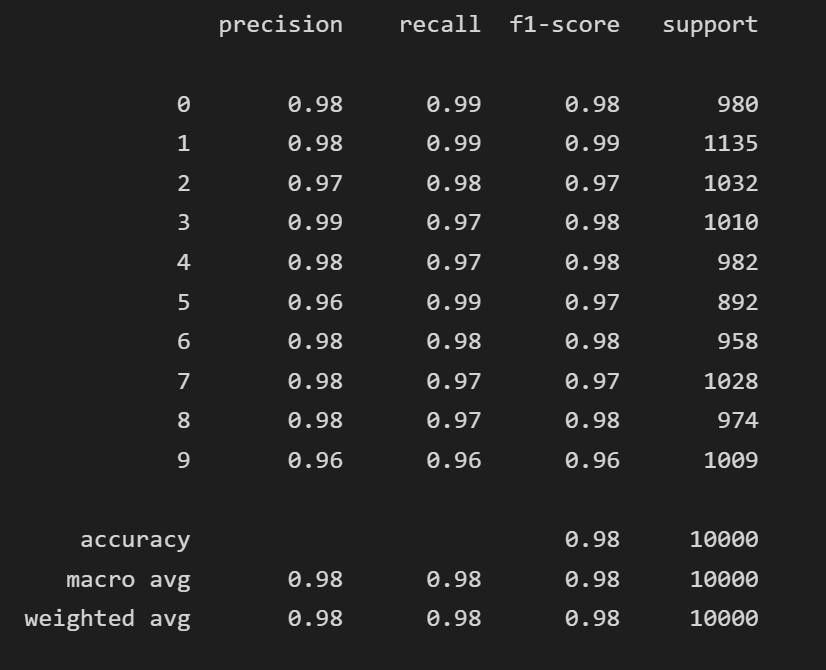
**Accuracy:**



**Confusion Matrix:**

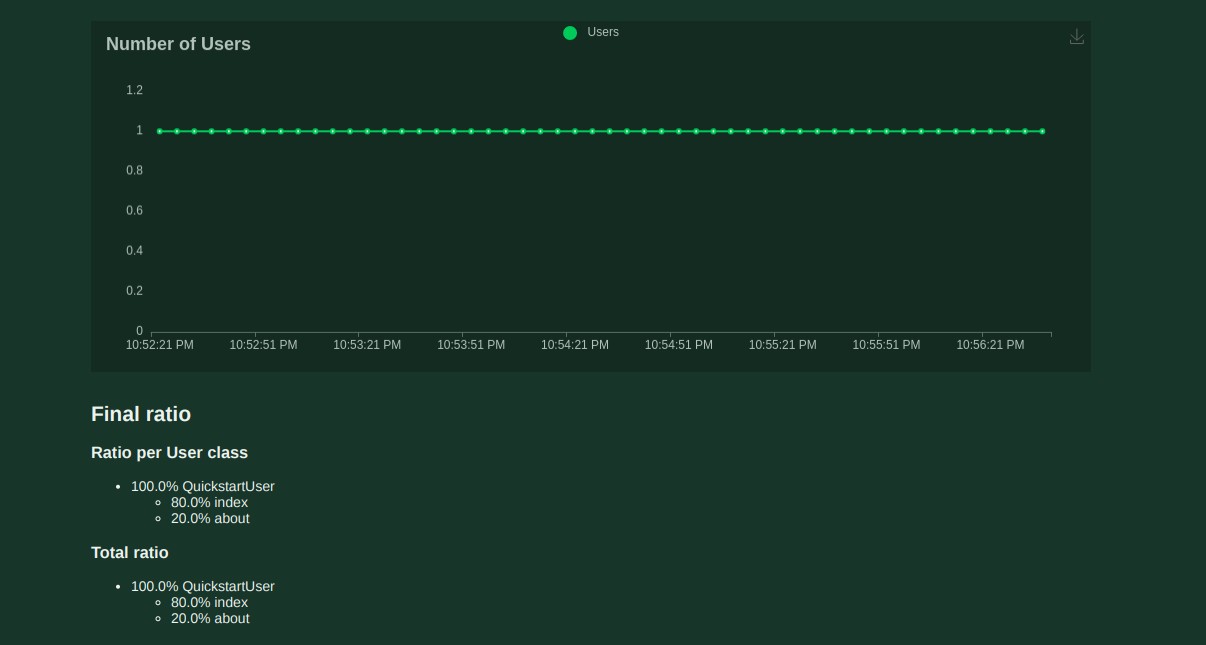


**Classification Report:**



**Performance Metrics Result:**





1. **ADVANTAGES & DISADVANTAGES**

**Advantages**

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

**Disadvantages**

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

1. **CONCLUSION**

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

1. **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.

1. **APPENDIX**

**Source Code HTML AND CSS:**

**index.html:**

<html>

<head>

<title>HDR</title>

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

<link href=["https://fonts.googleapis.com/css2?family=Prompt:wght@600&display=swap"](https://fonts.googleapis.com/css2?family=Prompt%3Awght%40600&display=swap) rel="stylesheet">

<link href="https://fonts.googleapis.com/css2?family=Varela+Round&display=swap" rel="stylesheet">

<link href=["https://fonts.go](https://fonts.googleapis.com/css2?family=Source%2BCode%2BPro%3Awght%40500&display=swap)o[gleapis.com/css2?family=Source+Code+Pro:wght@500&display=swap](https://fonts.googleapis.com/css2?family=Source%2BCode%2BPro%3Awght%40500&display=swap)" rel="stylesheet">

<link href="https://fonts.googleapis.com/css?family=Calistoga|Josefin+Sans:400,700|Pacifico&display

=swap" rel="stylesheet">

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">

<link rel="stylesheet" type= "text/css" href= "{{ url\_for('static',filename='css/style.css') }}">

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

<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="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](https://cdn.jsdelivr.net/npm/%40tensorflow/tfjs%40latest)"></script>

<link rel="stylesheet" href=["https://cdn.jsdelivr.net/npm/bootstrap@4.6.1/dist/css/bootstrap.min.cs](https://cdn.jsdelivr.net/npm/bootstrap%404.6.1/dist/css/bootstrap.min.css)s">

<script src="[https://cdn.jsdelivr.net/npm/jquery@3.6.0/dist/jquery.slim.min.js](https://cdn.jsdelivr.net/npm/jquery%403.6.0/dist/jquery.slim.min.js)"></script>

<script src=["https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js"](https://cdn.jsdelivr.net/npm/popper.js%401.16.1/dist/umd/popper.min.js)></script>

<script src=["https://cdn.jsdelivr.net/npm/bootstrap@4.6.1/dist/js/bootstrap.bundle.min.js"](https://cdn.jsdelivr.net/npm/bootstrap%404.6.1/dist/js/bootstrap.bundle.min.js)></script>

</head>

<style>

body{

background-image: url('static/images/bc1.jpg'); background-repeat: no-repeat;

background-size: cover;

}

</style>

<script>

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

}

$(document).ready(function() {

$('#clear\_button').on('click', function() {

$('#image').val('');

$('#frame').attr('src',"");

});

});

</script>

<body>

<h1>HandWritten Digit Recognition System</h1>

<div class="container p-3 my-3 bg-dark text-white">

<p>Handwritten Digit Recognition is a technology that is much needed in this world as of Today.This Digit Recognition System is used to recognize the digits from different sources like email, posts, cheque etc. Before proper implementation of this technology we have relied on writing text with our own hands which can result in error.It's difficult to store and access physical data with efficiency.The project presents in representing the recognization of handwritten digits (0 - 9) from the famous MNIST dataset. Here we will be using Convolutional Neural Network for the prediction.</p>

</div>

<section id="content">

<div class="leftside">

<form action="/predict" method="POST" enctype="multipart/form-data">

<label>Select a image:</label>

<input id="image" type="file" name="image" accept="image/png, image/jpeg" onchange="preview()"><br><br>

<img id="frame" width="100px" height="100px"/>

<div class="buttons\_div">

<button type="submit" class="btn btn-light">Predict</button>

<button type="button" class="btn btn-light">&nbsp Clear &nbsp</button>

</div>

</form>

</div>

</section>

</body>

</html>

**Predict.html:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Prediction</title>

</head>

<style>

body{

background-image: url('static/images/bc1.jpg'); background-repeat: no-repeat;

background-size: cover;

}

#rectangle{ width:600px; height:150px;

background-color: #000000; border-radius: 25px; position:absolute;

box-shadow: 0px 0px 10px 5px white; top:25%;

left:50%;

transform:translate(-50%,-50%);

}

#num{

text-align: center; font-size: 30px; margin: 0 auto;

padding: 3% 5%;

padding-top: 8%;

color: white;

}

</style>

<body>

<div id="rectangle">

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

</div>

</body>

</html>

**Style.css**

#clear\_button{ margin-left: 15px; font-weight: bold;

color: rgb(0, 174, 255);

}

#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: rgb(0, 32, 112); background-color: skyblue; padding-top: 1%;

padding-bottom: 1%; font-weight: bold;

font-family: 'Bookman', 'URW Bookman L', serif;

}

#team\_id{

text-align: right;

font-size: 25px; padding-right: 3%;

}

#predict\_button{ margin-right: 15px;

color: rgb(0, 255, 72); font-weight: bold;

}

#prediction\_heading{

font-family: 'Josefin Sans', sans-serif; margin-top: 7.5%;

}

#result{

font-size: 5rem;

}

#title{

padding: 1.5% 15%;

margin: 0 auto; text-align: center;

}

.btn {

font-size: 15px; padding: 10px;

/\* -webkit-appearance: none; \*/ background: #eee;

border: 1px solid #888; margin-top: 20px; margin-bottom: 20px;

}

.buttons\_div{

margin-bottom: 30px; margin-right: 80px;

}

.heading{

font-family:"American Typewriter", serif; font-weight: 700;

font-size: 2rem; display: inline;

}

.leftside{

text-align: center; margin: 0 auto;

margin-top: 2%;

/\* padding-left: 10%; \*/

}

#frame{

margin-right: 10%;

}

.predicted\_answer{ text-align: center; margin: 0 auto;

padding: 3% 5%;

padding-top: 0;

/\* padding-left: 10%; \*/

}

h1{

text-align: center; color: aliceblue;

padding: 100px 50px 65px 100px;

}

@media (min-width: 720px) {

.leftside{

padding-left: 10%;

}

}

**FLASK:**

**app.py:**

import numpy as np import os

from PIL import Image

from flask import Flask, request, render\_template, url\_for from werkzeug.utils import secure\_filename, redirect

#from gevent.pywsgi import WSGIServer from keras.models import load\_model from keras.preprocessing import image from flask import send\_from\_directory

UPLOAD\_FOLDER = 'D:/ibm/data'

app = Flask( name )

app.config['UPLOAD\_FOLDER'] = UPLOAD\_FOLDER

model = load\_model("./DigitRecog\_IBM\_model/mnistCNN.h5")

@app.route('/') def index():

return render\_template('index.html')

@app.route('/predict', methods=['GET', 'POST']) def upload():

if request.method == "POST": f = request.files["image"]

filepath = secure\_filename(f.filename) f.save(os.path.join(app.config['UPLOAD\_FOLDER'], filepath))

upload\_img = os.path.join(UPLOAD\_FOLDER, filepath)

img = Image.open(upload\_img).convert("L") # convert image to monochrome img = img.resize((28, 28)) # resizing of input image

im2arr = np.array(img) # converting to image

im2arr = im2arr.reshape(1, 28, 28, 1) # reshaping according to our requirement pred = model.predict(im2arr)

num = np.argmax(pred, axis=1) # printing our Labels return render\_template('predict.html', num=str(num[0]))

if name == ' main ': app.run(debug=True, threaded=False)

**MODEL CREATION:**

import numpy as np

import tensorflow #open source used for both ML and DL for computation from tensorflow.keras.datasets import mnist #mnist dataset

from tensorflow.keras.models import Sequential #it is a plain stack of layers

from tensorflow.keras import layers #A Layer consists of a tensor- in tensor-out computat ion funct ion

from tensorflow.keras.layers import Dense, Flatten #Dense-Dense Layer is the regular deeply connected r

#faltten -used fot flattening the input or change the dimension from tensorflow.keras.layers import Conv2D #onvoLutiona l Layer from keras.optimizers import Adam #opt imizer

from keras. utils import np\_utils #used for one-hot encoding

import matplotlib.pyplot as plt #used for data visualization (x\_train, y\_train), (x\_test, y\_test)=mnist.load\_data ()

x\_train=x\_train.reshape (60000, 28, 28, 1).astype('float32')

x\_test=x\_test.reshape (10000, 28, 28, 1).astype ('float32') number\_of\_classes = 10 #storing the no of classes in a variable

y\_train = np\_utils.to\_categorical (y\_train, number\_of\_classes) #converts the output in binary format

y\_test = np\_utils.to\_categorical (y\_test, number\_of\_classes) model.add(Conv2D(64, (3, 3), input\_shape=(28, 28, 1), activation='relu'))

model.add(Conv2D(32, (3, 3), activation = 'relu')) model.add(Flatten()) model.add(Dense(number\_of\_classes,activation = 'softmax'))

model.compile(loss= 'categorical\_crossentropy', optimizer="Adam", metrics=['accuracy']) x\_train = np.asarray(x\_train)

y\_train = np.asarray(y\_train)

history = model.fit(x\_train, y\_train, validation\_data=(x\_test, y\_test), epochs=5, batch\_size=32)

from sklearn.model\_selection import train\_test\_split from sklearn.metrics import confusion\_matrix

import itertools

fig, ax = plt.subplots(2,1)

ax[0].plot(history.history['loss'], color='b', label="Training loss") ax[0].plot(history.history['val\_loss'], color='r', label="validation loss",axes =ax[0]) legend = ax[0].legend(loc='best', shadow=True)

ax[1].plot(history.history['accuracy'], color='b', label="Training accuracy") ax[1].plot(history.history['val\_accuracy'], color='r',label="Validation accuracy") legend = ax[1].legend(loc='best', shadow=True)

def plot\_confusion\_matrix(cm, classes,

normalize=False, title='Confusion matrix', cmap=plt.cm.Blues):

"""

This function prints and plots the confusion matrix. Normalization can be applied by setting `normalize=True`. """

plt.imshow(cm, interpolation='nearest', cmap=cmap) plt.title(title)

plt.colorbar()

tick\_marks = np.arange(len(classes)) plt.xticks(tick\_marks, classes, rotation=45) plt.yticks(tick\_marks, classes)

if normalize:

cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]

thresh = cm.max() / 2.

for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])): plt.text(j, i, cm[i, j],

horizontalalignment="center",

color="white" if cm[i, j] > thresh else "black")

plt.tight\_layout() plt.ylabel('True label') plt.xlabel('Predicted label')

# Predict the values from the validation dataset Y\_pred = model.predict(x\_test)

# Convert predictions classes to one hot vectors Y\_pred\_classes = np.argmax(Y\_pred,axis = 1)

# Convert validation observations to one hot vectors Y\_true = np.argmax(y\_test,axis = 1)

# compute the confusion matrix

confusion\_mtx = confusion\_matrix(Y\_true, Y\_pred\_classes) # plot the confusion matrix plot\_confusion\_matrix(confusion\_mtx, classes = range(10))

import sklearn

print(sklearn.metrics.classification\_report(Y\_true, Y\_pred\_classes))

print(model.summary())

# Final evaluation of the model

metrics = model.evaluate(x\_test, y\_test, verbose=0) print("Metrics (Test loss &Test Accuracy) : ") print(metrics)

plt.imshow(x\_test[5100])

import numpy as np print(np.argmax(prediction, axis=1))

np.argmax(y\_test[5100:5101]) #printing the actual labels # Save the model

model.save('models/mnistCNN.h5')

**GitHub & Project Demo Link**

**GitHub**

**https://github.com/IBM-EPBL/IBM-Project-50881-1660927791**

**Demo Video**

**https://www.dropbox.com/s/itbkg05nlxhx5v5/VID-20221117-WA0012.mp4?dl=0v**