PROJECT REPORT

A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

1. INTRODUCTION

1.1PROJECT OVERVIEW

Handwritten recognition Is the capability of the computer to identify and understand handwritten digits or characters automatically. MNIST data set is widely used for this recognition process. Thus, use Artificial Neural Networks to train these images and build a deep learning model. The web application is created where the user can upload an image of a handwritten digit, this image then analysed by the model and the detected result is returned on to the user.

1.2PURPOSE

A general-purpose, fully interconnected neural-net chip was used to perform computationally intensive tasks for handwritten digit recognition. The chip has nearly 3000 programmable connections, which can be set for template matching. The templates can be reprogrammed as needed during the recognition sequence. The recognition process proceeds in four major steps. First, the image is captured using a TV camera and a digital frame grab. This image is converted, using a digital computer, to either black or white pixels and scaled to fill a 16*16-pixel frame. Next, using the neural-net chip, the image is skeletonized, i.e., the image is thinned to a backbone one pixel wide. Then, the chip is programmed, and a feature map is created by template-matching stored primitive patterns on the chip with regions on the skeletonized image.

Finally, recognition, based on the feature map, is achieved using any one of a variety of statistical and heuristic techniques on a digital computer. Best scores range between 90% and 99% correct classification, depending on the quality of the original handwritten digits.

2. LITERATURE SURVEY

2.1 EXISTING PROBLEM

The handwritten digits are not always of the same size, width, orientation and justified to margins as they differ from writing of person to person, so the general problem would be while classifying the digits due to the similarity between digits such as 1 and 7, 5 and 6, 3 and 8, 2 and 5, 2 and 7, etc. This problem is faced more when many people write a single digit with a variety of different handwritings. Lastly, the uniqueness and variety in the handwriting of different individuals also influence the formation and appearance of the digits. Now we introduce the concepts and algorithms of deep learning and machine learning.

- Education
- Health care
- Stock market
- Other users

2.2 REFERENCES

[1] Ishani Patel,Virag Jagtap ,Ompriya Kale ,"A Survey on Feature Extraction Methods for Handwritten Digits Recognition", IJCA(0975 – 8887), Volume 107 – No 12, Dec (2015).

[2] K. Gaurav and Bhatia P. K., “Analytical Review of Preprocessing Techniques for Offline Handwritten Character Recognition”, 2nd International Conference on Emerging Trends in Engineering & Management, ICETEM, 2014.

[3] Salvador España-Boquera, Maria J. C. B., Jorge G. M. and Francisco Z. M., “Improving Offline Handwritten Text Recognition with Hybrid HMM/ANN Models”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 33, No. 4, April 2014

[4] Reena Bajaj, Lipika Dey, and S. Chaudhury, “Devnagari numeral recognition combining decision of multiple connectionist classifiers”, Sadhana, Vol.27, part. 1, pp.-59-72, 2011..

[5] U. Pal, T. Wakabayashi and F. Kimura, “Handwritten numeral recognition of six popular scripts,” Ninth International conference on Document Analysis and Recognition ICDAR 07, Vol.2, pp.749-753, 2010.

[6] Ishani Patel,Virag Jagtap ,Ompriya Kale ,“A Survey on Feature Extraction Methods for Handwritten Digits Recognition”, IJCA (0975 – 8887), Volume 107 – No 12,Dec (2015).

[7] Bharatratna P. Gaikwad, Ramesh R. Manza, Ganesh R. Manza, "Automatic Video Scene Segmentation to Separate Script and Recognition", Advances in Intelligent Systems and Computing Volume 328, pp 225-235, (2015).

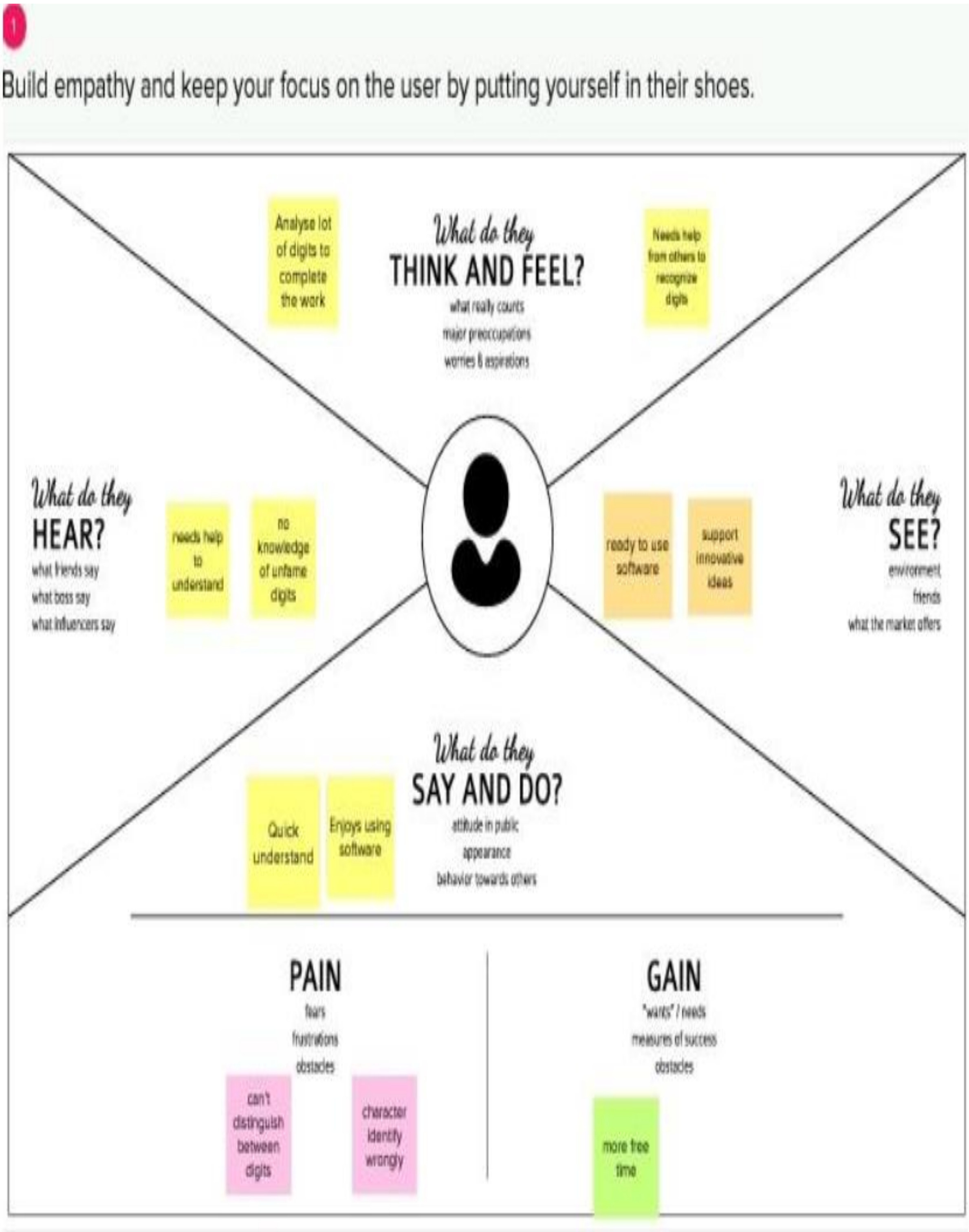
2.3 PROBLEM STATEMENT DEFINITION

The handwriting recognition system is the most basic and important step toward this huge and interesting area of Computer Vision. The agenda of a hand-written digit recognition system is to identify the digit. The ability of a machine to receive and interpret handwritten input from multiple reports like paper documents, photographs, touch screen devices, etc. By the use of AI, it can effectively recognize a particular character of type Format.

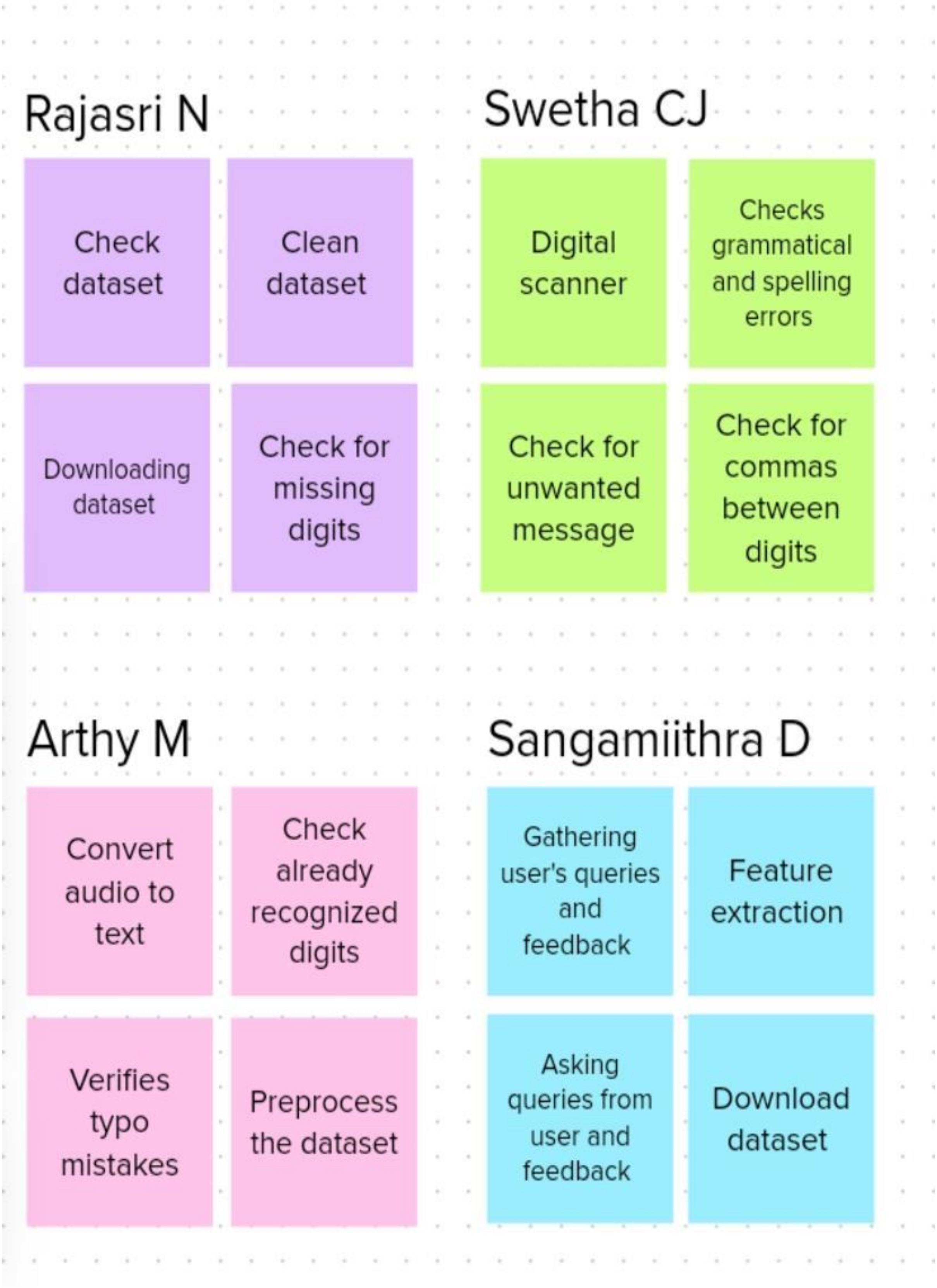
3. IDEATION & PROPOSED SOLUTION

Digit recognition system is the working of a machine to train itself or recognizing the digits from different sources like emails, bank cheque, papers, images, etc. and in different real-world scenarios for online handwriting recognition on computer tablets or system, recognize number plates of vehicles, processing bank cheque amounts, numeric entries in forms filled up by hand and so on. Developing such a system includes a machine to understand and classify the images of handwritten digits as 10 digits (0–9). Handwritten digits from the MNIST database are already famous among the community for many recent decades now, as decreasing the error rate with different classifiers and parameters along with pre-processing techniques from 12% error rate with linear classifier (1 layer NN) to achieving 0.23% error rate with hierarchy of 35 convolution neural networks. Machine learning and deep learning plays 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 many more areas. This article presents recognizing the handwritten digits (0 to 9) from the famous MNIST dataset, comparing classifiers like KNN, PSVM, NN and convolution neural network on basis of performance, accuracy, time, sensitivity, positive productivity, and specificity with using different parameters with the classifiers.

3.1 EMPATHY MAP CANVAS


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

3.2 IDEATION & BRAINSTORMING



Rajasri N

Check dataset

Clean dataset

Downloading dataset

Check for missing digits

Swetha CJ

Digital scanner

Checks grammatical and spelling errors

Check for unwanted message

Check for commas between digits

Arthy M

Convert audio to text

Check already recognized digits

Verifies typo mistakes

Preprocess the dataset

Sangamiithra D

Gathering user's queries and feedback

Feature extraction

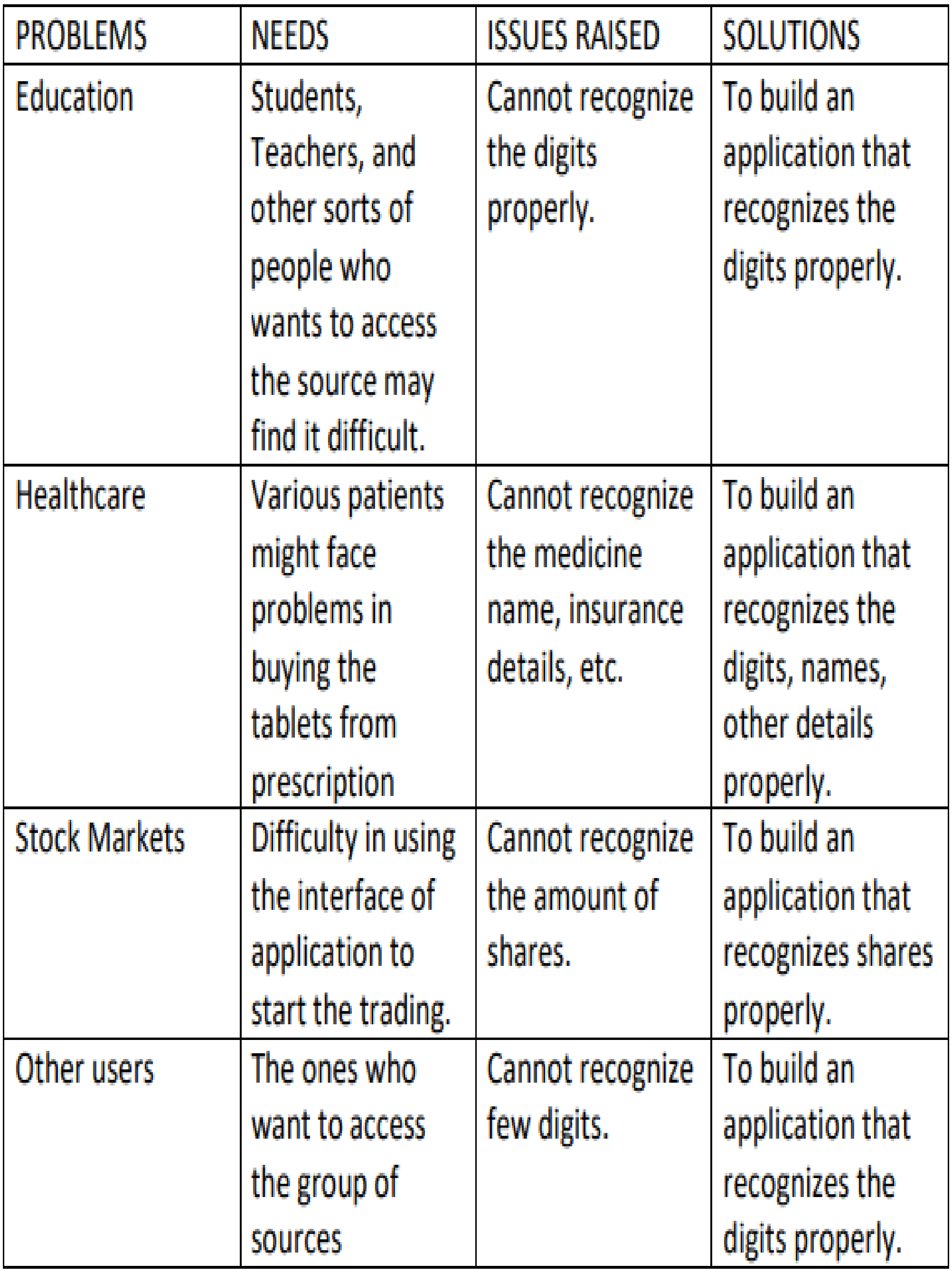
Asking queries from user and feedback

Download dataset

3.3 PROPOSED SOLUTION

- To build an application that recognizes the digits properly.

3.4 PROBLEM SOLUTION FIT



PROBLEMS	NEEDS	ISSUES RAISED	SOLUTIONS
Education	Students, Teachers, and other sorts of people who wants to access the source may find it difficult.	Cannot recognize the digits properly.	To build an application that recognizes the digits properly.
Healthcare	Various patients might face problems in buying the tablets from prescription	Cannot recognize the medicine name, insurance details, etc.	To build an application that recognizes the digits, names, other details properly.
Stock Markets	Difficulty in using the interface of application to start the trading.	Cannot recognize the amount of shares.	To build an application that recognizes shares properly.
Other users	The ones who want to access the group of sources	Cannot recognize few digits.	To build an application that recognizes the digits properly.

4. REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

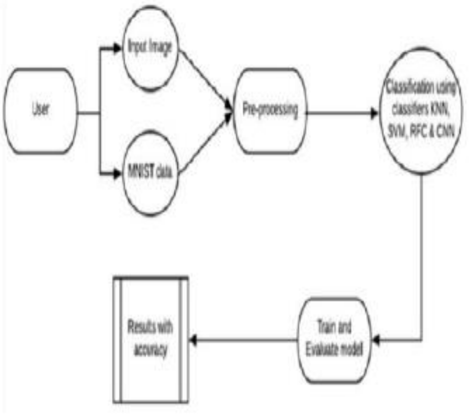
- User registration
- User login
- Image upload
- Digit recognition

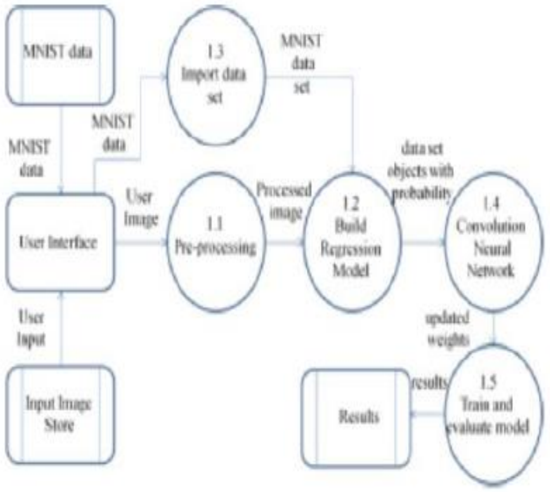
4.2 NON-FUNCTIONAL REQUIREMENTS

- Usability
- Security
- Reliability
- Performance
- Availability
- Scalability

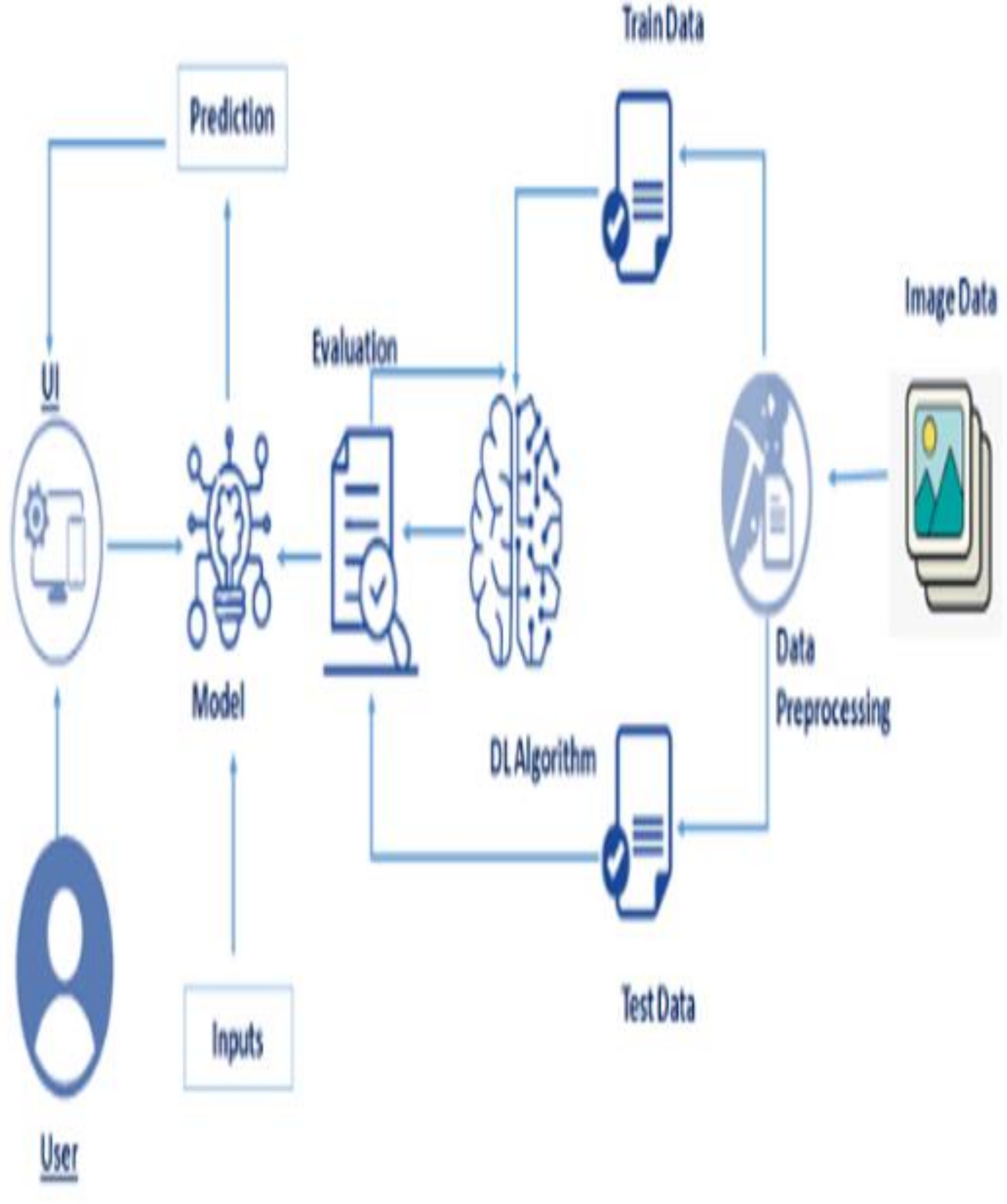
5. PROJECT DESIGN

5.1 DATA FLOW DIAGRAM





5.2 SOLUTION & TECHNICAL ARCHITECTURE



5.3 USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
Student	Authentication	USN-2	As a user, I will receive a confirmation email once I have registered for the application.	I can receive confirmation email & click confirm	High	Sprint-1
	Awareness	USN-3	As a user, I can view the guide and awareness to use this application.	I can view the awareness to use this application by a practical method.	Low	Sprint-2
	Instruction	USN-4	As a user, I can read the instructions to use this application.	I can read instructions to use it in a user-friendly method.	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	I can enter the application	High	Sprint-1

6. PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANNING & ESTIMATION

Sprint	Functional Requirement	User Story Number	User Story/ Task	Story Points	Priority	Team Members
Sprint-1	Data pre-processing	USN-1	Collect and load the dataset and split into train and test data	20	Medium	Swetha CJ, Sangamiithra.D
Sprint-2	Model building	USN-2	Build a Deep Learning Model with CNN to recognize the handwritten digit with high accuracy	10	High	Rajasri.N, Arthy.M
Sprint-2		USN-3	Train and test the model with the split Sprint-2image dataset.	5	medium	Swetha CJ, Sangamiithra.D
Sprint-2		USN-4	Models are saved and used for further integration with the Flask web user interface.	5	Medium	Rajasri.N, Arthy.M
Sprint-3	UI application	USN-5	Build a flask web application interface to upload the	5	Medium	Sangamiithra.D, Arthy.M

			handwritten image by clicking the upload button			
Sprint-3		USN-6	Integrating the flask web application with the saved deep learning model(CNN)	10	high	Rajasri.N, Swetha.CJ
Sprint-3		USN-7	User can see the predicted digits with the accuracy, in the web application that has been created.	5	Medium	Sangamiithra.D, Arthy.M
Sprint-4	Train the model on IBM	USN-8	Training the model on IBM Cloud and deploy the Flask web application with scoring end point.	20	high	Rajasri.N, Swetha.CJ

6.2 SPRINT DELIVERY SCHEDULE

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint end date(planned)	Story points completed(as on planned end date)	Sprint release date(actual)
Sprint-1	20	6 Days	2 Nov 2022	7 Nov 2022	20	7 Nov 2022
Sprint-2	20	6 Days	4 Nov 2022	9 Nov 2022	20	10 Nov 2022
Sprint-3	20	6 Days	09 Nov 2022	14 Nov 2022	20	14 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

7. CODING AND SOLUTIONING

IMPORTING THE LIBRARIES

```
from tensorflow.keras.datasets import mnist #mnist dataset
from tensorflow.keras.models import Sequential
from tensorflow.keras import layers
from tensorflow.keras.layers import Dense,Flatten
from tensorflow.keras.layers import Conv2D
from tensorflow import keras
from tensorflow.keras.optimizers import Adam
from keras.utils import np_utils
```

LOAD DATA

```
(x_train,y_train),(x_test,y_test)=mnist.load_data()
```

RESHAPING DATA

```
#(batch,height,width,channel)
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

```
no_of_classes=10
y_train=np_utils.to_categorical(y_train,no_of_classes)
y_test=np_utils.to_categorical(y_test,no_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(no_of_classes,activation='softmax'))
model.fit(x_train,y_train,validation_data=(x_test,y_test),epochs=5,batch_size=32)
```

Observing The Metrics

```
metrics=model.evaluate(x_test,y_test,verbose=0)
print("metrics-score=>test loss & accuracy")
print(metrics)
```

8. TESTING

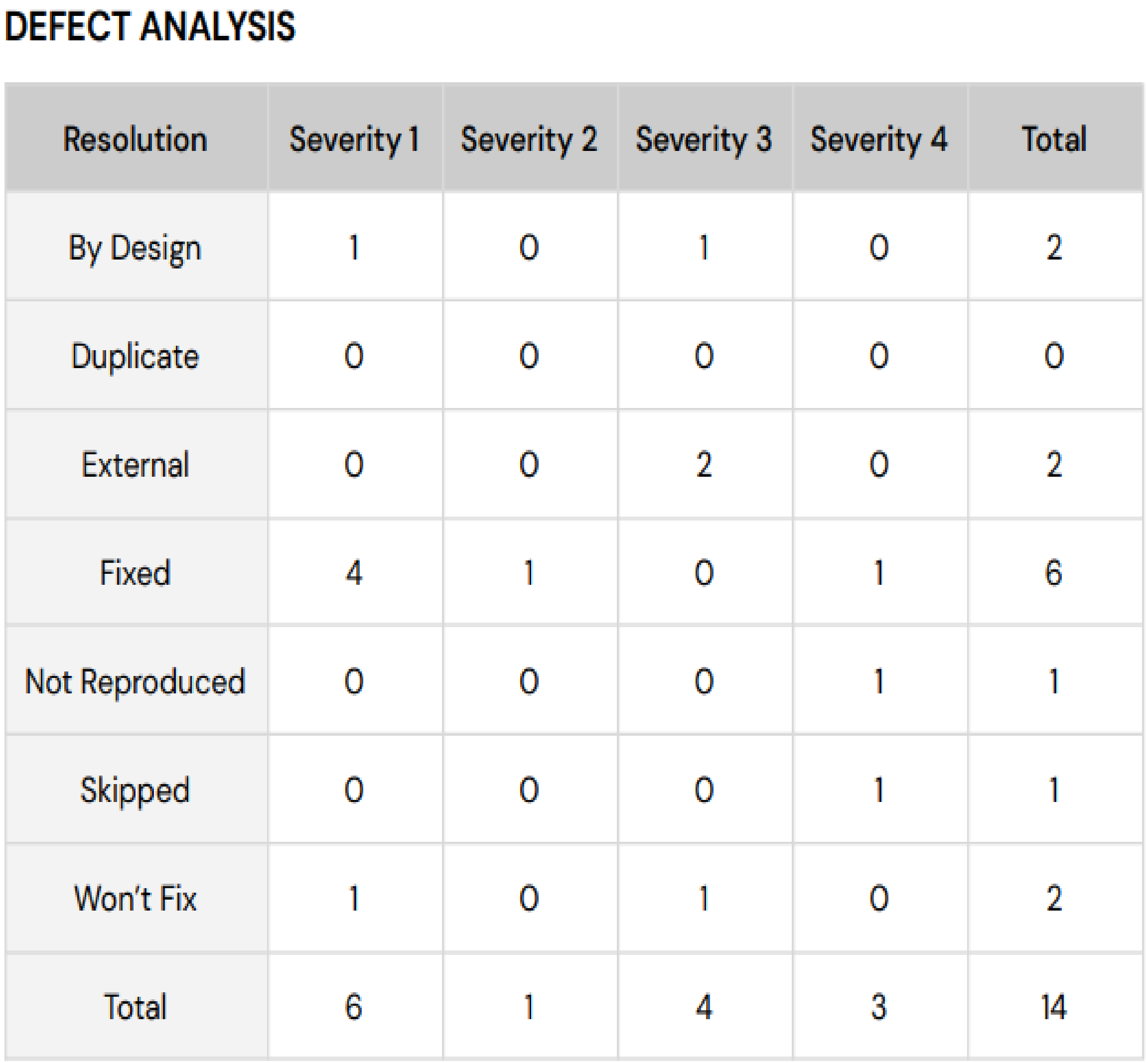
8.1 TEST CASES

- DATASET
- SUPPORT VECTOR MACHINE
- CONVOLUTIONAL NEURAL NETWORK
- . MULTILAYERED PERCEPTRON
- VISUALIZATION

8.3 USER ACCEPTANCE MODEL

PURPOSE OF THE DOCUMENT

The purpose of this document is to briefly explain the test coverage and open issues of the Handwritten Digit Recognition project at the time of the release to User Acceptance Testing (UAT). UAT is effective for ensuring quality in terms of time and software cost, while also increasing transparency with users. UAT also enables developers to work with real cases and data, and if successful, the process can validate business requirements. To be effective, UAT should be thorough and reflect user requirements, while also identifying potential problems not yet detected in previous tests. Without UAT, tested software may be released with bugs or a lack of a clearly defined goal for end users. These issues can be costly and potentially damaging to the software vendor's reputation. Internal functional experts also play a role in UAT, as they help shape UAT cycles and test management, as well as interpret the results. End users normally perform user acceptance testing. They are the most effective group to test software in this form because they know exactly how the software will be used on a daily basis and what changes need to be made to be suitable for this day-to-day use.



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't Fix	1	0	1	0	2
Total	6	1	4	3	14

9. ADVANTAGES & DISADVANTAGES

9.1 ADVANTAGES

This approach has many advantages:

1) the system not only produces a classification of the digit but also a rich description of the instantiation parameters which can yield information such as the writing style

2) the generative models can perform recognition driven segmentation.

3) the method involves a relatively small number of parameters and hence training is relatively easy and fast

4) unlike many other recognition schemes, it does not rely on some form of pre-normalization of input images, but can handle arbitrary scalings, translations and a limited degree of image rotation.

9.2 DISADVANTAGE

The disadvantage is that it is not done in real-time as a person writes and therefore not appropriate for immediate text input. Applications of offline handwriting recognition are numerous: reading postal addresses, bank check amounts, and forms. Furthermore, OCR plays an important role for digital libraries, allowing the entry of image textual information into computers by digitization, image restoration, and recognition methods.

10. CONCLUSION

As using machine learning algorithms like Neural networks along with different parameters and feature scaling vectors, we also saw the different comparison among the classifiers in terms of the most important feature of accuracy and timing. Accuracy can alter as it depends on the splitting of training and testing data, and this can further be improved if the number of training and testing data is provided. There is always a chance to improve accuracy if the size of data increases. Every classifier has its own accuracy and time consumption. We can also include the fact that if the power of CPU changes to GPU, the classifier can perform with better accuracy and less time and better results can be observed.

The performance of the classifier can be measured in terms of ability to identify a condition properly (sensitivity), the proportion of true results (accuracy), number of positive results from the procedure of classification as false positives (positive predictions) and ability to exclude condition correctly (specificity). In this, we saw a brief comparison to the classifiers of Machine learning and deep learning. Till now, the algorithms of Deep learning have performed better in the application of Handwritten Digit Recognition.

11. FUTURE SCOPE

Future studies might consider using the architecture of the convolution network which gave the best result on the MNIST database and the proposed recognition system is implemented on handwritten digits. Such more system can be designed for handwritten characters recognition, object recognition, image segmentation, handwriting recognition, text language recognition, and future studies also might consider on hardware implementation on online digit recognition system with more performance and efficiency with live results from live testing case scenarios.

12. APPENDIX

FLASK PYTHON CODE:

```
from flask import Flask, render_template, request, url_for
app = Flask(_name_)
@app.route('/')
def main():
   return
render_template("C:/Users/Swetha/PycharmProjects/pythonProject2/Templates/front pg.html")
@app.route('/predict', methods=['POST'])
def predict():
   if request.method == 'POST':
      image = request.files.get('photo', '')
```

```
        best, others, img_name = ["image"]
        return
render_template("C:/Users/Swetha/PycharmProjects/pythonProject2/Templat
es/predict.html", best=best, others=others, img_name=img_name)

    @app.route('/final', methods=['POST'])
    def final():
        if request.method == 'POST':
            image = request.files.get('photo', '')
            best, others, img_name = ["image"]
            return
render_template("C:/Users/Swetha/PycharmProjects/pythonProject2/Templat
es/final.html", best=best, others=others, img_name=img_name)
if _name_ == "_main_":
    app.run()
```

MAIN PYTHON CODE

```
import os
import random
import string
from pathlib import Path
import numpy as np
import tensorflow.keras.models
from PIL import Image, ImageOps
from keras.saving.experimental.saving_lib import load_model
def random_name_generator(n):
    return ''.join(random.choices(string.ascii_uppercase + string.digits, k=n))
def recognize(image):
    model = load_model(Path("./model/model.h5"))
    img = Image.open(image).convert("L")
    img_name = random_name_generator(10) + '.jpg'
    if not os.path.exists(f"./static/data/"):
        os.mkdir(os.path.join('./static/', 'data'))
    img.save(Path(f"./static/data/{img_name}"))
    img = ImageOps.grayscale(img)
    img = ImageOps.invert(img)
    img = img.resize((28, 28))
    img2arr = np.array(img)
    img2arr = img2arr / 255.0
    img2arr = img2arr.reshape(1, 28, 28, 1)
    results = model.predict(img2arr)
    best = np.argmax(results, axis=1)[0]
    pred = list(map(lambda x: round(x * 100, 2), results[0]))
    values = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
    others = list(zip(values, pred))
    best = others.pop(best)
    return best, others, img_nam
```

HTML CODE:

FRONT PAGE

```
<html>

<head>

   <title>Digit Recognition WebApp</title>

</head>

<body>

<style>


body{

  background-image: url('image.jpeg');

  background-repeat: no-repeat;

  background-size: cover;


 }


div.background {

  background: url('image.jpeg') repeat;

  border: 10px solid black;

}


div.transbox {

  margin: 30px;

  background-color: #ffffff;
  border: 1px solid black;
  opacity: 0.8;
}

div.transbox p {
  margin: 5%;
  font-weight: bold;
  color: #000000;
}
.button {
  display: inline-block;
  padding: 15px 25px;
  font-size: 24px;
  cursor: pointer;
  text-align: center;
  text-decoration: none;
  outline: none;
  color: #fff;
  background-color: #8A2BE2;
  border: none;
  border-radius: 15px;
  box-shadow: 0 9px #999;
}
```

```
.button:hover {background-color: #FF1493}
div.container {
  text-align: center;
}

.button:active {
  background-color: blue;
  box-shadow: 0 5px #666;
  transform: translateY(4px);
text-align:center;
}
#rectangle{
     width:400px;
     height:150px;
     background-color: #000000;
     border-radius: 15px;
     position;
     box-shadow: 0px 0px 10px 5px white;
     top:25%;
     left:50%;
     transform:translate(-50%,-50%);
    }

    #head{
```

```
  text-align: center;
  font-size: 30px;
  margin: 0 auto;
  padding: 3% 5%;
  font-family: Arial, Helvetica, sans-serif;
  color: white;
    }

    #num{
        font-size: 50px;
    }

</style>
</body>
<div class="background">
    <div class="transbox">
        <h1 class="welcome";align = "center"; style="font-
family:courier;">IBM PROJECT </h1>
        <div id="team_id"><p align = "center"; style="font-
family:courier;">TEAM ID :PNT2022TMID37375</p> </div>

        <section id="title">
            <h4 class="heading">
            <p style="font-family:courier" ; align = "center";
>Handwritten Digit Recognition Website</p></style></h4>


          <p align = "center" ; style="font-family:courier">
            The website is designed to predict the handwritten digit.
          </p>
          <p align = "center" ; style="font-family:courier">
            Handwritten digit recognition is the ability of a computer to
recognize the human handwritten digits from different sources like
images, papers, touch screens, etc, and classify them into 10
predefined classes (0-9).Handwritten character recognition is one of
the practically important issues in pattern recognition applications.
The applications of digit recognition include in postal mail sorting,
bank check processing, form data entry, etc.
          </p>


          <br>
          <p align = "center" ; style="font-family:courier">
            Hence, there comes a need for handwritten digit
recognition in many real-time applications. MNIST data set is widely
used for this recognition process and it has 70000 handwritten digits.
We use Artificial neural networks to train these images and build a
deep learning model. Web application is created where the user can
```

```
upload an image of a handwritten digit. This image is analyzed by the
model and the detected result is returned on to UI</p>

            <div class="buttons_div">
                <div class ="container">
                    <a href="predict.html"><button class="button"><p
style="font-family:courier" ; align = "center"; > NEXT
</p></button></a>
                </div>
            </div>
        </section>
    </div>
</div>
</html>
```

PREDICT PAGE:

```
<html>
<head>
  <title>Digit Recognition WebApp</title>
  <meta name="viewport" content="width=device-width">
<style>
body{
     background-image: url('image.jpeg');
```

```
    background-repeat: no-repeat;

    background-size: cover;

}

div.background {

  background: url('image.jpeg') repeat;

  border: 10px solid black;

}


div.transbox {

  margin: 30px;

  background-color: #ffffff;

  border: 1px solid black;

  opacity: 0.8;

}


div.transbox p {

  margin: 5%;

  font-weight: bold;

  color: #000000;

}
```

```
.button {
  display: inline-block;
  padding: 15px 25px;
  font-size: 24px;
  cursor: pointer;
  text-align: center;
  text-decoration: none;
  outline: none;
  color: #fff;
  background-color: 8A2BE2;
  border: none;
  border-radius: 15px;
  box-shadow: 0 9px #999;
}

.button:hover {background-color: #FF1493}
div.container {
  text-align: center;
}
```

```
.button:active {
  background-color: blue;
  box-shadow: 0 5px #666;
  transform: translateY(4px);
  text-align:center;
}
#rectangle{
     width:400px;
     height:150px;
     background-color: #000000;
     border-radius: 15px;
     position;
     box-shadow: 0px 0px 10px 5px white;
     top:25%;
     left:50%;
     transform:translate(-50%,-50%);
    }


    #head{
```

```
  text-align: center;

  font-size: 30px;

  margin: 0 auto;

  padding: 3% 5%;

  font-family: Arial, Helvetica, sans-serif;

  color: white;

    }

    #num{

        font-size: 50px;

    }

.pagination a {

  color: black;

  float: left;

  padding: 8px 16px;

  text-decoration: none;

  transition: background-color .3s;

  border: 1px solid #ddd;

}

</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>
<div class="background">
<div class="transbox">
<h1 style="font-family:courier" ; align = "center">Handwritten Digit
Recognition Website</h1>
  <section id="content">
        <div class="rightside">
        <form action="/predict" method="POST"
enctype="multipart/form-data">
       <p align='center'> <label>Select a image:</label>
```

```
<input id="image" type="file" name="image" accept="image/png,
image/jpeg" onchange="preview()"><br><br>
          <img id="frame" src="" width="100px" height="100px"/></p>
          <div class="buttons_div"/>
         <div class ="container"/>
        </form>
        </div>
  </section>
<a href="final.html"><button class="button"><p style="font-
family:courier" ; align = "center"; >Predict</p></button></a>
</body>
</html>
```

<u>FINAL PAGE</u>

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

<style>
```

```
 body{
  background-image: url('image.jpeg');
  background-repeat: no-repeat;
  background-size: cover;
 }

 #rectangle{
  width:400px;
  height:150px;
  background-color: #000000;
  border-radius: 15px;
  position:absolute;
  box-shadow: 0px 0px 10px 5px white;
  top:25%;
  left:50%;
  transform:translate(-50%,-50%);
 }

 #head{
text-align: center;
font-size: 30px;
margin: 0 auto;
padding: 3% 5%;
font-family: Arial, Helvetica, sans-serif;
```

```
 color: white;
   }

   #num{
      font-size: 50px;
   }

</style>
<body>
   <div id="rectangle">
      <h1 id="head">Predicted Number : <br><center
id="num">{{num}}</center></h1>
   </div>
</body>
</html>
```

PROJECT DEMO VIDEO LINK

https://drive.google.com/file/d/1nuwiLRHl5CEHuGH1m6umv30GGTjx-y2U/view?usp=drivesdk