PROJECT REPORT

A NOVEL MEHOD FOR HANDWRITTEN DIGIT RECOGNITION

submitted by
PNT2022TMID34853

Madhu Aravind S - 962819104058
Vengatesh R - 962819104086
Vishnu S - 962819104091
Vishak V J - 962819104090

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Project Overview

In recent times, with the increase of Artificial Neural Network (ANN) deep learning has brought a dramatic twist in machine learning by making it more artificially intelligent. Deep learning is remarkably used in vast ranges of fields because of its diverse range of applications such as surveillance, health, medicine, sports, robotics, drones etc. The Handwritten Digit Recognition is one such ability of computers to recognize human handwritten digits. It is a hard task for the machine because handwritten digits are not perfect and can be made with many different flavours. The goal of this project is to provide the solution to this problem which uses the image of a digit and recognizes the digit present in the image by using the concept of Convolutional Neural Network. In this project, we train our model using Modified National Institute of Standards and Technology (MNIST)dataset. This dataset is trained using convolutional neural network algorithm with the help of Keras, which is python library for extensive computation of neural nodes supported by Tensor flow framework at backend. After training, iterative testing with more accurate model is formed. With this formed model, we will be able to predict the handwritten digits in an image.

1.2 Purpose

This project aims to meet the following objectives:

i. To develop handwritten digit recognizing system that enables users to automate the processof digit recognition using this deep learning model.
ii. To test the accuracy of the model.
iii. Efficient model which is less computation intensive.

2. LITERATURE SURVEY

2.1 Existing Problem

i. Handwritten digit recognition finds its application in various fields such as post mail sorting system, where scanned images of mail envelopes are made into queue and extract the section describing postcode to be delivered. With the help of digit recognizer, sorting of mails can be done based on these postcodes according to their region.

ii. Another application that utilizes this technique is form processing, digits are extracted from certain columns of a form and users put certain filters to get the desired results they want.

iii. But there is no interface for a user to get their images scanned and recognized which makes thetask complicated to use for a normal user.

2.2 References

1. Pal A, Singh D (2010) Handwritten English character recognition using neural network. IntJ Comput Sci Commum 1(2): 141-144,

2. Dutt A, Dutt A (2017) Handwritten digit recognition using deep learning. Int J AdvResComput Eng Technol6(7):990–997

3. Ghosh MMA, Maghari AY (2017) A comparative study on handwriting digit recognition using neural networks. In: International conference on promising electronic technologies, pp 77–81

4. Hamid,N.B.A., Sjarif, N.N.B.A.:Handwritten recognition using SVM, KNN and neural network.

2.3 Problem Statement Definition

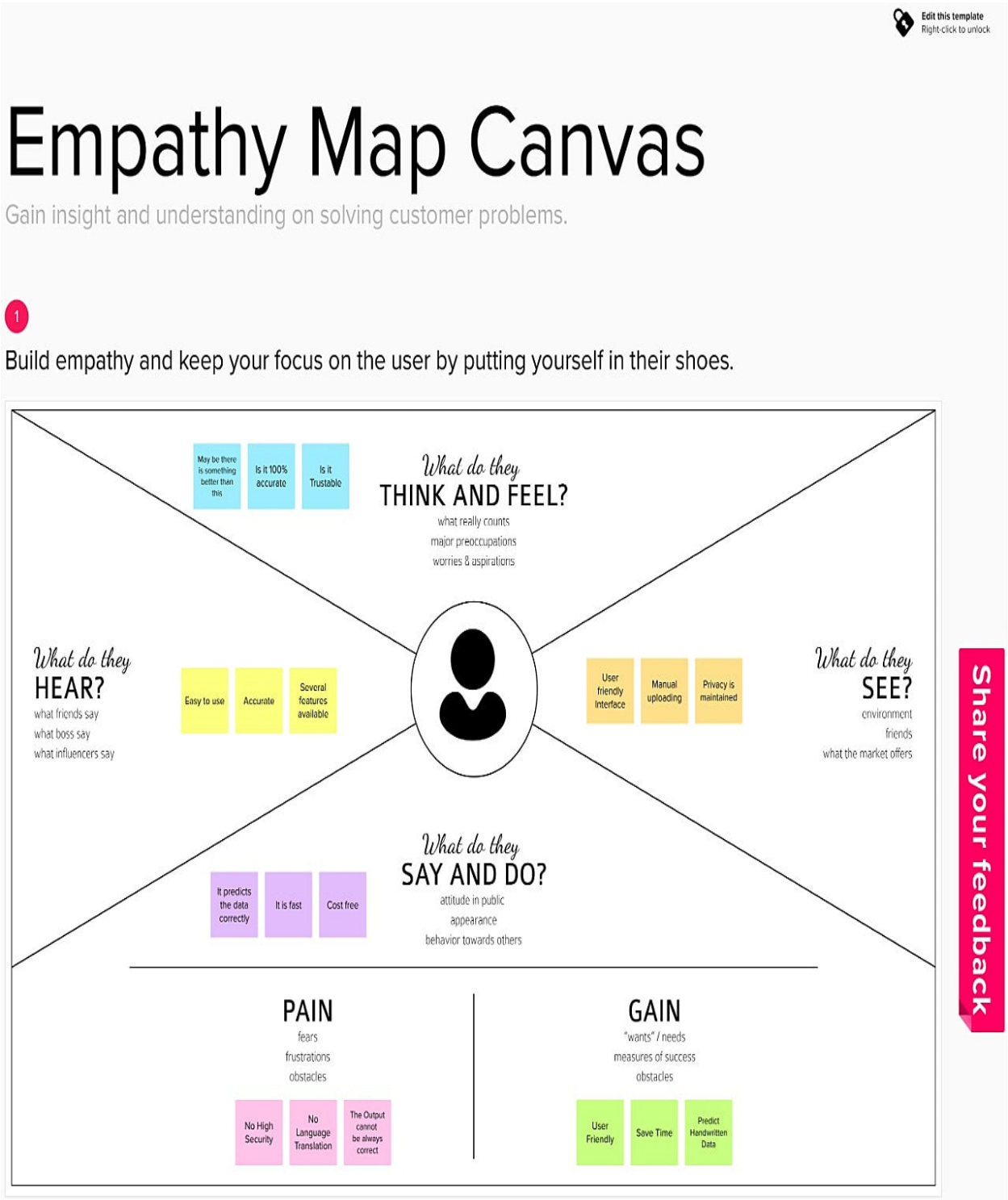
i. The Handwritten digits are not always of same size, width, orientation and justified to margins as they differ from writing of person to person.

ii. The similarity between digits such as 1 and 7, 5 and 6, 3 and 8, 2 and 7 etc. So, classifying between these numbers is also a major problem for computers.

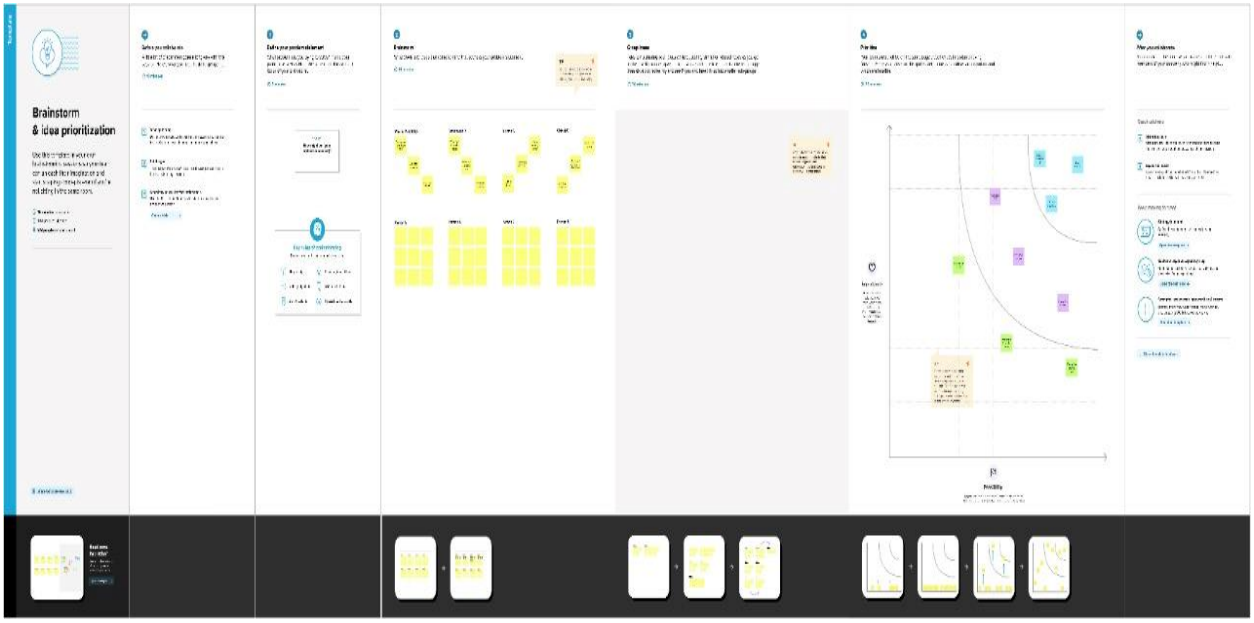
iii. The uniqueness and variety in the handwriting of different individuals also influence the formation and appearance of the digits.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming

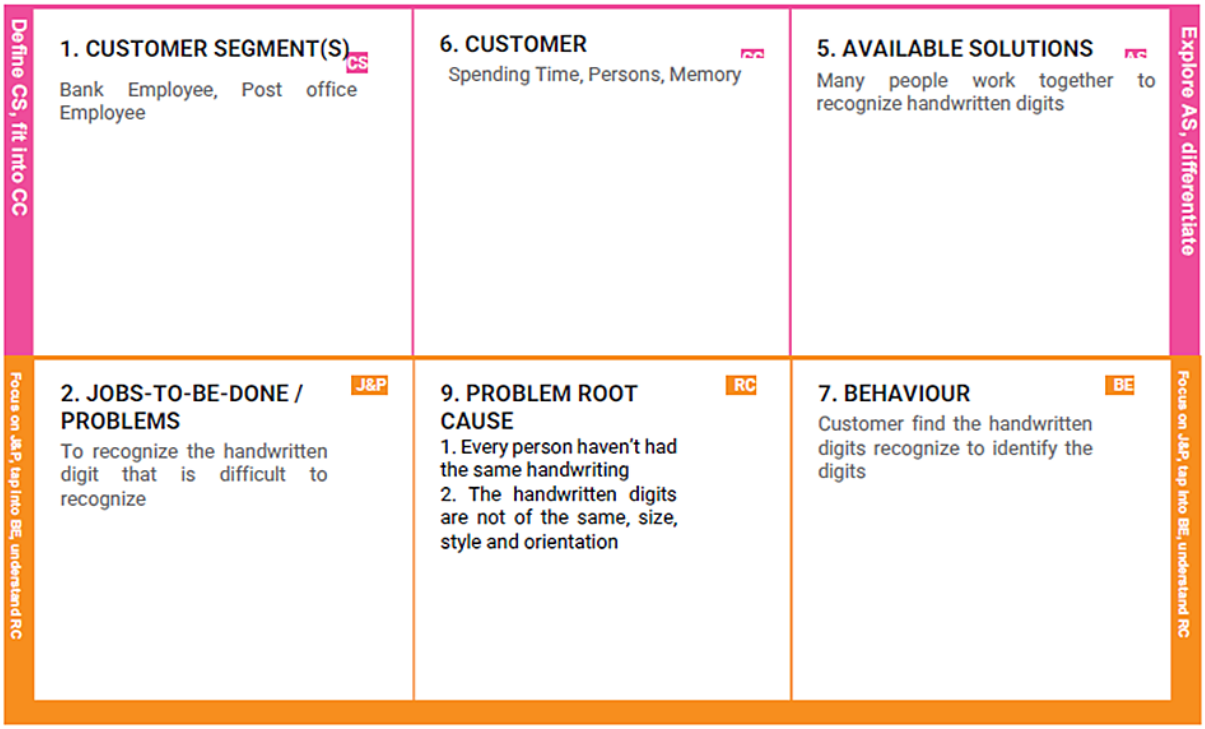


3.3 Proposed Solution

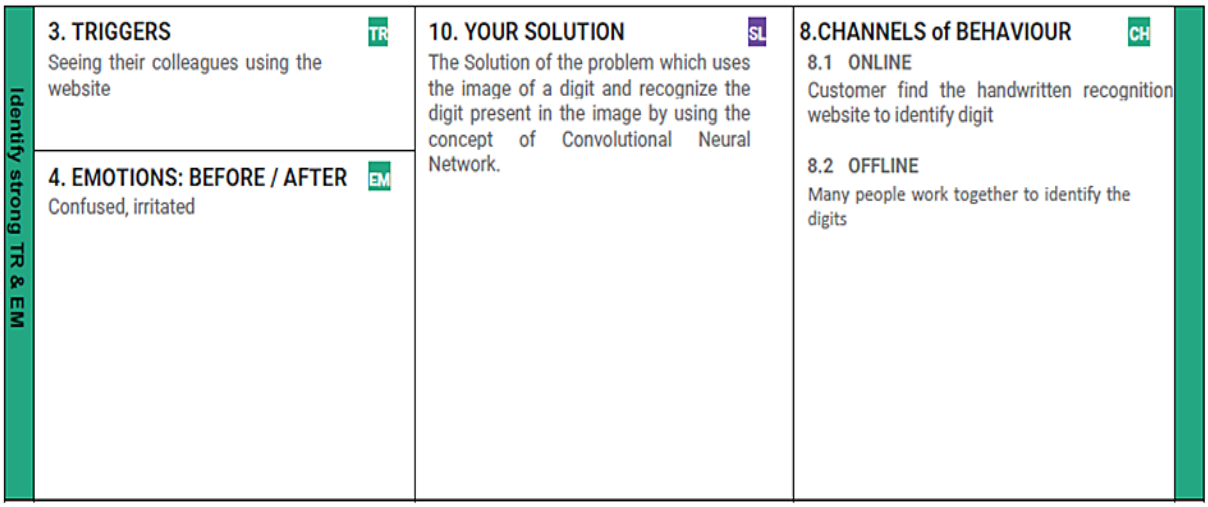
S.No.	Parameter	Description
1.	Problem Statement (Problem to besolved)	The Handwritten digits are not always of the same size, width, orientation and justified tomargins as they from writing of person to person.

2.	Idea / Solution description	By using MNIST dataset handwritten digits can be recognized. MNIST dataset contains 60000 training images of handwritten digits from Zeroto Nine and Ten thousand images for testing.
3.	Novelty / Uniqueness	User can store data.
4.	Social Impact/ Customer Satisfaction	Postal department and courier services caneasily find the digits written. Applications of handwriting recognition are numerous: **reading postal addresses, bank check amounts, and forms.**
5.	Business Model (Revenue Model)	Used in Banking sector and Postal sector. In banking sectors Numerous handwritten numbers are involved our system reduces the human mistakes.
6.	Scalability of the Solution	It recognizes the handwritten digit in high levelof accuracy.

3.4 Problem Solution Fit



8



4. REQUIREMENT ANALYSIS

4.1 Functional Requirement

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Image Data	Handwritten digit recognition refers to a computer's capacity to identify human handwritten digits from avariety of sources, such as photographs, documents,touch screens, etc., and categorize them into ten established classifications (0-9).In the realm of deep learning, this has been the subject of countless studies.
FR-2	Website	Web hosting makes the code, graphics, and other items that make up a website accessible online. A server hosts every website you've ever visited. The type of hosting determines how much space is allotted to a website on a server. Shared, dedicated, VPS, and reseller hosting are the four basic varieties.
FR-3	Digit Classifier Model	To train a convolution network to predict the digit froman image, use the MNIST database of handwritten digits.Get the training and validation data first.

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

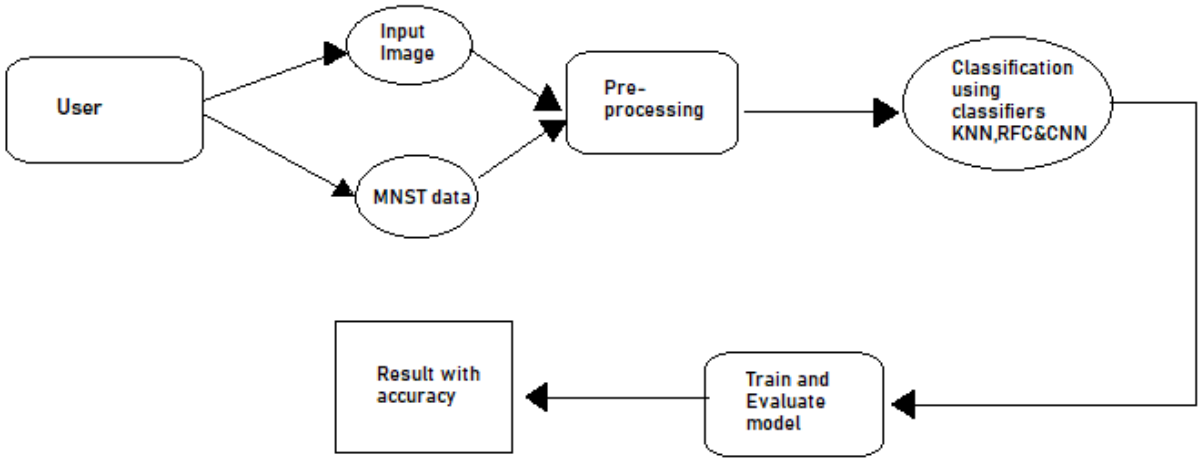
4.2 Non-Functional Requirement

FR No.	**Non-Functional Requirement**	**Description**
NFR-1	**Usability**	One of the crucial problems in pattern recognition applications is the recognition of handwritten characters. Applications for digit recognition include filling out forms,processing bank checks,and sorting mail.

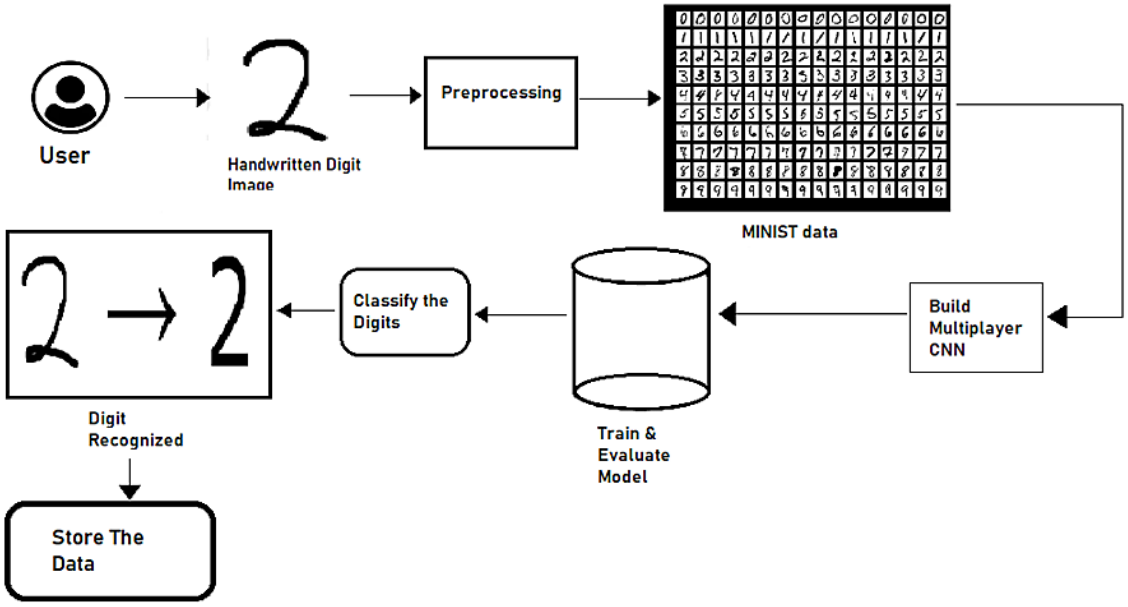
NFR-2	**Security**	1) The system generates a thorough description of the instantiation parameters, which might reveal information like the writing style, also a categorization of the digit. 2) The generative modelsare capable of segmentation driven by recognition. 3) The procedure uses a relatively.
NFR-3	**Reliability**	The samples are used by the neural network to automatically deduce rules for reading handwrittendigits. Furthermore, the network may learn more about handwriting and hence enhance its accuracy by increasing the quantity of training instances. Numerous techniques and algorithms, such as DeepLearning/CNN, SVM,Gaussian Naive Bayes,KNN, Decision Trees,Random Forests, etc.,can be used to recognize handwritten numbers.
NFR-4	**Accuracy**	With typed text in high -quality photos, optical character recognition (OCR) technology offers accuracy rates of greater than 99%. However, variances in spacing, abnormalities in handwriting, and the variety of human writing styles result in lessprecise character identification.

5. PROJECT DESIGN

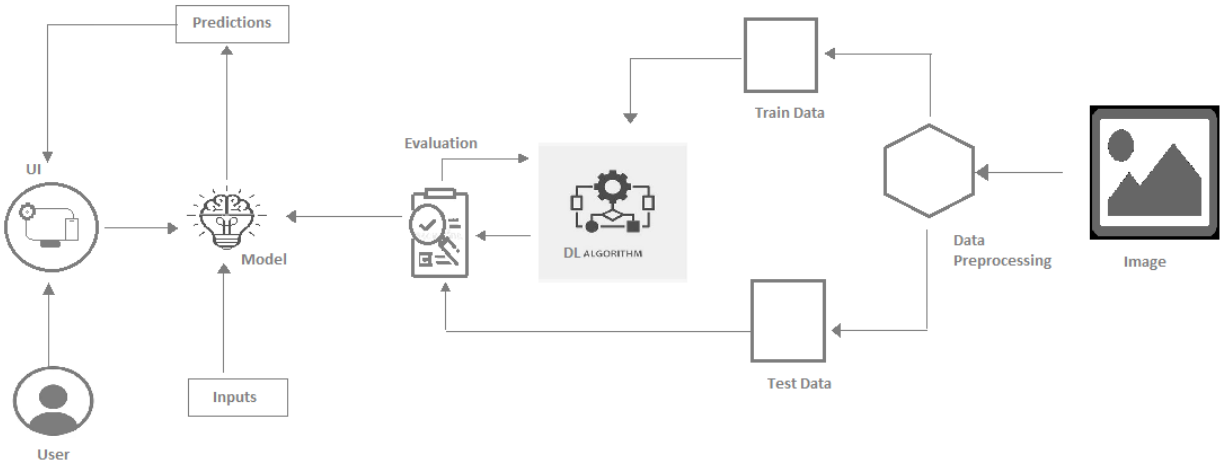
5.1 Data Flow Diagram



5.2 Solution & Technical Architecture



Solution Architecture



Technical Architecture

5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria
Customer (Bank Employee)	Recognize digits	USN-1	As a user, I need to recognize handwritten digits in the cheque.	I can guess the numbers.
		USN-2	As a user, I need to take image of digits and insert it into the application.	I can easily recognize hard to recognize handwritten digits.
		USN-3	As a user, I can check the accuracy level.	I can find the digit by high accuracy level.

Customer (Post office Employee)		USN -5	As a user, I need to recognize handwritten pin code digits in the address.	I can guess the numbers
			As a user, I need to take image of digits and insert it into the application.	I can easily recognize hard to recognize handwritten digits.
			As a user, I can check the accuracy level.	I can find the digit by high accuracy level.

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

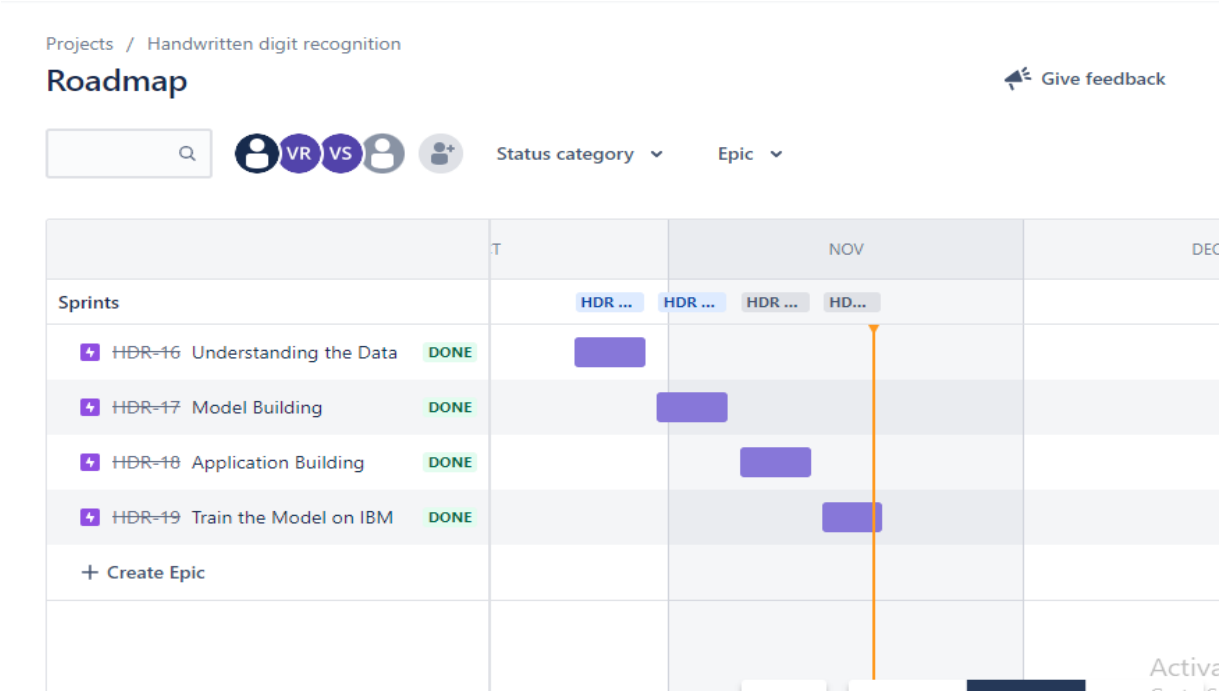
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Priority	Story Points
Sprint-1	Data Collection	USN-1	I can collect the dataset from various resourceswith different handwritings.	Low	10
Sprint-1	Data Processing	USN-2	I can load the dataset and split the data intotrain and test.	Medium	10
Sprint-2	Add CNN layers	USN-3	Creating the model and adding the input,hidden, and output layer to it.	High	5
Sprint-2	Compiling the model	USN-4	With both the training data defined and modeldefined, it's time to configure the learning process	Medium	2
Sprint-2	Train & Test themodel	USN-5	Let us train and test our model with image dataset.	Medium	6
Sprint-2	Save the model	USN-6	Model is to be saved for future purposes. This saved model can also be integrated with an webapplication in order to predict something.	Medium	2

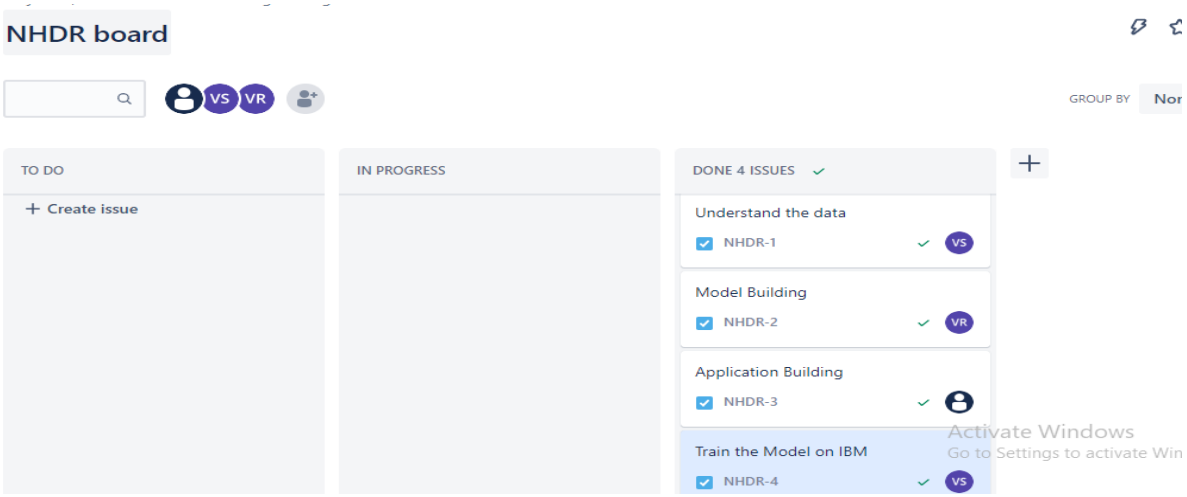
Sprint-3	UI Application Building	USN-7	Building a web application that is integrated into the model we built. A UI is provided for the uses where he has uploaded an image. The uploaded image is given to the saved model and prediction is showcased on the UI.	High	5
Sprint-3		USN-8	We use HTML to create the front end part of the web page	High	5
Sprint-3		USN-9	Build the flask file which is a web framework written in python for server-side scripting.	High	5
Sprint-3		USN-10	Run the application.	High	5
Sprint-4	Train the model on IBM	USN-11	Build Deep Learning Model Using the IBM cloud.	High	5

6.2 Sprint Delivery Schedule

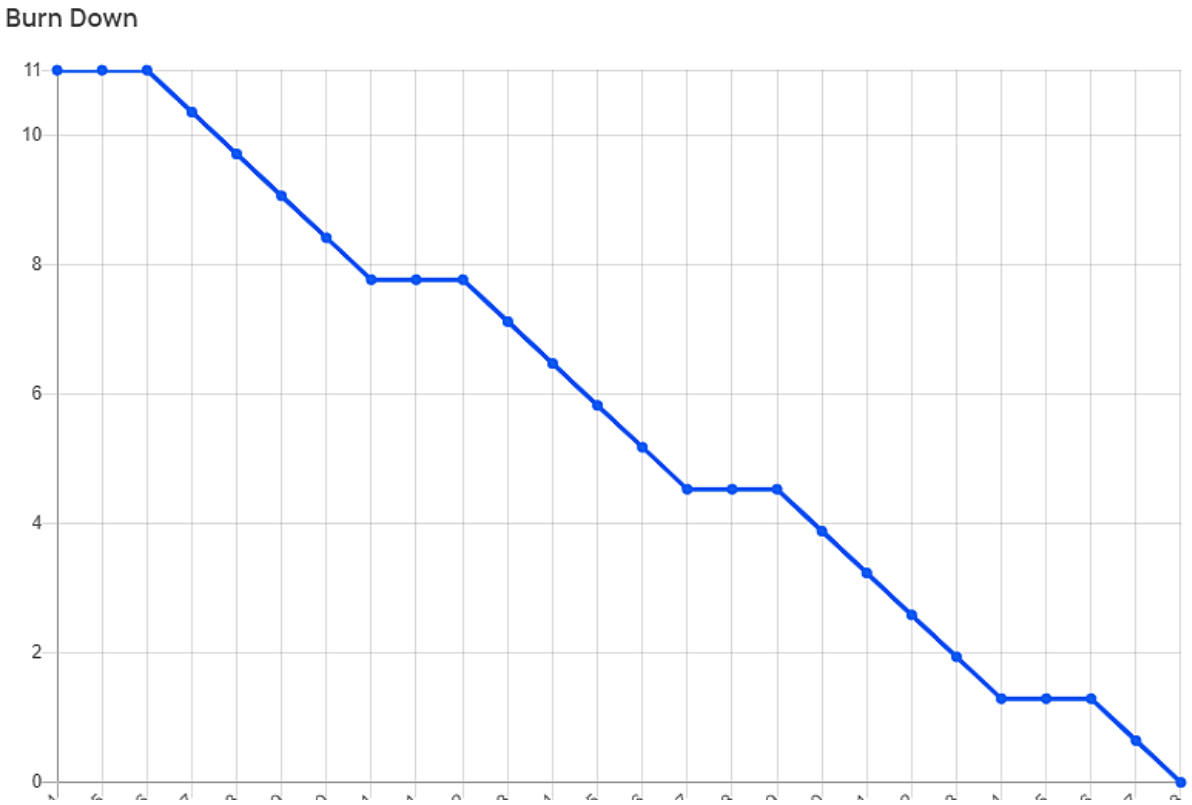
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	StoryPoints Completed (as on Planned End Date)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20
Sprint-4	20	6 Days	14 Nov 2022	18 Nov 2022	20

6.3 Reports from JIRA





Burn down Chart



7. CODING & SOLUTIONING

7.1 Feature 1

The Handwritten digit recognition web app provide an UI which is easy to use.

7.2 Feature 2

The image recognized is stored in the local directory and can be used for future reference if needed.

```
<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&display=swap" rel="stylesheet">
  <!-- bootstrap -->
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">
  <link rel="stylesheet" type= "text/css" href= "{{ url_for('static',filename='css/style.css') }}">
  <!-- fontawesome -->
  <script src="https://kit.fontawesome.com/b3aed9cb07.js" crossorigin="anonymous"></script>

  <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo" crossorigin="anonymous"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js" integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1" crossorigin="anonymous"></script>
  <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js" integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM" crossorigin="anonymous"></script>
  <script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@latest"></script>
  <style>
    body{
     background-image: url('static/image/number.jpeg');
     background-repeat: no-repeat;
     background-size: cover;
    }
  </style>

</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 class="myBox">
     <h1 class="welcome">Handwritten Digit Recognition Website</h1>
     <div>
       <section id="content">
         <div class="center">
           <form action="/predict" method="POST" enctype="multipart/form-data">
           <label><b>SELECT IMAGE TO BE RECOGNIZED</b></label><br>
           <input id="image" type="file" name="image" accept="image/png, image/jpeg" onchange="preview()"><br><br>
           <img id="frame" src="" width="150px" height="150px"/>
             <div class="buttons_div">
               <button type="submit" class="btn btn-dark" id="predict_button">Predict</button>
               <button type="button" class="btn btn-dark" id="clear_button">  Clear  </button>
             </div>
           </form>
         </div>
       </section>
     </div>
   </body>
</html>
```

```css
#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(255, 216, 62);
    border: 1px solid rgb(24, 6, 6);
    margin-top: 20px;
    margin-bottom: 20px;
}

.buttons_div{
  margin-bottom: 30px;
  margin-right: 80px;
}

.heading{
  font-family: 'Varela Round', sans-serif;
  font-weight: 700;
  font-size: 2rem;
  display: inline;
}

.center{
  width: 600px;
  box-shadow: 0 4px 8px 0 hwb(0 15% 34%), 0 6px 20px 0 rgba(217, 9, 9, 0.19);
  text-align: center;
  text-align: center;
  background-color: rgb(255, 247, 247);
```

8. TESTING

8.1 Test Cases

Test caseID	Feature Type	Component	Test Scenario	Expected Result	Actual Result	Status
HP_TC_001	Functional	Home Page	Check if usercan upload their file	The input image should be uploaded to the application successfully	Workingas expected	PASS
HP_TC_002	Functional	Home Page	Check if user upload unsupported files	The application should not allow user to select a non image file	User is not able to upload any file	PASS
HP_TC_003	Functional	Home Page	Check if the page redirects to the result page once the input is given	The page should redirect to the results page	Working as expected	PASS
BE_TC_001	Functional	Backend	Check if all theroutes are working properly	All the routes should properlywork	Working as expected	PASS
M_TC_001	Functional	Model	Check if the model can handle various image sizes	The model shouldrescale the image and predict the results	Working as expected	PASS
M_TC_002	Functional	Model	Check if themodel predicts the digit	The model should predict the number	Working as expected	PASS

M_TC_003	Functional	Model	Check if the model can handle complex 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	Workingas expected	PASS
RP_TC_002	UI	Result Page	Check if theresult is displayed properly	The result shouldbe displayed properly	Working as expected	PASS

8.2 User Acceptance Testing

DEFECT ANALYSIS

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Total
By Design	1	0	1	0	2
Duplicate	0	0	0	0	0
External	0	0	2	0	2
Fixed	4	1	0	1	6
Not Reproduced	0	0	0	1	1
Skipped	0	0	0	1	1
Won’tFix	1	0	1	0	2

Total	6	1	4	3	14

TEST CASE ANALYSIS

Section	Total Cases	Not Tested	Fail	Pass
Client Application	10	0	3	7
Security	2	0	1	1
Performance	3	0	1	2
Exception Reporting	2	0	0	2

9. RESULTS

9.1 Performance Metrics

CONTENT	VALUE
Training Accuracy	99.4%
Training Loss	3.50%
Validation Accuracy	97.7%
Validation Loss	12.23%

10. ADVANTAGES & DISADVANTAGES

10.1 Advantages

- Postal department and courier services caneasily find the digits written.
- The similarity between digits such as 1 and 7, 5 and 6, 3 and 8, 2 and 7 are easily identified.
- It is an user friendly application.

10.2 Disadvantages

- Image with dim lighting and blurry or unclear is not properly predicted.
- Only predict single image.
- Cannot handle complex data.

11. CONCLUSION

The objectives with which this project was initiated such as to develop handwritten digit recognizing system that enables users to recognize their handwritten digits using this deep learning model less computation intensive efficient model has been achieved. The model which I built got an average accuracy of 98.23%. Also the underlying problems of not having the same size, width, orientation, and margin always has been taken care of with the help of computer vision's OpenCV library's functionalities. The problem of difficulty in distinguishing the difference between digits such as 1 and 7, 5 and 6, 3 and 8 etc has been resolved to a great extent with the OpenCV's edge detection and contour features. This project is based on a deep neural network where users are going to get an interface for recognition of their digit images. On the top of this model, this project can be extended to append various functionalities which can be used to filter the desired results based on digits recognized by this model.

12. FUTURE SCOPE

The task of handwritten digit recognition, using a classifier, has great importance and use such as – online handwriting recognition on computer tablets, recognize zip codes on mail for postal mail sorting, processing bank check amounts, numeric entries in forms filled up by hand (for example - tax forms) and so on.

13. APPENDIX

SOURCE CODE

index.html

```
<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&display=swap" rel="stylesheet">
  <!-- bootstrap -->
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">
  <link rel="stylesheet" type= "text/css" href= "{{ url_for('static',filename='css/style.css') }}">
  <!-- fontawesome -->
  <script src="https://kit.fontawesome.com/b3aed9cb07.js" crossorigin="anonymous"></script>

  <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo" crossorigin="anonymous"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js" integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1" crossorigin="anonymous"></script>
  <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js" integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM" crossorigin="anonymous"></script>
  <script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@latest"></script>
  <style>
    body{
     background-image: url('static/image/number.jpeg');
     background-repeat: no-repeat;
     background-size: cover;
    }
  </style>

</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 class="myBox">
     <h1 class="welcome">Handwritten Digit Recognition Website</h1>
     <div>
       <section id="content">
         <div class="center">
           <form action="/predict" method="POST" enctype="multipart/form-data">
           <label><b>SELECT IMAGE TO BE RECOGNIZED</b></label><br>
           <input id="image" type="file" name="image" accept="image/png, image/jpeg" onchange="preview()"><br><br>
           <img id="frame" src="" width="150px" height="150px"/>
             <div class="buttons_div">
               <button type="submit" class="btn btn-dark" id="predict_button">Predict</button>
               <button type="button" class="btn btn-dark" id="clear_button">  Clear  </button>
             </div>
           </form>
         </div>
       </section>
     </div>
   </body>
</html>
```

predict.html

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

<style>
    body{
     background-image: url('static/image/numbers3.jpg');
     background-repeat: no-repeat;
     background-size: cover;
    }

    #circle{
        background: lightblue;
        border-radius: 50%;
        width: 200px;
        height: 200px;
        position:absolute;
        top:25%;
        left:25%;
        transform:translate(-50%,-50%);
    }

    #ans{
     text-align: center;
     font-size: 35px;
     margin: 0 auto;
     padding: 3% 5%;
     padding-top: 40%;
     color: rgb(57, 1, 7);
    }

</style>
<body>
    <div id="circle">
        <h1 id="ans">Result : {{num}}</h1>
    </div>
</body>
</html>

#clear_button{
    margin-left: 15px;
    font-weight: bold;
    color: rgb(22, 24, 6);;
}

#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;
    text-shadow: 0 0 3px #340808;
    position: relative;
    color: honeydew;
    background-image: linear-gradient(rgb(252, 195, 61), rgb(221, 220, 246));
    padding-top: 1%;
    padding-bottom: 1%;
    font-weight: bold;
    font-family: 'Prompt', sans-serif;
}

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

#predict_button{
    margin-right: 15px;
    color: rgb(21, 23, 8);
    font-weight: bold;
}
```

```css
#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(255, 216, 62);
    border: 1px solid rgb(24, 6, 6);
    margin-top: 20px;
    margin-bottom: 20px;
}

.buttons_div{
  margin-bottom: 30px;
  margin-right: 80px;
}

.heading{
  font-family: 'Varela Round', sans-serif;
  font-weight: 700;
  font-size: 2rem;
  display: inline;
}

.center{
  width: 600px;
  box-shadow: 0 4px 8px 0 hwb(0 15% 34%), 0 6px 20px 0 rgba(217, 9, 9, 0.19);
  text-align: center;
  text-align: center;
  background-color: rgb(255, 247, 247);
```

style.css

```css
#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(255, 216, 62);
    border: 1px solid rgb(24, 6, 6);
    margin-top: 20px;
    margin-bottom: 20px;
}

.buttons_div{
  margin-bottom: 30px;
  margin-right: 80px;
}

.heading{
  font-family: 'Varela Round', sans-serif;
  font-weight: 700;
  font-size: 2rem;
  display: inline;
}

.center{
  width: 600px;
  box-shadow: 0 4px 8px 0 hwb(0 15% 34%), 0 6px 20px 0 rgba(217, 9, 9, 0.19);
  text-align: center;
  text-align: center;
  background-color: rgb(255, 247, 247);
```

```css
#frame{
  margin-right: 10%;
}

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

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

#main{
  width:100%;
  height:100vh;
}
.left-column{
  float:left;
  width: 50%
}
.right-column{
  float:right;
  width: 50%
}

.myBox {
  background: url(static/image/number.jpeg');
  /*background-image: linear-gradient(rgb(215, 242, 251), rgb(113, 144, 129));*/
  height: 700px; /* You must set a specified height */
  background-repeat: no-repeat; /* Do not repeat the image */
  background-size: cover;
}

label{
  font-size: 1.5em;
  color: rgb(230, 24, 44);
}
```

```css
#frame{
  margin-right: 10%;
}

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

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

#main{
  width:100%;
  height:100vh;
}
.left-column{
  float:left;
  width: 50%
}
.right-column{
  float:right;
  width: 50%
}

.myBox {
  background: url(static/image/number.jpeg');
  /*background-image: linear-gradient(rgb(215, 242, 251), rgb(113, 144, 129));*/
  height: 700px; /* You must set a specified height */
  background-repeat: no-repeat; /* Do not repeat the image */
  background-size: cover;
}

label{
  font-size: 1.5em;
  color: rgb(230, 24, 44);
}
```

app.py

```python
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 keras.models import load_model

UPLOAD_FOLDER = (r'C:\Users\HP\PycharmProjects\Digit recognition\uploads')

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

model = load_model("Digits.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()
```

Model Creation

Importing the required libraries

```
[ ] 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 #convolutiona 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
```

load dataset

```
[ ] (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 [==============================] - 0s 0us/step
```

```
print (x_train.shape)  #shape is used for give the dimens ion values #60000-rows 28x28-pixels
print (x_test.shape)
```

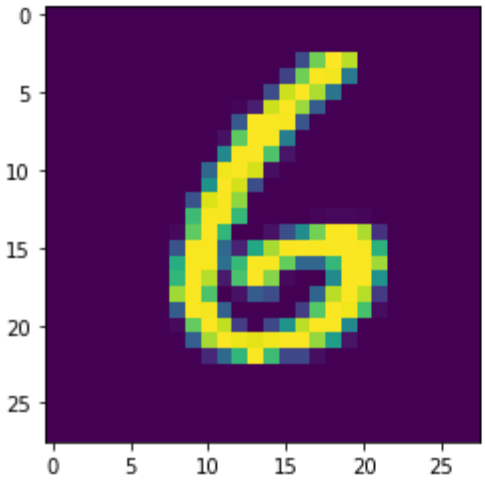
```
(60000, 28, 28)
(10000, 28, 28)
```

```
[ ] x_train[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],
      [  0,   0,   0,   0,   0,   0,   0,   0,   0,  14,   1, 154, 253,
        90,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
         0,   0],
      [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0, 139, 253,
```

```
plt.imshow(x_train[6000])
```

```
<matplotlib.image.AxesImage at 0x7fbcceef9d90>
```




```
[ ] np.argmax(y_train[6000])
```

```
0
```

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

Applying One Hot Encoding

```
[ ] number_of_classes = 10
```

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

Add CNN Layers

```
[ ] model=Sequential ()
```

```
[ ] model.add(Conv2D(64, (3, 3), input_shape=(28, 28, 1), activation='relu'))
    model.add(Conv2D(32, (3, 3), activation = 'relu'))
```

```
[ ] model.add(Flatten())
```

```
[ ] model.add(Dense(number_of_classes,activation = 'softmax'))
```

Compiling the model

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

```
[ ] x_train = np.asarray(x_train)
    y_train = np.asarray(y_train)
```

Train the model

```
[ ] model.fit(x_train, y_train, validation_data=(x_test, y_test), epochs=5, batch_size=32)
```

```
Epoch 1/5
1875/1875 [==============================] - 189s 101ms/step - loss: 0.2617 - accuracy: 0.9476 - val_loss: 0.0952 - val_accuracy: 0.9728
Epoch 2/5
1875/1875 [==============================] - 182s 97ms/step - loss: 0.0720 - accuracy: 0.9783 - val_loss: 0.0830 - val_accuracy: 0.9750
Epoch 3/5
1875/1875 [==============================] - 180s 96ms/step - loss: 0.0510 - accuracy: 0.9846 - val_loss: 0.0832 - val_accuracy: 0.9766
Epoch 4/5
1875/1875 [==============================] - 181s 97ms/step - loss: 0.0412 - accuracy: 0.9873 - val_loss: 0.0857 - val_accuracy: 0.9772
Epoch 5/5
```



```
metrics = model.evaluate(x_test, y_test, verbose=0)
print("Metrics (Test loss &Test Accuracy) : ")
print(metrics)
```

```
Metrics (Test loss &Test Accuracy) :
[0.1200210228562355, 0.9760000109672546]
```

Test The Model

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

```
1/1 [==============================] - 0s 99ms/step
[[1.0023182e-17 7.0110084e-20 7.4308840e-19 2.4388989e-09 4.2247588e-09
  7.5422962e-10 1.0877366e-13 2.6907335e-06 1.6234138e-10 9.9999726e-01]]
```

```
[ ] np.argmax(y_test[6000:6001])
```

```
9
```

```
[ ] model.save('models/Digits.h5')
```

```
from keras.preprocessing import image
from PIL import Image
import numpy as np
```

```
[ ] img = Image.open("/content/3rd.png").convert("L") # convert image to monochrome
    img = img.resize( (28,28) )
```

```
[ ] img
```

```
[ ] im2arr = np.array(img)
    im2arr = im2arr.reshape(1, 28, 28, 1)
```

```
[ ] pred = model.predict(im2arr)
    print(pred)
```

```
1/1 [==============================] - 0s 64ms/step
[[2.1331334e-21 5.4386332e-14 5.4377144e-16 1.0000000e+00 1.0853302e-19
  1.8976945e-16 1.9485798e-19 1.6560036e-20 1.6833331e-13 4.1503456e-14]]
```

```
[ ] num = np.argmax(pred, axis=1)
    print(num[0])
```

```
3
```

GITHUB Link
https://github.com/IBM-EPBL/IBM-Project-28294-1660110089

Demo Video Link
https://drive.google.com/file/d/15O67NnzTMr3AR97curiXE5kGeswEwULk/view?usp=share_link