A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSYTEM

TEAM ID: PNT2022TMID06256

SUBMITTED BY :

1) D.V.BHARATHI PRASAD - 71051902011

2) PRASHANTH.R - 71051902038

3)THARUNRAJ.T -71051902053

4)THIRUKUMARAN.PK -71051902054

TABLE OF CONTENTS

1) INTRODUCTION :

1.1) Project Overview:

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.

1.2) Purpose:

Humans with the help of their brain can recognize the things that they see. Similarly, deep neural networks are developed for the computers to recognize what they see through the User Interface(UI). Handwritten digit recognition is the ability of a computer to receive and interpret intelligible handwritten digit input from sources such as paper documents, photographs, touch-screens and other devices. The applications of digit recognition includes 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 handwritten digits and which is submitted by users by way of a scanner, tablet, and other digital devices.

2)LITERATURE SURVEY:

2.1)EXISTING 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.

2.2)REFERENCES:

S.NO	Author Name	Paper Title	Journal/ Conference- title	Page No/Volume No	Yearof Publication	Description
1	Savita Ahlawat, Amit Choudhary, Anand Nayyar, Saurabh Singh and Byungun Yoon	Improved Handwritt en Digit Recogniti on Using Convoluti onal Neural Networks (CNN)	IEEE Sensors Journal		2020	In this paper, with the aim of improving the performance of handwritten digit recognition,the y valuated variants of a convolutional neural network to avoid complex preprocessin g, costly feature extraction and a complex ensemble (classifier combination) approach of a traditional recognition system

2	Vijayalaxmi R Rudraswamimath, Bhavanish ankar and Channasa ndra.	Handwritt en Digit Recogniti on using CNN	Internati onal Journal of Innovative Science and ResearchTechnol ogy	Volume-4 Issue-6	2019	In this paper, the most widely used Machine learning algorithms, KNN, SVM, RFC and CNN have been trained and tested on the same data in order acquire the comparison between the classifiers
3	Akanksha Gupta, Ravindra Pratap Narwaria and Madhav Singh.	Review on Deep Learning Handwritt en Digit Recogniti on using Convoluti onal Neural Network	Internati onal Journal of Recent Technol ogy and Engine ering (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.

2.3)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 realtime 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 to UI(User Interface).

3.IDEATION & PROPOSED SOLUTION:

3.1)EMPATHY MAP CANVAS:

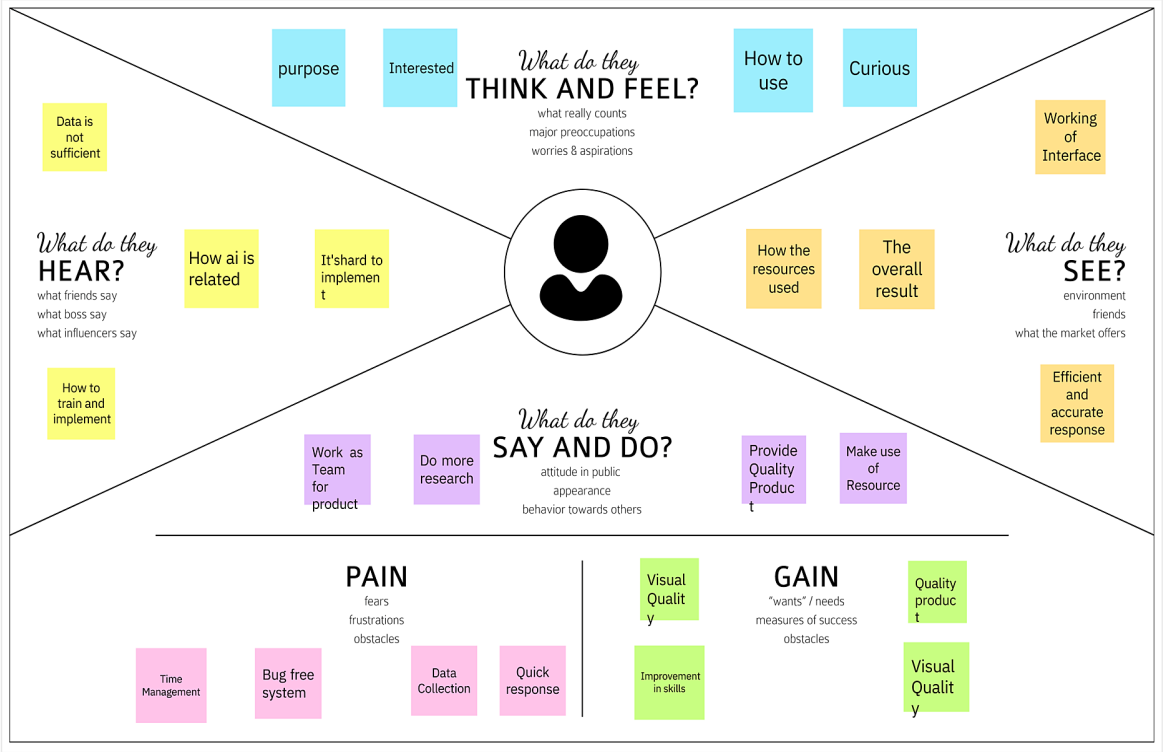
Edit this template
Right-click to unlock

Empathy Map Canvas

Gain insight and understanding on solving customer problems.

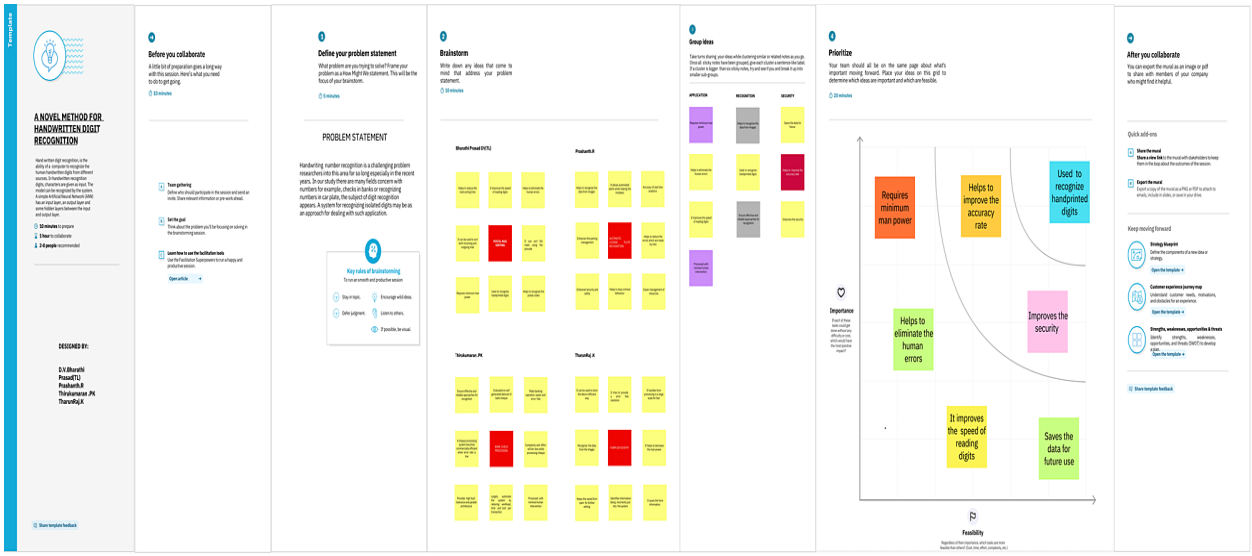
1

Build empathy and keep your focus on the user by putting yourself in their shoes.



3.2) IDEATION & BRAINSTORMING:

Brainstorming:

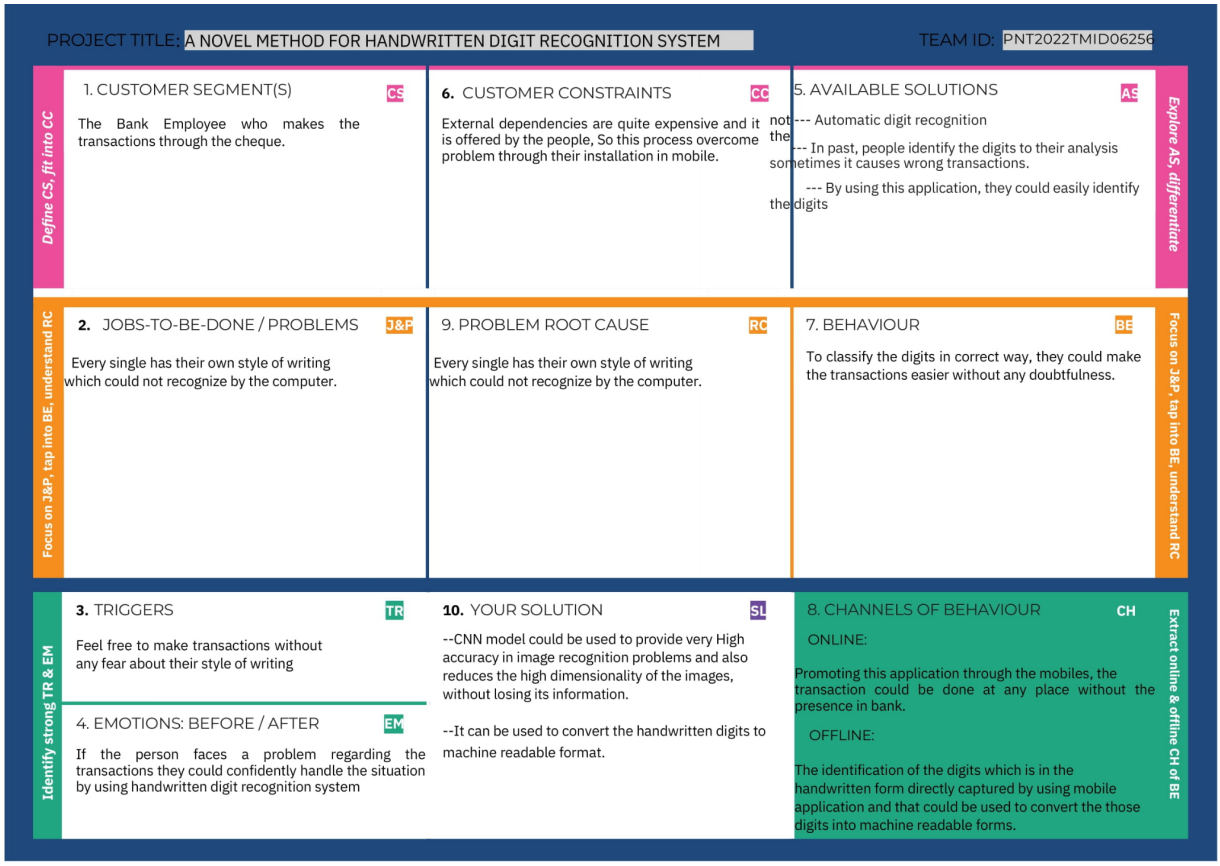


3.3) Proposed Solution:

S.No	Parameter	Description
1.	Problem Statement (Problem to be solved)	To design an application that identifies numbers from written script.
2.	Idea / Solution description	Using the concepts of neural networks, an application will be created. From an image the user gives as input the application will identify the numbers in it and display it.
3.	Novelty / Uniqueness	We will provide the options to give mathematical operations from which they can choose. Accordingly the result will be displayed. Our application will only recognize digits and not all text like OCR

4.	Social Impact / Customer Satisfaction	To correctly and accurately identify handwritten scriptures. This is to ease people when they don't understand something that's written somewhere. For example a prescription given by a doctor
5.	Business Model (Revenue Model)	This application can be used in the world as well in places like i. Traffic surveillance - To make note of the license plate of vehicles. This application can be improved upon to solve complicated mathematical equations.
6.	Scalability of the Solution	It will be a very scalable application as it will be generic and easily adaptable.

3.4) Problem Solution Fit:



4)REQUIREMENT ANALYSIS:

4.1)Functional Requirement:

The functional requirement for the proposed solution are

FR NO.	FUNCTIONAL REQUIREMENT	SUB REQUIREMENT(story/subtask
FR1	The product essentially converts handwritten digits to digital form.	The user is first asked to draw a number on the canvas, and the model that is built is then utilised to compare the data and provide an output in digitalized form.
FR2	Recognizing the handwritten digit and displaying.	Recognizing the handwritten digit and displaying
FR3	Import dataset file directly to the program from a command that will download the dataset from its website. Save the dataset file in the same directory as the program	Installing packages and applications.

FR4	Build a Neural Network with a number of nodes in the input layer equal to the number of pixels in the arrays	NIL
FR5	Activating the Neural Network	Packages – tensorflow

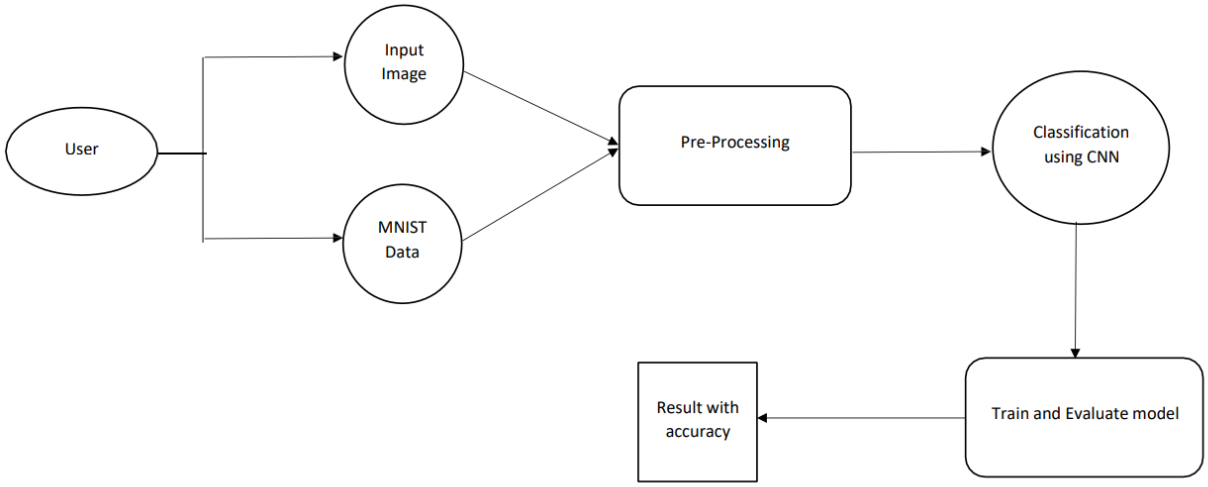
4.2)Non- Functional Requirements:

The non-functional requirements of the proposed solution

FR NO.	NON FUNCTIONAL REQUIREMENT	DESCRIPTION
NFR1	**Usability**	System design should be easily understood and user friendly to users. Furthermore, users of all skill levels of users should be able to navigate it without problems
NFR2	**Security**	The system should automatically be able to authenticate all users with their unique username and password
NFR3	**Performance**	Should reduce the delay in information when hundreds of requests are given.
NFR4	**Availability**	Information is restricted to each users limited access
NF	**Scalability**	the system should be able to handle 10000 users accessing the site at the same time

5)PROJECT DESIGN:

5.1)Data Flow Diagrams:



5.2)Solution & Technical Architecture:

Solution :

S.NO	PARAMETER	DESCRIPTION
1.	Problem Statement (Problem to be solved)	A Novel Method for Handwritten Digit Recognition System
2.	Idea / Solution description	The proposed solution is to classify the digits which is in handwritten format by using CNN based model and this model can be trained by using MNIST database which contains 60,000 training samples and 10,000 test samples.
3.	Novelty / Uniqueness	To classify the image datasets by using CNN, which provides efficient solution compare to other methods. Here ANN algorithm is used for voice recognition which helps blind people.
4.	Social Impact / Customer Satisfaction	Users no need to use external dependencies or devices to recognize the digits, this process can be done through our mobile phones.

5.	Business Model (Revenue Model)	1. Input module 2. Image processing module 3. Segmentation module 4. Feature extraction module 5. Data set training module 6. Classification module
6.	Scalability of the Solution	The accuracy of the result for the training data set is 99.98%, and 99.40% with 50% noise by using MNIST. Even we can improve this model to achieve the better results by training different types of datasets.

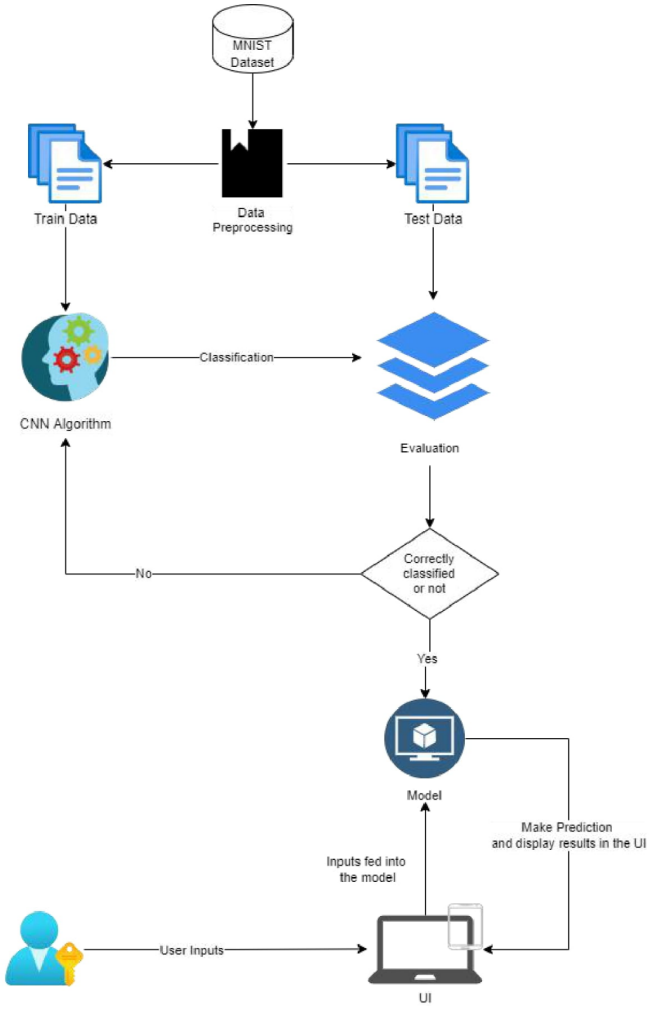
Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

● Find the best tech solution to solve existing business problems.

● Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.

● Define features, development phases, and solution requirements.

● Provide specifications according to which the solution is defined, managed, and delivered



TECHNICAL ARCHITECTURE

5.3)User Stories :

The user stories for the solution:

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer	Application	USN-1	As a user, I can application by opening it easily.	I can download the application	High	Sprint-1
		USN-2	As a user, I will be given access to the canvas board to draw or write the number	I can access the canvas	High	Sprint-1
		USN-3	As a user, I can change the colour of the pen ink.	I can use the canvas pen	Medium	Sprint-2

6)PROJECT PLANNING AND SCHEDULING:

6.1)Sprint Planning & Estimation:

Sprint	Functional Requirement	Task
Sprint-1	Image Data	As a User need to collect the Image Data of Handly Written Images to train the model.
Sprint-2	Dash Board or Website	We using Python Flask Framework to create a dynamic Webpage to host our model (UI).
Sprint-3	Classifier Model	Using CNN Model for Image Classification.

Sprint-4	Cloud	Hosting the Organized appication in Cloud platform.

6.2)Sprint Delivery Schedule:

The backlog and sprint delivery schedule for the solution:

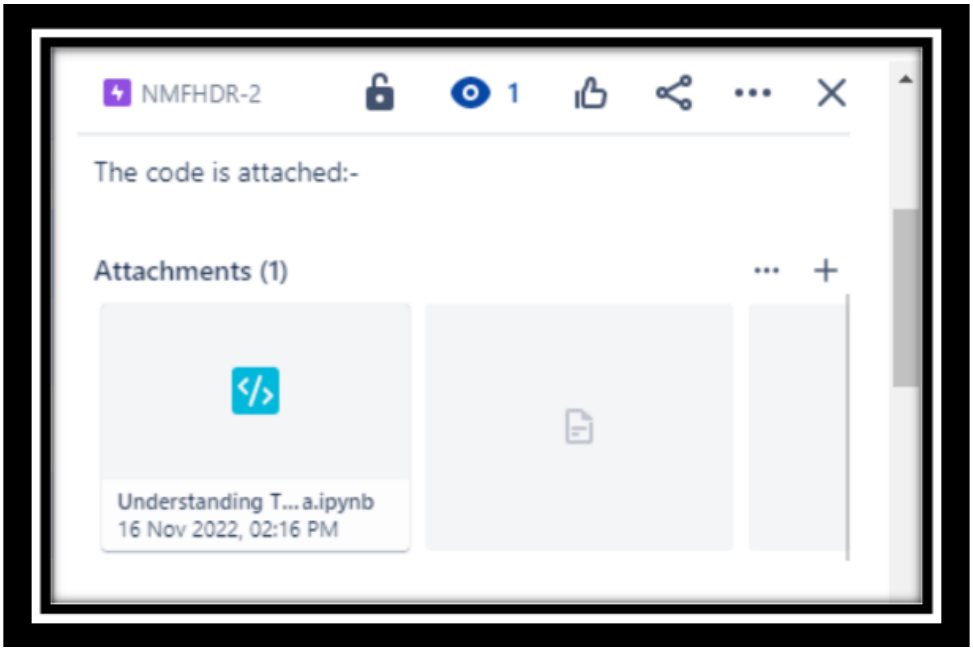
Sprint	**Functional Requirement (Epic)**	**User Story Number**	**User Story / Task**	**Story Points**	**Priority**	**Team Members**
Sprint-1	Data Collection & pre processing	USN-1	As a user, I can upload any kind of image with the pre-processing step is involved in it.	10	High	Tharunraj.T
Sprint-1		USN-2	As a user, I can upload the image in any resolution.	10	High	BharathiPrasad.DV
Sprint-2	Building the Machine learning model	USN-3	As a user, I will get a application with ML model which provides high accuracy of recognized handwritten digit	3	Medium	Thirukumaran.PK

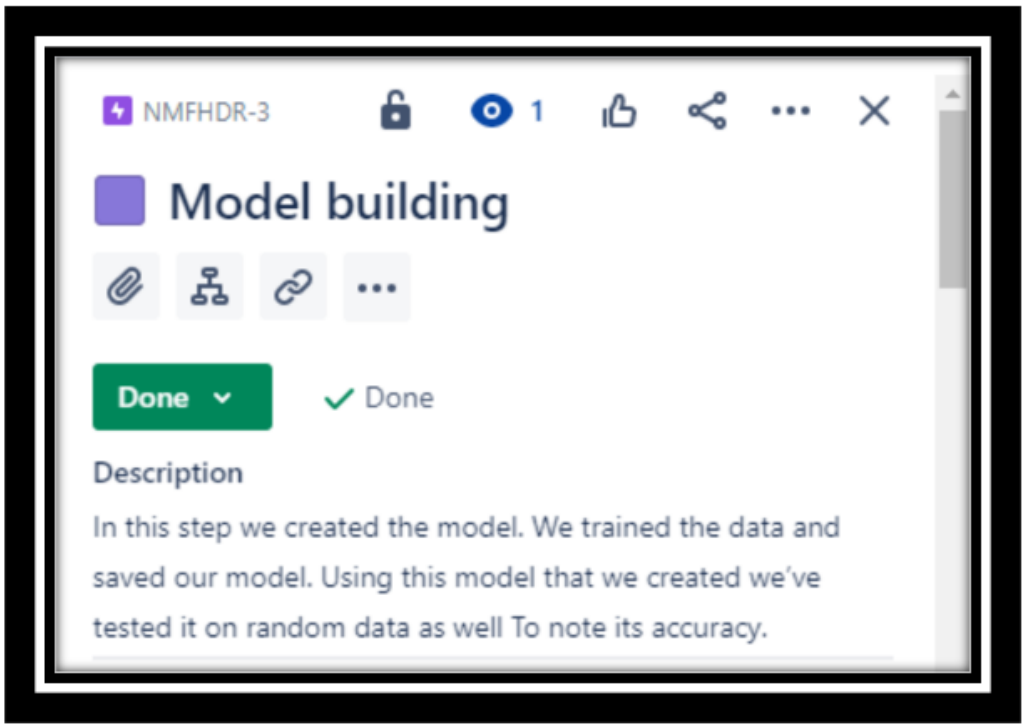
Sprint-2		USN-4	As a user, I can pass the handwritten digit image for recognizing the digit.	2	Medium	Prashanth.R

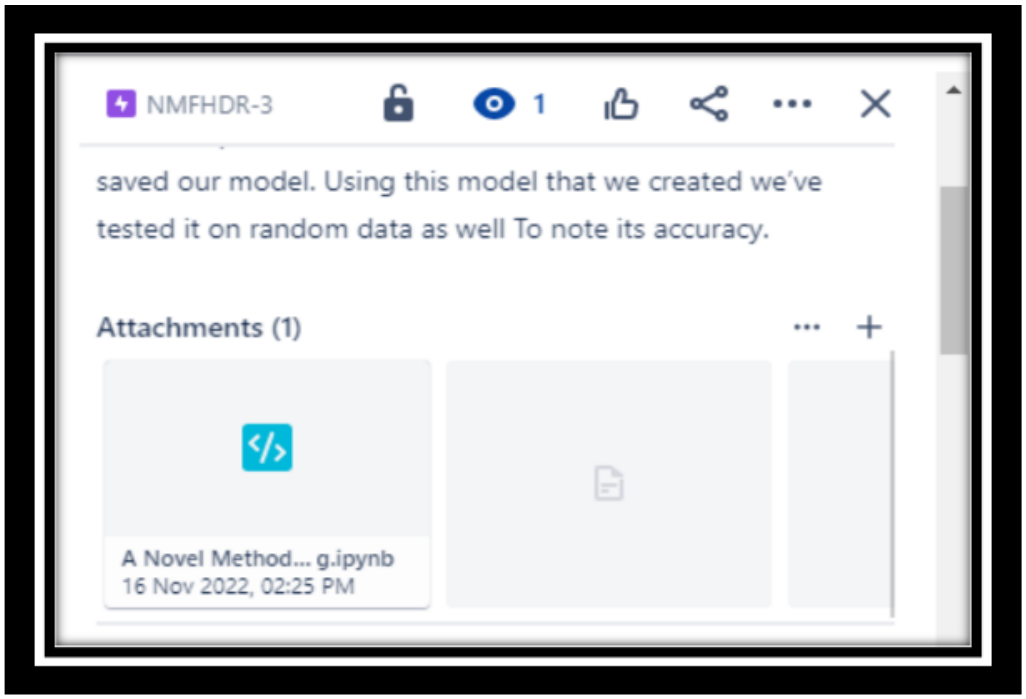
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-2		USN-5	As a user, I can get the most suitable recognized digit.	10	High	Prashanth.R
Sprint-3	Building User Interface Application	USN-6	As a user, I will upload the handwritten digit image to the application by clicking a upload button.	8	Medium	Tharunraj.T
Sprint-3		USN-7	As a user, I can know the details of the fundamental usage of the application	2	High	BharathiPrasad.DV
Sprint-3		USN-8	As a user, I can see the predicted / recognized digits in the application	10	Medium	Prashanth.R

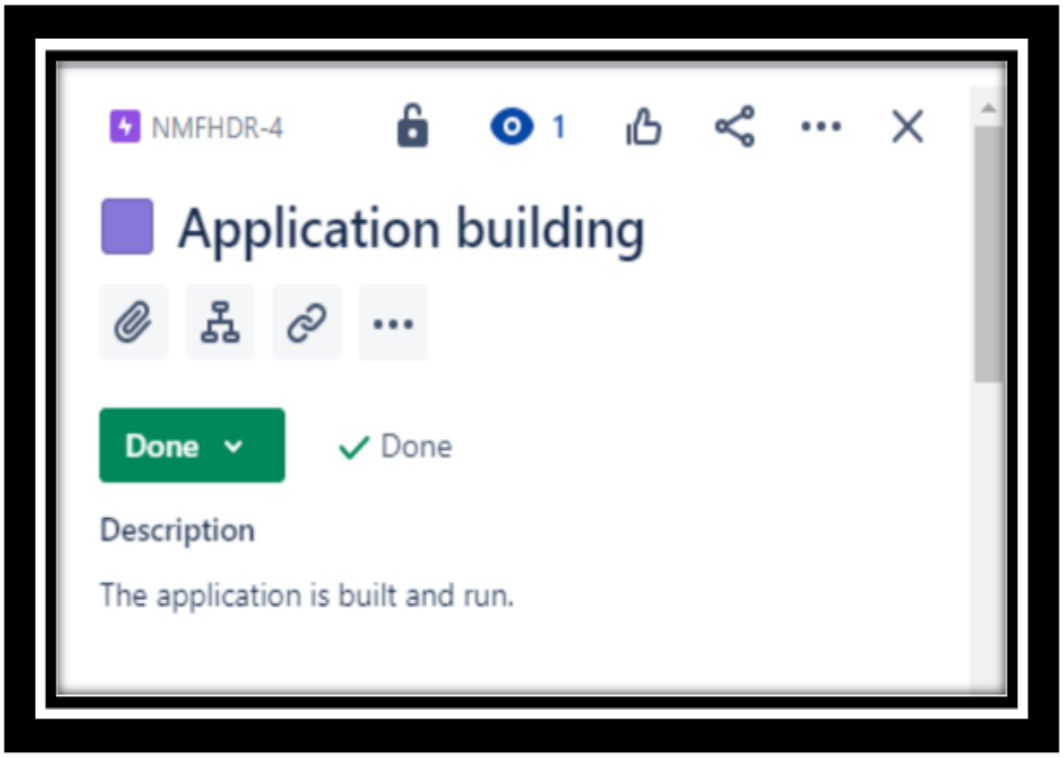
Sprint-4	Train and deployment of model in IBM Cloud	USN-9	As a user, I can access the web application and make the use of the product from anywhere	20	Medium	Tharunraj.T Thirukumaran.PK

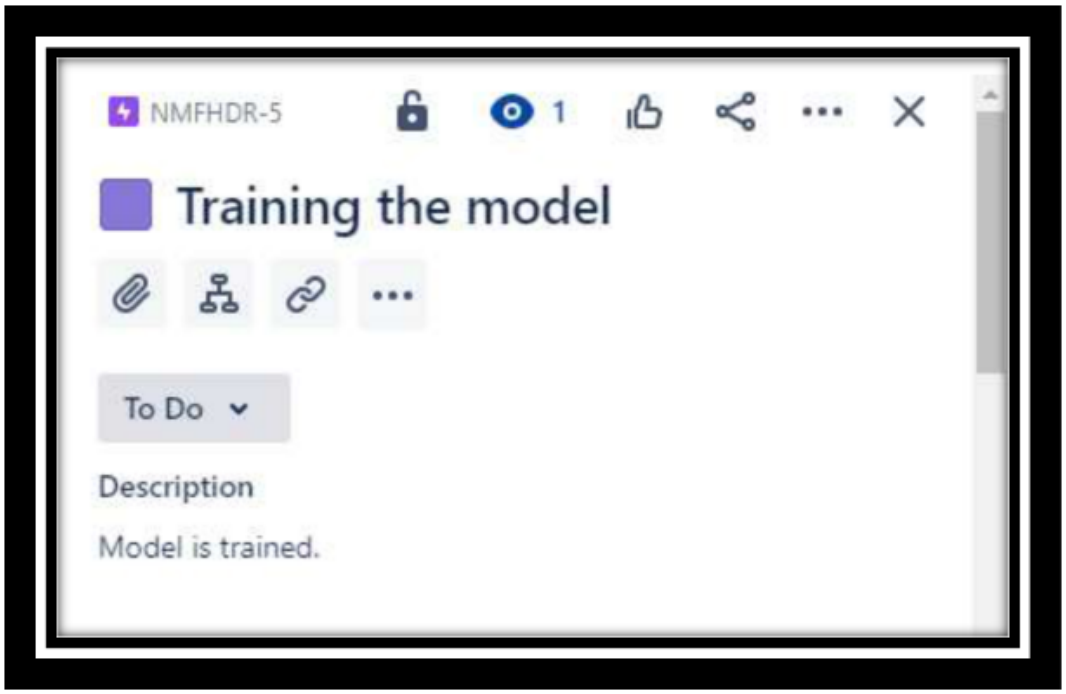
6.3)Reports From JIRA:

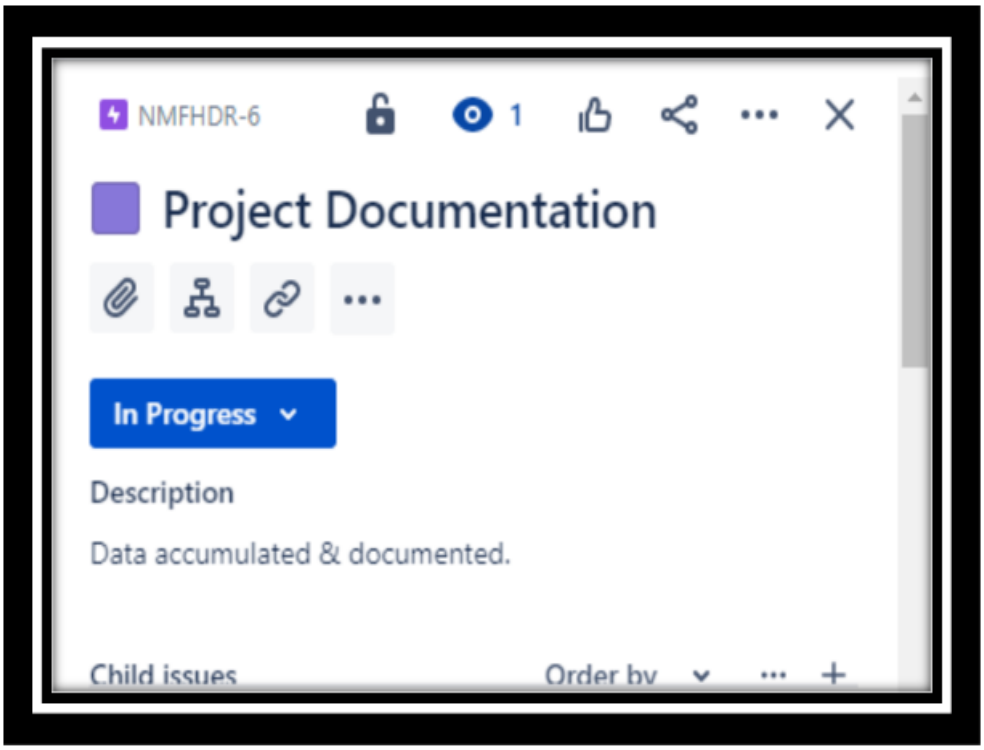


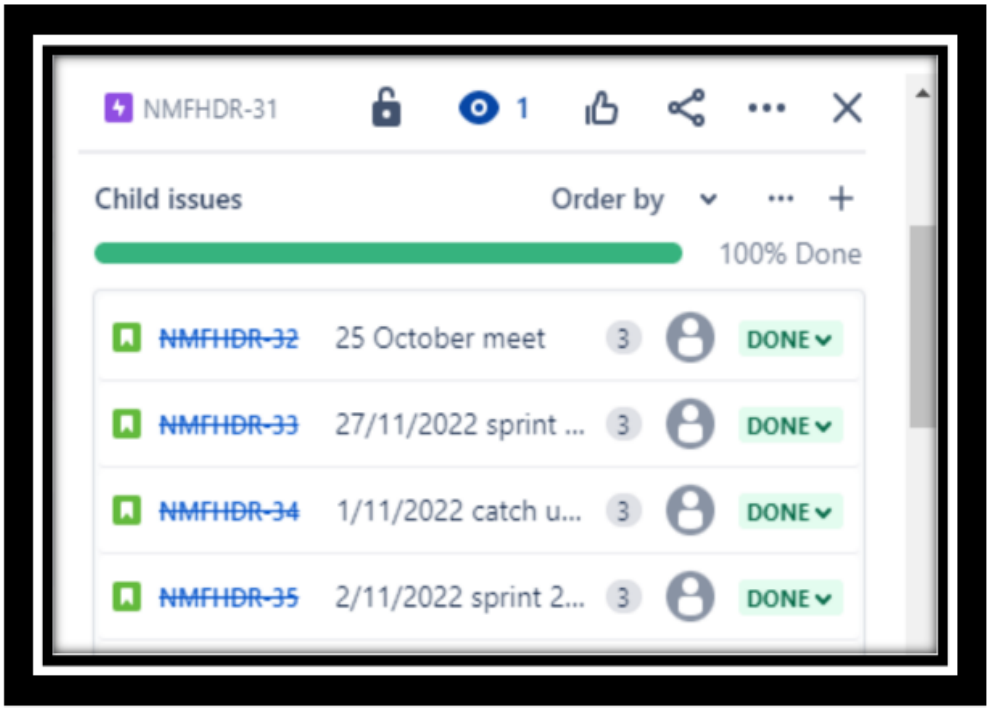

NMFHDR-3
1
Model building
Done
Done
Description
In this step we created the model. We trained the data and saved our model. Using this model that we created we've tested it on random data as well To note its accuracy.

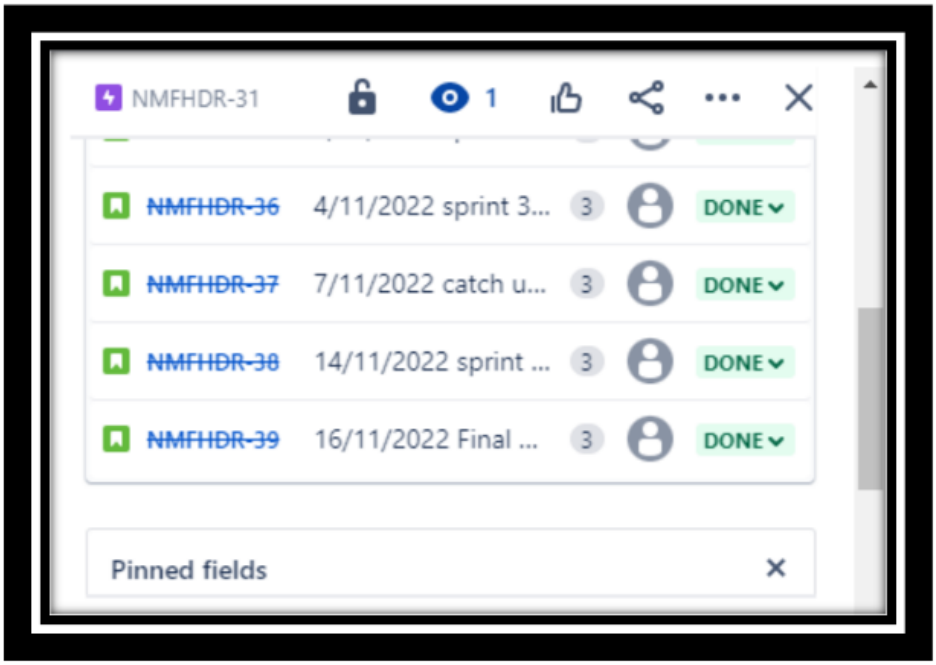

NMFHDR-3
1
saved our model. Using this model that we created we've tested it on random data as well To note its accuracy.
Attachments (1)
A Novel Method... g.ipynb
16 Nov 2022, 02:25 PM


NMFHDR-4
1
Application building
Done
Done
Description
The application is built and run.


NMFHDR-5
1
Training the model
To Do
Description
Model is trained.


NMFHDR-6
1
Project Documentation
In Progress
Description
Data accumulated & documented.
Child issues
Order by


NMFHDR-31
1
Child issues
Order by
100% Done
NMFHDR-32 25 October meet 3 DONE
NMFHDR-33 27/11/2022 sprint ... 3 DONE
NMFHDR-34 1/11/2022 catch u... 3 DONE
NMFHDR-35 2/11/2022 sprint 2... 3 DONE



7)CODING & Solutioning:

7.1)Feature-1:

This is the index or the home page of the web application which describes what this web application is for and how to use this web application to predict the numbers. There is an option where we can choose a file and upload the file and click on the 'Predict' option to predict the number.

There is also one more option 'Clear', next to 'Predict' which can be used to clear the current image and upload a new one.

7.2)Feature-2:

An image of a handwritten digit is uploaded into the web application and then by clicking on predict, it leads to opening of a new web page where the predicted number by the model is displayed.

CODES:

Code for the required features is written in the files "index.html", "predict.html", "style.css" and "app.py"

The "index.html" file renders the home page which gives the information about prediction and the 'predict' and 'clear' options.

"index.html" code:

```
<html>

<head>
  <title>HDR</title>
```

```
  <meta name="viewport" content="width=device-width">
  <link rel="stylesheet" href="../static/css/style.css">
  <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=sw
ap" rel="stylesheet">

  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1
T" 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-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz
0W1" 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>
  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.1/dist/css/bootstrap.min.css">
  <script src="https://cdn.jsdelivr.net/npm/jquery@3.6.0/dist/jquery.slim.min.js"></script>
  <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js"></script>
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.1/dist/js/bootstrap.bundle.min.js"></script>

</head>
```

```
<style>
   body{
    background-image: url('../static/images/1.png');
    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>
   <br><br><br><br><br><br><br>
      <div class="container p-3 my-3 text-dark" id="inst">
         <p>This application will recognise hand written digits in pictures and print the result. The
model used here is Convolutional Neural Networks.
         </p>
         <p>Instructions:
          <ol>
           <li>Choose your file that is image.</li>
            <li>Press the predict button</li>
            <li>The digit will be printed on the next screen!!</li>
          </ol>
         </p>
      </div>
      <section id="content">
```

```
        <div class="leftside">
        <form action="./predict" method="POST" enctype="multipart/form-data">
        <label>Upload Image File: </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" id="bt1">Predict</button>
            <button type="button" class="btn btn-light" id="bt1">  Clear  </button>
          </div>
        </form>
        </div>
    </section>

  </body>

</html>
```

The “style.css” file contains the code for styling the index page and the predict page.

“style.css” code:

```
#clear_button{
  margin-left: 15px;
  font-weight: bold;
  color: rgb(0, 174, 255);
}

#inst{
  color : rgb(172, 167, 167);
  background-color:rgb(250, 214, 250);
  font-family:URW Chancery L, cursive;
  font-weight: bold;

}

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

```css
#content{
  color: rgb(15, 15, 15);
  margin: 0 auto;
  padding: 2% 15%;
  padding-bottom: 0;
  font-family:URW Chancery L, cursive;

}

.welcome{
   text-align: center;
   position: relative;
   color: rgb(253, 253, 253);
   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: rgb(250, 214, 250);
    border: 1px solid #888;
    margin-top: 20px;
    margin-bottom: 20px;
}

.buttons_div{
  margin-bottom: 30px;
  margin-right: 80px;
}
#bt1{
  background-color: rgb(255, 210, 10);
}

#bt1:hover{
  background-color: white;
}
.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:black;
    padding: 100px 50px 65px 100px;
}

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

The app.py contains the backend code and also integrates front end with backend.

“app.py” 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 = r"C:\Users\Bharathi
Prasad\Downloads\IBM--5\IBM-Project-16835-1659623726-main\Final Deliverables\Code\data"

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

model = load_model("./models/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)
```

```
app.debug = True
app.run()
```

8)TESTING :

8.1)Test Cases:

Test caseID	Feature Type	Component	Test Scenario	Expected Result	Actual Result	Status
HP_TC_001	UI	Home Page	Verify UI elements in the Home Page	The Home page must be displayed properly	Working as expected	PASS
HP_TC_002	UI	Home Page	Check if the UI elements are displayed properly in different screen sizes	The Home page must be displayed properly in all sizes	The UI is not displayed properly in screen size 2560 x 1801 and 768 x 630	FAIL
HP_TC_003	Functional	Home Page	Check if user can upload their file	The input imageshould be uploaded to theapplication successfully	Working as expected	PASS
HP_TC_004	Functional	Home Page	Check if user cannot upload unsupported files	The application should not allowuser to select a non image file	User is able to uploadany file	FAIL
HP_TC_005	Functional	Home Page	Check if the page redirectsto the result page once	The page shouldredirect to theresults page	Working as expected	PASS

			theinputis given			
BE_TC_001	Functional	Back end	Check if all theroutes are working properly	All the routes should properly work	Working as expected	PASS
M_TC_001	Functional	Model	Check if the model can handle various image sizes	The model shouldrescale the imageand predict the results	Working as expected	PASS
M_TC_002	Functional	Model	Check if the model predicts thedigit	The model shouldpredict the number	Working as expected	PASS
M_TC_003	Functional	Model	Check if the model can handle complex inputimage	The model shouldpredict the number in the complex image	The model fails to identify the digit since the model is not built to handle such data	FAIL
RP_TC_001	UI	Result Page	Verify UI elements in the Result Page	The Result page must be displayed properly	Working as expected	PASS
RP_TC_002	UI	Result Page	Check if the input image is displayed properly	The input image should be displayed properly	The size of theinput image exceeds the display container	FAIL

RP_TC_003	UI	Result Page	Check if the result is displayed properly	The result shouldbe displayed properly	Working as expected	PASS

8.2)User Acceptance Testing:

Purpose of Document:
The purpose of this document is to briefly explain the test coverage and open issues of the A Novel Method for Handwritten Digit Recognition System project at the time of the release to User Acceptance Testing (UAT).

Defect Analysis:

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

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	2	4	7	10	23
Duplicate	1	0	1	2	4
External	2	3	0	1	6
Fixed	6	2	4	20	32
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	11	14	16	35	76

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	80	0	0	80
Security	2	0	0	2

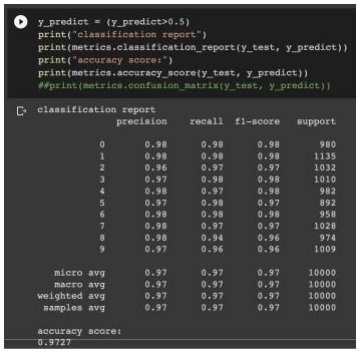
9)RESULTS

9.1) Performance Metrics:

Model Performance Testing:

Project team shall fill the following information in model performance testing template

S.No.	Parameter	Values	Screenshot
1.	Metrics	**Regression Model:** MAE - 0.005652044, MSE - 0.004229254, RMSE - 0.06503271,R2 score0.9529207866394055	

		Classification Model: Confusion Matrix Accuracy Score0.9727 & Classification Report	

10.Advantages & Disadvantages:

Advantages:

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

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

11. Conclusion:

Using Neural Network system, back-propagation learning, to recognize handwritten digits was very successful. An image, which contained 100 samples of each number, was trained and tested. The accuracy rate of recognizing the number was 96%. This accuracy rate is high. It gave different training and testing results every day for each numeral. It will need to take a close look at the system and should look for improvements for the future. From the net-file, the system was able to produce an image-file. This part will also need more improvements. Apart from the above problems and parts that need improvements, the overall recognition system was successful.

12.Future Scope:

The task of handwritten digit recognition using a classifier has a great importance and use such as online hand writing recognition on computer tablet,recognize zipcodes on

mail for postal mail sorting, processing bank check amount, numeric enteries informs filledup by hand and so on.

13.Appendix:

Source Code:

Firstly the model has been trained and saved as “mnistCNN.h5” and is used in the “app.py” file where everything is integrated. Following are the codes of all the files for the project.

The model (saved as mnistCNN.h5):

Importing Libraries

In [1]:
```
import numpy
import tensorflow
from tensorflow.keras.datasets import mnist
from tensorflow.keras.models import Sequential
from tensorflow.keras import layers
from tensorflow.keras.layers import Dense, Flatten
from keras.layers import Convolution2D as Conv2D
from keras.optimizers import Adam
from keras.utils import np_utils
```

Loading Data

In [2]:
```
(X_train,y_train),(X_test,y_test)=mnist.load_data()
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz
11490434/11490434 [==============================] - 1s 0us/step
```

In [ ]:
```
print(X_train.shape)
print(X_test.shape)
```

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

Analyzing the data

[3]:
```
X_train[0]
```

Out[3]:
```
array([[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   3,
         18,  18,  18, 126, 136, 175,  26, 166, 255, 247, 127,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,  30,  36,  94, 154, 170,
        253, 253, 253, 253, 253, 225, 172, 253, 242, 195,  64,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,  49, 238, 253, 253, 253, 253,
        253, 253, 253, 253, 251,  93,  82,  82,  56,  39,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,  18, 219, 253, 253, 253, 253,
        253, 198, 182, 247, 241,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,  80, 156, 107, 253, 253,
        205,  11,   0,  43, 154,   0,   0,   0,   0,   0,   0,   0,   0,
```

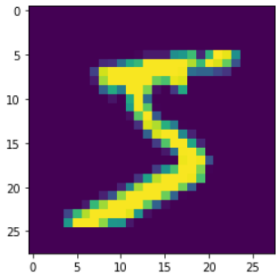
```
       0,   0]], dtype=uint8)
```

In [4]:
```
y_train[0]
```

Out[4]: 5

In [5]:
```
import matplotlib.pyplot as plt
plt.imshow(X_train[0])
```

Out[5]: <matplotlib.image.AxesImage at 0x7f34a90fb3d0>



#Reshaping the data

In [6]:
```
X_train = X_train.reshape(60000,28,28,1).astype('float32')
X_test = X_test.reshape(10000,28,28,1).astype('float32')
```

One-hot Encoding

In [7]:
```
num_of_classes = 10
y_train = np_utils.to_categorical(y_train,num_of_classes)
y_test = np_utils.to_categorical(y_test,num_of_classes)
```

In [8]:
```
y_train[0]
```

Out[8]: array([0., 0., 0., 0., 0., 1., 0., 0., 0., 0.], dtype=float32)

Model Building

Creating Model (Add CNN Layers)

In [9]:
```
model = Sequential()
model.add(Conv2D(64,(3,3),input_shape=(28,28,1),activation='relu'))
model.add(Conv2D(32,(3,3),activation='relu'))
model.add(Flatten())
model.add(Dense(num_of_classes,activation='softmax'))
```

Compiling the model

In [10]:
```
model.compile(loss='categorical_crossentropy',optimizer="Adam",metrics=['accuracy'])
```

Fitting the model

In [11]:
```
model.fit(X_train,y_train,validation_data=(X_test,y_test),epochs=5,batch_size=32)
```

```
Epoch 1/5
1875/1875 [==============================] - 124s 66ms/step - loss: 0.2578 - accuracy: 0.9482 - val_loss: 0.1180 - val_accur
acy: 0.9652
Epoch 2/5
1875/1875 [==============================] - 124s 66ms/step - loss: 0.0778 - accuracy: 0.9769 - val_loss: 0.0907 - val_accur
acy: 0.9739
Epoch 3/5
1875/1875 [==============================] - 126s 67ms/step - loss: 0.0549 - accuracy: 0.9829 - val_loss: 0.0921 - val_accur
acy: 0.9734
Epoch 4/5
1875/1875 [==============================] - 122s 65ms/step - loss: 0.0421 - accuracy: 0.9865 - val_loss: 0.1090 - val_accur
acy: 0.9707
Epoch 5/5
1875/1875 [==============================] - 122s 65ms/step - loss: 0.0315 - accuracy: 0.9898 - val_loss: 0.1161 - val_accur
acy: 0.9731
```

Out[11]: <keras.callbacks.History at 0x7f34a4958790>

Observing the metrics

In [28]:
```
project_metrics=model.evaluate(X_test,y_test,verbose=0)
print("Metrics(Loss& Accuracy)")
print(project_metrics)
```

```
Metrics(Loss& Accuracy)
[0.11606404930353165, 0.9731000065803528]
```

In [27]:
```
import sklearn.metrics as metrics
import numpy as np
import matplotlib.pyplot as plt

y_predict = model.predict(X_test)

mae = metrics.mean_absolute_error(y_test, y_predict)
mse = metrics.mean_squared_error(y_test, y_predict)
rmse = np.sqrt(mse) # or mse**(0.5)
r2 = metrics.r2_score(y_test,y_predict)

print("Results of sklearn.metrics:")
print("MAE:",mae)
print("MSE:", mse)
print("RMSE:", rmse)
print("R-Squared:", r2)
```

```
313/313 [==============================] - 5s 17ms/step
Results of sklearn.metrics:
MAE: 0.005652044
MSE: 0.004229254
RMSE: 0.06503271
R-Squared: 0.9529207866394055
```

Predicting the model

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

```
1/1 [==============================] - 0s 66ms/step
[[5.4520988e-12 1.0888041e-21 7.5880036e-13 1.2055289e-11 2.6268963e-19
  3.1448650e-15 1.1666435e-19 1.0000000e+00 2.1508427e-11 2.9282635e-11]
 [4.4279619e-12 1.2889054e-12 1.0000000e+00 2.0362232e-17 5.6934279e-20
  5.3493605e-19 4.8295710e-13 5.0981827e-21 1.0086084e-15 1.9516000e-19]
 [9.2632938e-08 9.9921167e-01 6.2528164e-07 4.8549702e-11 7.8550651e-04
  1.5237079e-07 6.6308399e-09 1.0095750e-09 1.9272488e-06 1.2933888e-12]
 [1.0000000e+00 1.6884626e-16 2.7246841e-10 5.4617701e-15 1.6128288e-14
  1.5063359e-10 5.7584715e-10 9.0131895e-14 7.9959858e-11 2.1980879e-08]]
```

In [ ]:
```
metrics=model.evaluate(X_test,y_test,verbose=0)
print("Metrics(Loss& Accuracy)")
print(metrics)
```

```
Metrics(Loss& Accuracy)
[0.09232578426599503, 0.9797999858856201]
```

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

```
1/1 [==============================] - 0s 48ms/step
[[5.4520988e-12 1.0888041e-21 7.5880036e-13 1.2055289e-11 2.6268963e-19
  3.1448650e-15 1.1666435e-19 1.0000000e+00 2.1508427e-11 2.9282635e-11]
 [4.4279619e-12 1.2889054e-12 1.0000000e+00 2.0362232e-17 5.6934279e-20
  5.3493605e-19 4.8295710e-13 5.0981827e-21 1.0086084e-15 1.9516000e-19]
 [9.2632938e-08 9.9921167e-01 6.2528164e-07 4.8549702e-11 7.8550651e-04
  1.5237079e-07 6.6308399e-09 1.0095750e-09 1.9272488e-06 1.2933888e-12]
 [1.0000000e+00 1.6884626e-16 2.7246841e-10 5.4617701e-15 1.6128288e-14
  1.5063359e-10 5.7584715e-10 9.0131895e-14 7.9959858e-11 2.1980879e-08]]
```

In [ ]:
```
import numpy as np
print(np.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.]]
```

Saving the model

In [ ]:
```
model.save('models/mnistCNN.h5')
```

The “index.html” file renders the home page which gives the information about prediction and the ‘predict’ and ‘clear’ options.

“index.html” code:

```
<html>

<head>
  <title>HDR</title>
```

```
  <meta name="viewport" content="width=device-width">
  <link rel="stylesheet" href="../static/css/style.css">
  <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=sw
ap" rel="stylesheet">

  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1
T" 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-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz
0W1" 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>
  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.1/dist/css/bootstrap.min.css">
  <script src="https://cdn.jsdelivr.net/npm/jquery@3.6.0/dist/jquery.slim.min.js"></script>
  <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js"></script>
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.1/dist/js/bootstrap.bundle.min.js"></script>

</head>
```

```
<style>
   body{
    background-image: url('../static/images/1.png');
    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>
   <br><br><br><br><br><br><br>
      <div class="container p-3 my-3 text-dark" id="inst">
         <p>This application will recognise hand written digits in pictures and print the result. The
model used here is Convolutional Neural Networks.
         </p>
         <p>Instructions:
          <ol>
           <li>Choose your file that is image.</li>
            <li>Press the predict button</li>
            <li>The digit will be printed on the next screen!!</li>
          </ol>
         </p>
      </div>
      <section id="content">
```

```
        <div class="leftside">
        <form action="./predict" method="POST" enctype="multipart/form-data">
        <label>Upload Image File: </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" id="bt1">Predict</button>
            <button type="button" class="btn btn-light" id="bt1">  Clear  </button>
          </div>
        </form>
        </div>
    </section>

  </body>

</html>
```

The “style.css” file contains the code for styling the index page and the predict page.

“style.css” code:

```
#clear_button{
  margin-left: 15px;
  font-weight: bold;
  color: rgb(0, 174, 255);
}

#inst{
  color : rgb(172, 167, 167);
  background-color:rgb(250, 214, 250);
  font-family:URW Chancery L, cursive;
  font-weight: bold;

}

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

```css
#content{
  color: rgb(15, 15, 15);
  margin: 0 auto;
  padding: 2% 15%;
  padding-bottom: 0;
  font-family:URW Chancery L, cursive;

}

.welcome{
   text-align: center;
   position: relative;
   color: rgb(253, 253, 253);
   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%;
}
```

```css
#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: rgb(250, 214, 250);
    border: 1px solid #888;
    margin-top: 20px;
    margin-bottom: 20px;
}

.buttons_div{
  margin-bottom: 30px;
  margin-right: 80px;
}
#bt1{
  background-color: rgb(255, 210, 10);
}

#bt1:hover{
  background-color: white;
}
.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:black;
   padding: 100px 50px 65px 100px;
}

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

The app.py contains the backend code and also integrates front end with backend.

“app.py” 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 = r"C:\Users\Bharathi
Prasad\Downloads\IBM--5\IBM-Project-16835-1659623726-main\Final Deliverables\Code\data"

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

model = load_model("./models/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)
app.debug = True
app.run()
```

Github link:IBM-EPBL/IBM-Project-874-1658327661: A Novel Method for Handwritten Digit Recognition System (github.com)

DEMOLINK:
https://drive.google.com/file/d/1PRz01XYyClVGsKUxr2u4Fbbex_Gm1gLw/view?usp=share_link

https://drive.google.com/file/d/1PRz01XYyClVGsKUxr2u4Fbbex_Gm1gLw/view?usp=share_link