PROJECT REPORT

NOVEL METHOD FOR DIGIT RECOGNITION SYSTEM

TEAM ID: PNT2022TMID42634

TEAM MEMBERS:

RENUGA DEVI K - 711119104709
SANDHIYA A - 711119104710
SRIRAM S - 711119104711
SURYA NARAYANAN P - 711119104712

1. **INTRODUCTION**
    - 1.1 Project Overview
    - 1.2 Purpose
2. **LITERATURE SURVEY**
    - 2.1 Existing problem
    - 2.2 References
    - 2.3 Problem Statement Definition
3. **IDEATION & PROPOSED SOLUTION**
    - 3.1 Empathy Map Canvas
    - 3.2 Ideation & Brainstorming
    - 3.3 Proposed Solution
    - 3.4 Problem Solution fit
4. **REQUIREMENT ANALYSIS**
    - 4.1 Functional requirement
    - 4.2 Non-Functional requirements
5. **PROJECT DESIGN**
    - 5.1 Data Flow Diagrams
    - 5.2 Solution & Technical Architecture
    - 5.3 User Stories
6. **PROJECT PLANNING & SCHEDULING**
    - 6.1 Sprint Planning & Estimation
    - 6.2 Sprint Delivery Schedule
    - 6.3 Reports from JIRA
7. **CODING & SOLUTIONING (Explain the features added in the project along with code)**
    1. Feature 1
    2. Feature 2
    3. Database Schema (if Applicable)
8. **TESTING**
    1. Test Cases
    2. User Acceptance Testing
9. **RESULTS**

1. Performance Metrics

10. **ADVANTAGES & DISADVANTAGES**

11. **CONCLUSION**

12. **FUTURE SCOPE**

13. **APPENDIX**

Source Code

GitHub & Project Demo Link

1. INTRODUCTION

1.1 PROJECT OVERVIEW

Handwritten character recognition is one of the practically important issues in pattern recognition applications. The main purpose of this projectis to build an automatic handwritten digit recognition method for the recognition of handwritten digit strings. To accomplish the recognition task, first, the digits will be segmented into individual digits. Then, a digit recognition module is employedto classify each segmented digit completing the handwritten digitstring recognition task. The applications of digit recognition include postal mail sorting, bank check processing, form dataentry, etc. The heart of the problemlies within the ability to develop an efficient algorithmthat can recognize handwritten digits and which is submitted by users by the way of a scanner, tablet, and other digital devices.

1.2 PURPOSE

Hand written 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 , numeric entries in forms filled up by hand and so on.

2.LITERATURE SURVEY

2.1 EXISTING PROBLEM

An early notable attempt in the area of character recognition research is by Grimsdale in 1959. The origin of a great deal of research work in the early sixties was based on an approach known as analysisby-synthesis method suggested by Eden in 1968. The great importance of Eden's work was that he formally proved that all handwritten characters are formed by a finite number of schematic features, a point that was implicitly included in previous works. This notion was later used in all methods in syntactic (structural) approaches of character recognition.

1. **K. Gaurav, Bhatia P. K.** , his paper deals with the various pre-processing techniques involved in the character recognition with different kind of images ranges from a simple handwritten form based documents and documents containing colored and complex background and varied intensities.In this, different preprocessing techniques like skew detection and correction, image enhancement techniques of contrast stretching, binarization, noise removal techniques, normalization and segmentation, morphological processing techniques are discussed.

2. **Sandhya Arora** , used four feature extraction techniques namely, intersection, shadow feature, chain code histogram and straight line fitting features. Shadow features are computed globally for character image while intersection features, chain code histogram features and line fitting features are computed by dividing the character image into different segments. On experimentation with a dataset of 4900 samples the overall recognition rate observed was 92.80% for Devanagari characters.

3. **Brakensiek, J. Rottland, A. Kosmala, J. Rigoll**, in their paper a system for offline cursive handwriting recognition is described which is based on Hidden Markov Models (HMM) using discrete and hybrid modelling techniques. Handwriting recognition experiments using a discrete and two different hybrid approaches, which consist of a discrete and semi-continuous structures, are compared. It is found that the recognition rate performance can be improved of a hybrid modelling technique for HMMs, which depends on a neural vector quantizer (hybrid MMI), compared to discrete and hybrid HMMs, based on tired mixture structure (hybrid - TP), which may be caused by a relative small data set.

4. **R. Bajaj, L. Dey, S. Chaudhari** , employed three different kinds of features, namely, the density features, moment features and descriptive component features for classification of Devanagari Numerals. They proposed multi classifier connectionist architecture for increasing the recognition reliability and they obtained 89.6% accuracy for handwritten Devanagari numerals.

5. **G. Pirlo and D. Impedovo** in his work on , presented a new class of membership functions, which are called Fuzzymembership functions (FMFs), for zoning-based classification. These FMFs can be easily adapted to the specific characteristics of a classification problem in order to maximize classification performance. In this research, a realcoded genetic algorithm is presented to find, in a single optimization procedure, the optimal FMF, together with the optimal zoning described by Voronoi tessellation. The experimental results, which are carried out in the field of handwritten digit and character recognition, indicate that optimal FMF performs better than other membership functions based on abstract level, ranked-level, and measurement-level weighting models, which can be found in the literature.

6. **Sushree Sangita Patnaik and Anup Kumar Panda** May 2011 , this paper proposes the implementation of particle swarm optimization (PSO) and bacterial foraging optimization (BFO) algorithms which are intended for optimal harmonic compensation by minimizing the undesirable losses occurring inside the APF itself. The efficiency and effectiveness of the implementation of two approaches are compared for two different conditions of supply. The total harmonic distortion (THD) in the source current which is a measure of APF performance is reduced drastically to nearly 1% by employing BFO. The results demonstrate that BFO outperforms the conventional and PSO based approaches by ensuring excellent functionality of APF and quick prevail over harmonics in the source current even under unbalanced supply.

7. **M. Hanmandlu, O.V. Ramana Murthy** have presented in their study the recognition of handwritten Hindi and English numerals by representing them in the form of exponential membership functions which serve as a fuzzy model. The recognition is carried out by modifying the exponential membership functions fitted to the fuzzy sets. These fuzzy sets are derived from features consisting of normalized distances obtained using the Box approach. The membership function is modified by two structural parameters that are estimated by optimizing the entropy subject to the attainment of membership function to unity. The overall recognition rate is found to be 95% for Hindi numerals and 98.4% for English numerals.

8. **Renata F. P. Neves** have proposed SVM based offline handwritten digit recognition. Authors claim that SVM outperforms the Multilayer perceptron classifier. Experiment is 12 carried out on NIST SD19 standard dataset. Advantage of MLP is that it is able to segment non-linearly separable classes. However, MLP can easily fall into a region of local minimum, where the training will stop assuming it has achieved an optimal point in the error surface. Another hindrance is defining the best network architecture to solve the problem, considering the number of layers and the number of perceptron in each hidden layer. Because of these disadvantages, a digit recognizer using the MLP structure may not produce the desired low error rate.

2.2 REFERENCES

1. https://www.tensorflow.org/learn

2. https://www.geeksforgeeks.org/python-tkinter-tutorial/

3.https://medium.com/the-andela-way/applying-machine-learning-to-recognize handwritten-characters-babcd4b8d705

4.https://nanonets.com/blog/handwritten-character-recognition/

5. https://www.geeksforgeeks.org/python-tkinter-tutorial/

6.https://medium.com/the-andela-way/applying-machine-learning-to-recognize handwritten-characters-babcd4b8d705

7. https://nanonets.com/blog/handwritten-character-recognition/

8. https://www.tensorflow.org/learn

9.R. Alhajj and A. Elnagar, "Multiagents to separating handwritten connected digits," in IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, vol. 35, no. 5, pp. 593-602, Sept. 2005. https://doi.org/10.1109/TSMCA.2005.843389

10.https://towardsdatascience.com/useful-plots-to-diagnose-your-neural-network-521907fa2f45

11.https://towardsdatascience.com/the-best-machine-learning-algorithm-for-handwritten-digits-recognition-2c6089ad8f09

12.https://www.datacamp.com/community/tutorials/convolutional-neural-networks-python

13.https://www.analyticsvidhya.com/blog/2021/01/building-a-cnn-model-with-95-accuracy/

14.https://www.analyticsvidhya.com/blog/2020/10/what-is-the-convolutional-neural-network- architecture/

15.https://www.kdnuggets.com/2018/11/top-python-deep-learning-libraries.html

16.https://analyticsindiamag.com/7-types-classification-algorithms/

17.https://towardsdatascience.com/image-recognition-with-machine-learning-on-python-image-processing-3abe6b158e

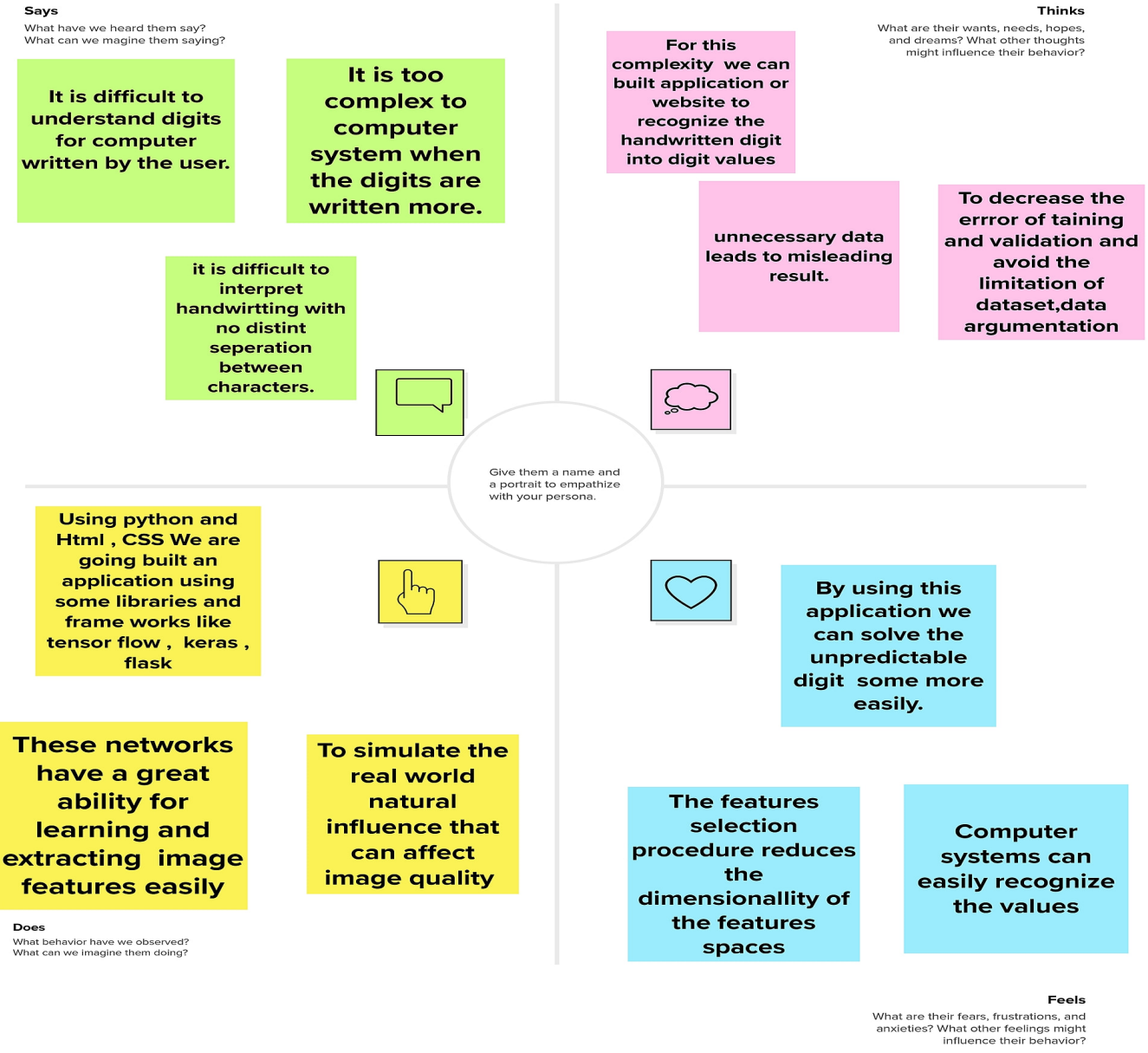
.

2.3 PROBLEM STATEMENT DEFINITION

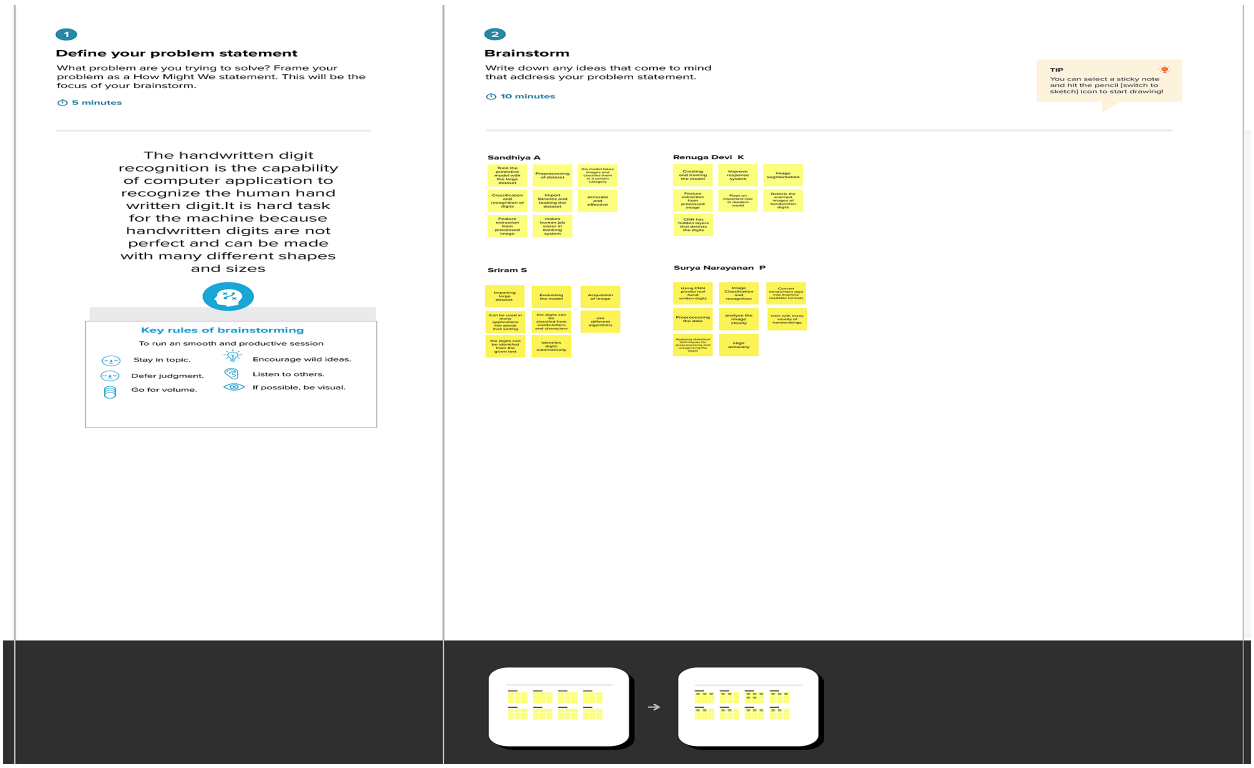
- The problem statement is to classify handwritten digits. The goal is to take an image of a handwritten digit and determine what that digit and character is.
- It is easy for the human to perform a task accurately by practicing it repeatedly and memorizing it for the next time. Human brain can process and analyse images easily. Also, recognize the different elements present in the images.
- the goal is to correctly identify digits from a dataset of tens of thousands of handwritten images and experiment with different algorithms to learn first-hand what works well and how techniques compare.
- The handwritten digit recognition is the capability of computer applications to recognize the human handwritten digits. It is a hard task for the machine because handwritten digits are not perfect and can be made with many different shapes and sizes.
- The handwritten digit recognition system is a way to tackle this problem which uses the image of a digit and recognizes the digit present in the image. Convolutional Neural Network model created using Python library over the MNIST dataset to recognize handwritten digits.
- Handwriting number recognition is a challenging problem researchers had been research into this area for so long especially in the recent years.

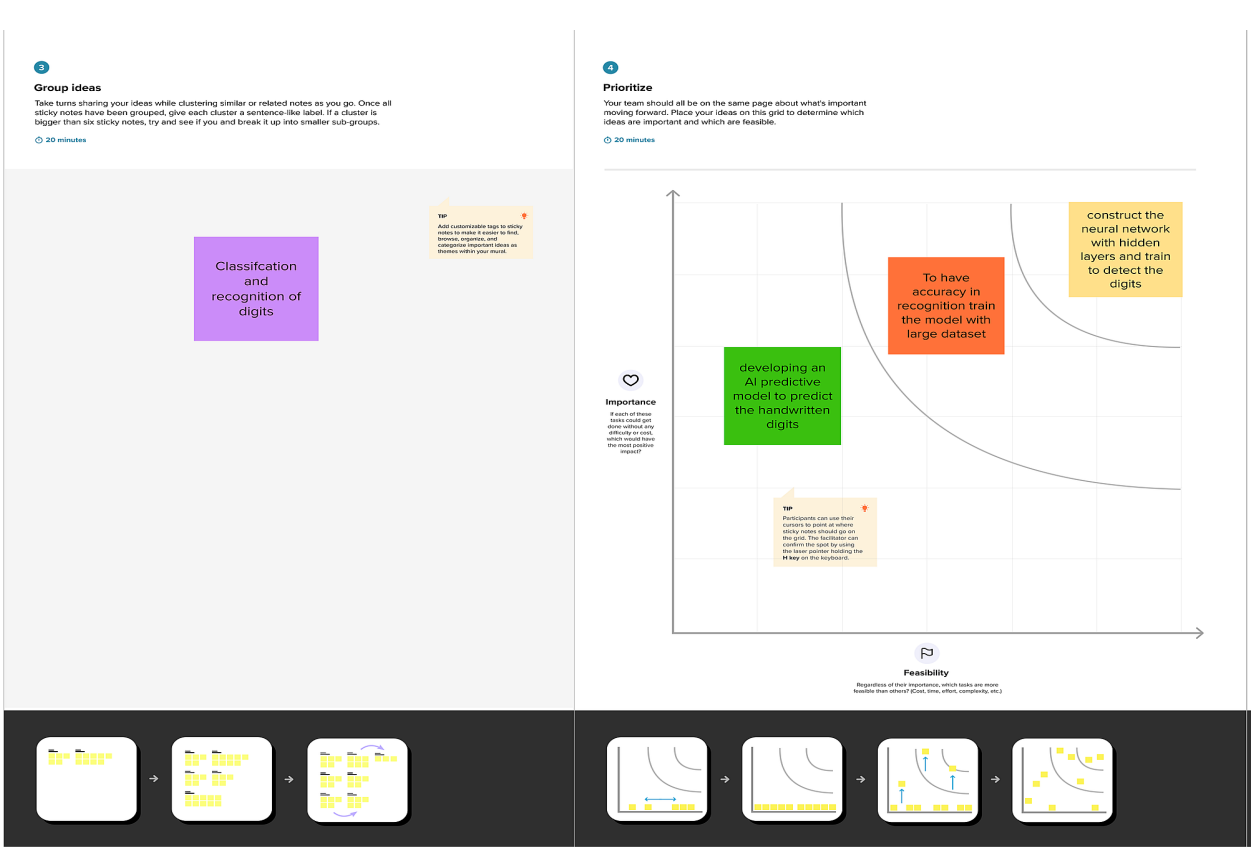
3.IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP



3.2 IDEATION AND BRAINSTROM





3.3 PROPOSED SOLUTION

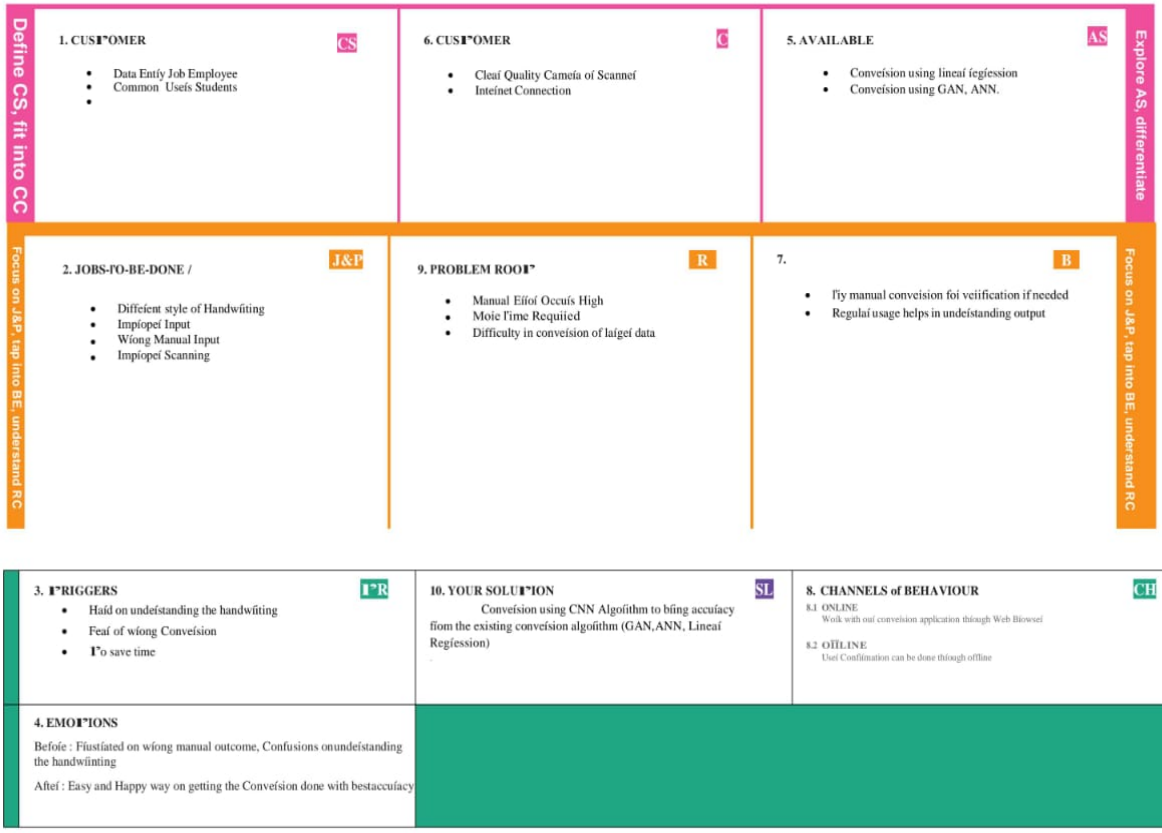
S.NO	PARAMETER	DESCRIPTION
1.	Problem Statement (Problem to be solved)	With the progress in the field of science and technology,everything is being digitized to reduce human effort.Hence there comes the need for handwritten digit recognition system in real time applications
2.	Idea / Solution description	Various AI/ML packages are used to analyse, train the model based on the problem statement. The major goal of the proposed system is understanding Convolutional Neural Network, and applying it to the handwritten recognition system
3.	Novelty / Uniqueness	It can recognize different varieties of handwriting patterns.
4.	Social Impact/ Customer Satisfaction	This system can be used to recognize the number plates in vehicles, postal codes & phone numbers
5.	Business Model (Revenue Model)	We are building a service that can be integrated into any application and pricing model will be based
6.	Scalability of the Solution	This can be extended to recognizing letters in English as a base testing and even extended to different

3.4 PROBLEM SOLUTION FIT

A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

PROBLEM SOLUTION FIT

Team ID: PNT2022TMID42634



4.REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT

FR No.	Functional Requirement (Epic)
FR-1	Image Data:An image data include digital images curated for testing,training and evaluating the performance of machine learning and artificial intelligence algorthims.
FR-2	Website: Web hosting makes the code, graphics, and other items that make up a websiteaccessible online. A server hostsevery website you'veever visited. Thetype 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:Use the MNIST database of the handwritten digits to trainthe convolutional networkto predict the digitgiven an image
FR-4	Cloud: The cloud offers a range of IT services, including virtual storage, networking, servers, databases, and applications. In plain English, cloud computing is described as a virtualplatform that enablesunlimited storage andaccess to your dataover theinternet.

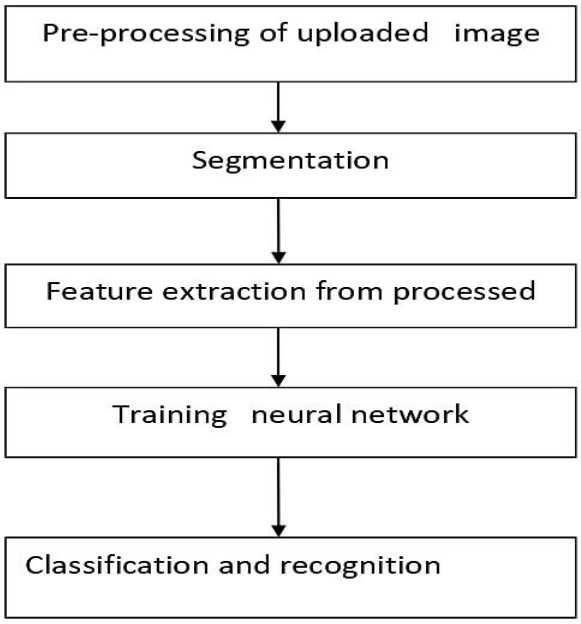
4.2 NON-FUNCTIONAL REQUIREMENTS

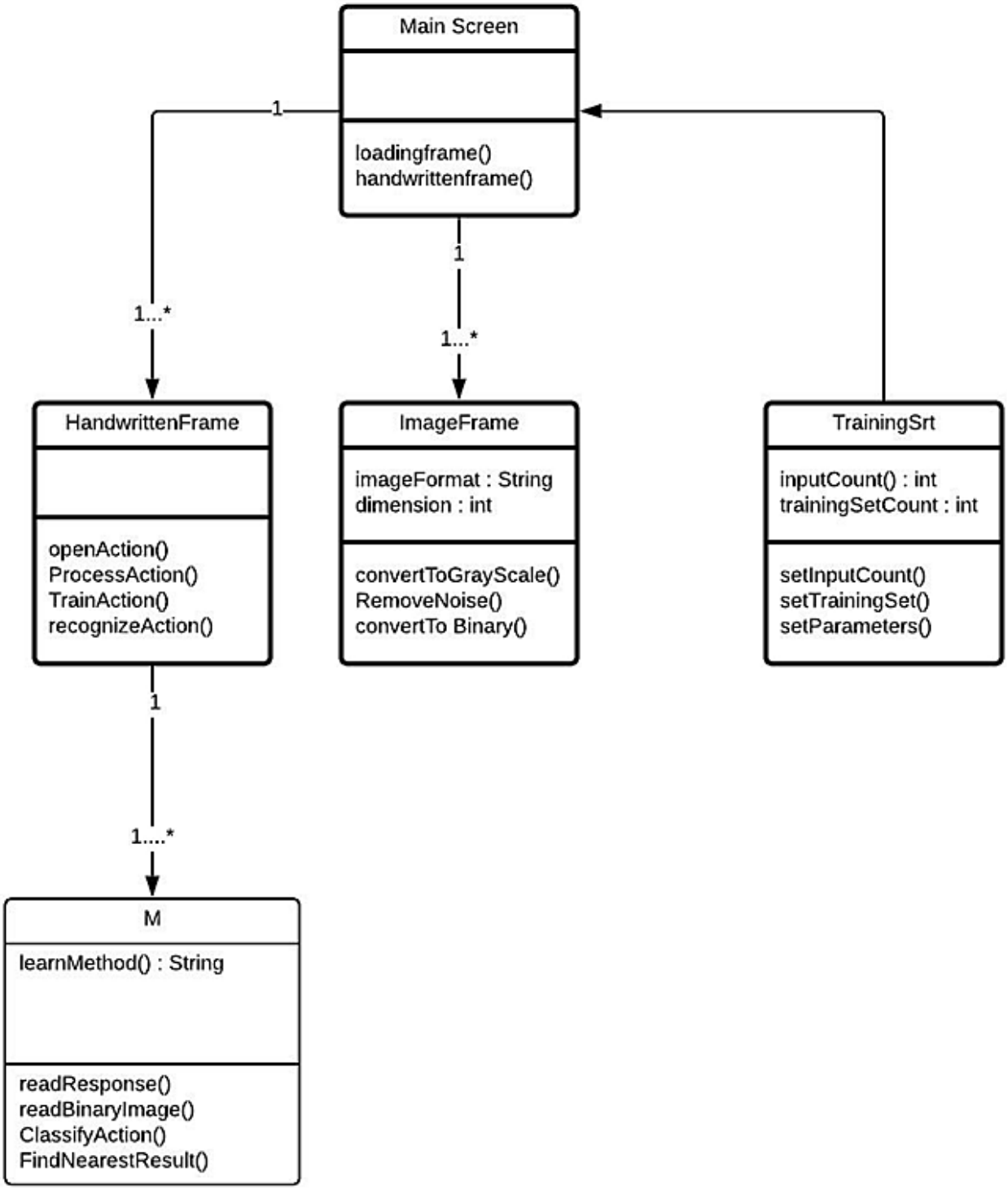
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The proposed system gives good results for images thatcontain handwritten textwritten in different styles, different size and alignment With varying background.
NFR-2	**Security**	Only authorized people can accessthe system data And modifythe database.
NFR-3	**Reliability**	The Database is frequently updated with handwriting of different stylesand size andwillRollback whenany update fails.

NFR-4	**Performance**	The proposed system is advantageous as it usesfewer features to train the neural network, Which results in faster convergence.
NFR-5	**Availability**	The systemfunctionality and services are available for usewith all operations.
NFR-6	**Scalability**	The websitetraffic limit must be scalable enough to support 2 lakhsusers at a time.

5.PROJECT DESIGN

5.1 DATA FLOW DIAGRAM




Main Screen
loadingframe()
handwrittenframe()
1
1
1...*
1...*
HandwrittenFrame
openAction()
ProcessAction()
TrainAction()
recognizeAction()
ImageFrame
imageFormat : String
dimension : int
convertToGrayScale()
RemoveNoise()
convertTo Binary()
TrainingSrt
inputCount() : int
trainingSetCount : int
setInputCount()
setTrainingSet()
setParameters()
1
1....*
M
learnMethod() : String
readResponse()
readBinaryImage()
ClassifyAction()
FindNearestResult()

5.2 SOLUTION AND TECHNICAL ARCHITECTURE

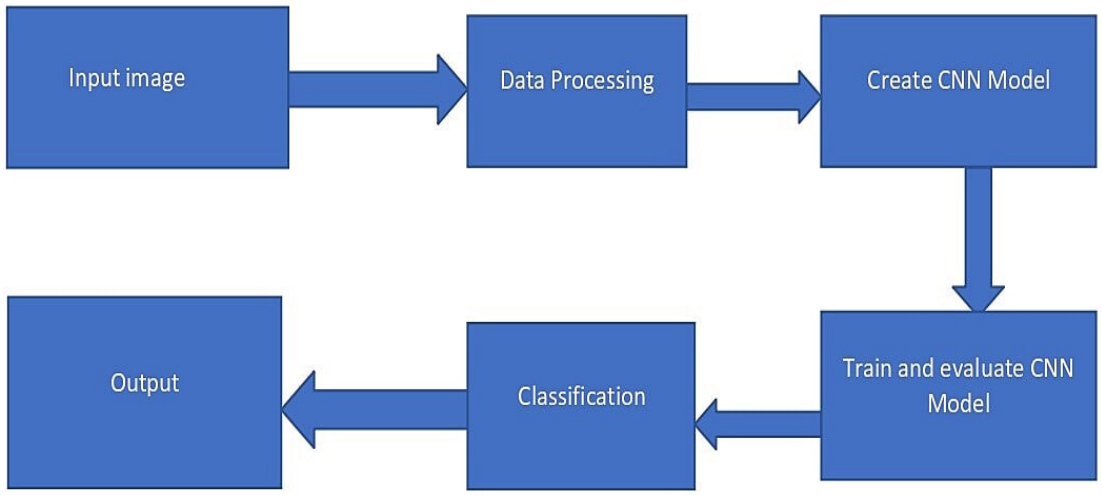
Solution Architecture:

1. 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.

2. Because of the progress in the field of science and technology, everything is being digitalized to reduce human effort. Hence, there comes a need for handwritten digit recognition in many real-time applications. MNIST data set is widelyused for this recognition process.

3. 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 returnedon to UI.

Solution Architecture Diagram:



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 will receive confirmation email onceI have registered for the application	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I can log into the application by entering email & password	I can receive confirmation email& click confirm	High	Sprint-1
	Login	USN-3	As a user, I can view the application's home page whereI can readthe instructions to usethis application	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I canable to inputthe images of digital documents to the application	As a user, I can able to input the images of digital documents to the application	Medium	Sprint-1

		USN-5	As a user I can able to get the recognised digitas output from the images of digital documents or images	I can access the recogniz ed digits from digital document or images	High	Sprint-1
	Dashboard	USN-6	As a user, I will trainand test the input to get the maximum accuracy of output.	I can able to train and test the application until it gets maximum accuracy of theresult	High	Sprint 3
Customer (Webuser)	Accessbility	USN-8	As a user, I can use the web application virtually anywhere.	I can use the application in any device with a browser	Low	Sprint 4

6.PROJECT PLANNING AND SCHEDULING

6.1 Sprint Planning & Estimation

SPRINT	USER STORY /TASK	STORY POINTS	PRIORITY	TEAM MEMBERS
Sprint - I	Get the dataset	3	High	Renuga devi K
	Explore the data	2	Medium	Renuga devi K Surya narayanan P
	Data Pre-Processing	3	High	Surya Narayanan P
	Prepare training and testing data	3	High	Sandhiya A
Sprint - II	Create the model	3	High	Sriram S
	Train the model	3	High	Sandhiya A
	Test the model	3	High	Renuga devi K

Sprint - III	Improve the model	2	Medium	Surya Narayanan P Sriram S
	Save the model	3	High	Renuga Devi K
	Build the HomePage	3	High	Sandhiya A Renuga Devi K
	Setup a database to store inputimages	2	Medium	Surya Narayanan P
Sprint -IV	Build the results page	3	High	Sriram S Sandhiya A

6.2 SPRINT DELIVERY SCHEDULE

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as onPlanned End Date)	Sprint Release Date(Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

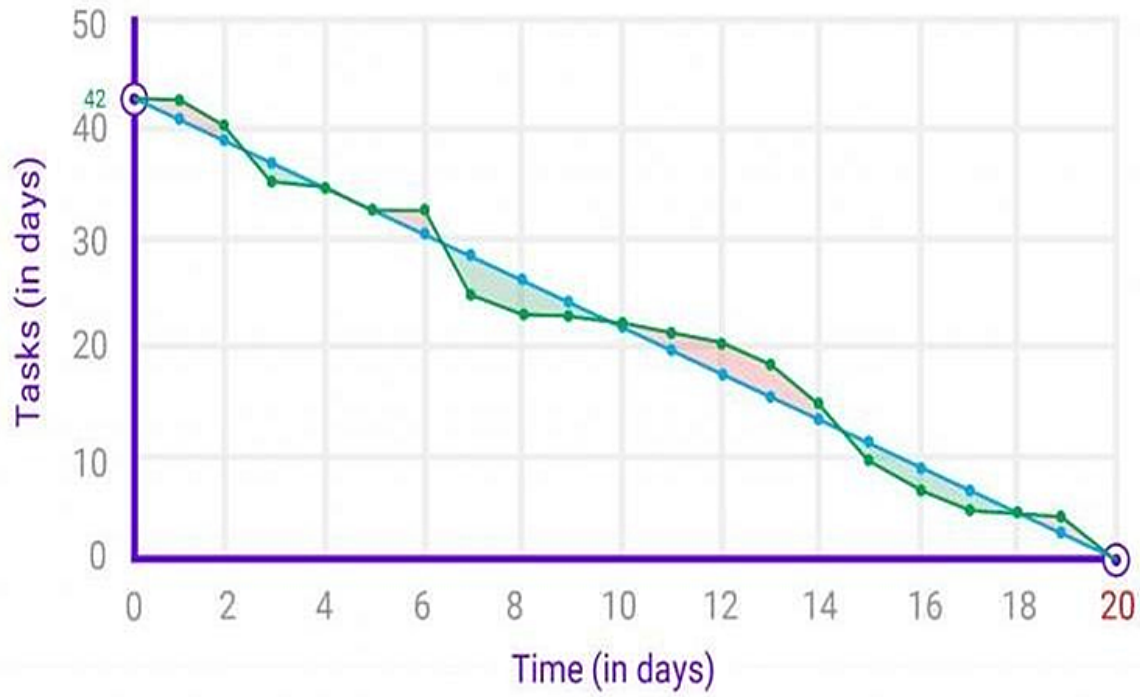
Velocity:

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

Average velocity=20/6=3.33

Burndown Chart:

A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down chartscan be applied to any project containingmeasurable progress over time.


50
42
40
30
20
10
0
Tasks (in days)
0
2
4
6
8
10
12
14
16
18
20
Time (in days)

7.CODING AND SOLUTIONING

7.1 FEATURE 1

Index.html

```
<html>


<head>
  <title>Digit Recognition WebApplicationS</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|Pacific
o&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='C:\IBM Project')
}}">
  <!-- fontawesome -->
```

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


  <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-
q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
crossorigin="anonymous"></script>
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"
integrity="sha384-
UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1"
crossorigin="anonymous"></script>
  <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"
integrity="sha384-
JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM"
crossorigin="anonymous"></script>
  <script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@latest"></script>


</head>
<style>
#clear_button{
  margin-left: 15px;
  font-weight: bold;
  color: blue;
}

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

#content{
```

```
  margin: 0 auto;
  padding: 2% 15%;
  padding-bottom: 0;
}

.welcome{
   text-align: center;
   position: relative;
   color: honeydew;
   background-color: greenyellow;
   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: blue;
  font-weight: bold;
}
```

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

#result{
  font-size: 5rem;
}

#title{
  padding: 1.5% 15%;
  margin: 0 auto;
  text-align: center;
}

.btn {
    font-size: 15px;
    padding: 10px;
    -webkit-appearance: none;
    background: #eee;
    border: 1px solid #888;
    margin-top: 20px;
    margin-bottom: 20px;
}

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

```
}

.heading{
  font-family: 'Varela Round', sans-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%; */
}
```

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

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




</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>
  <section id="title">
    <h4 class="heading">Handwritten Digit Recognition Website</h4>
    <br><br>
      <p>
        The website is designed to predict the handwritten digit(0-9).
      </p>
    <p>
        Handwriting recognition is one of the compelling research works going on because every individual in this world
        has their own style of writing. It is the capability of the computer to identify and understand
        handwritten digits or characters automatically. Because of the progress in the field of science and technology,
        everything is being digitalized to reduce human effort.</p>

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

  </section>
```

```
 <section id="content">


      <div class="leftside">
      <form action="/predict" method="POST" enctype="multipart/form-data">
      <label>Select a image:</label>
      <input id="image" type="file" name="image" accept="image/png, image/jpeg"
onchange="preview()"><br><br>
        <img id="frame" src="" width="100px" height="100px"/>
        <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>

</body>

</html>
```

7.2 FEATURE 2

Predict.html

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

```
</head>

<style>
   body{
    background-image: url('C:/IBM Project/templates/image/num5.jpg');
    background-repeat: no-repeat;
    background-size: cover;
   }

   #rectangle{
    width:400px;
    height:150px;
    background-color: #5796a5;
    border-radius: 25px;
    position:absolute;
    top:25%;
    left:50%;
    transform:translate(-50%,-50%);
   }

   #ans{
  text-align: center;
  font-size: 40px;
  margin: 0 auto;
  padding: 3% 5%;
  padding-top: 15%;
```

```
  color: white;
    }

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

8.TESTING

8.1 TEST CASES

Importing the Required Libraries

```
import numpy
import matplotlib.pyplot as plt  from
keras.utils import np_utils  from
tensorflow.keras.datasets i mport mnist
from tensorflow.keras.models
import Sequential from tensorflow.keras.layers import Conv2D, Dense,
Flatten  from tensorflow.keras.optimizers import Adam  import numpy
as np  import pandas as pd  from tensorflow.keras.models
import load_model from PIL import Image, ImageOp
```

Loading the Data:

```
 (X_train, y_train), (X_test, y_test) = mnist.load_data()
```

Analyzing the Data:

```
print(X_train.shape)
print(X_test.shape)
(60000, 28, 28)
(10000, 28, 28)  X_train[0]
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]
```

8.2 User Acceptance Testing:

Unit Testing:

Unit testing verification efforts on the smallest unit of software design, module. This is known as “Module Testing”. The modules are tested separately. This testing is carried out during programming stage itself. In these testing steps, each module is found to be working satisfactorily as regard to the expected output from the module.

Integration Testing:

Integration testing is a systematic technique for constructing tests to uncover error associated within the interface. In the project, all the modules are combined and then the entire programmer is tested as a whole. In the integration-testing step, all the error uncovered is corrected for the next testing steps.

Validation Testing:

To uncover functional errors, that is, to check whether functional characteristics confirm to specification or not specified.

System Testing:

Once individual module testing completed, modules are assembled to perform as a system. Then the top down testing, which begins from upper level to lower level module testing, has to be done to check whether the entire system is performing satisfactorily.

9.RESULTS

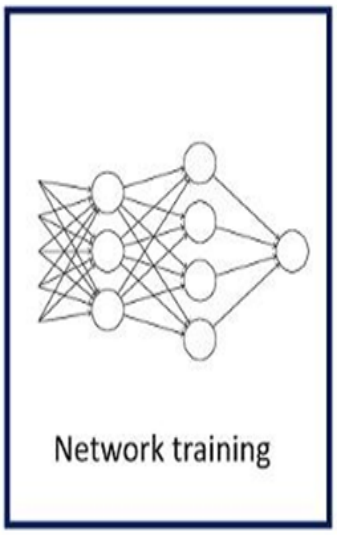
9.1 Performance Metrics:

Handwritten digit recognition is one of the important problems in computer vision these days. There is a great interest in this field because of many potential applications, most importantly where a large number of documents must be dealt such as post mail sorting, bank cheque analysis, handwritten form processing etc. So a system should be designed in such a way that it is capable of reading handwritten digits and providing appropriate results. We propose a solution on neural network approaches to recognize handwritten digits.

Classification:Convolutional neural network that is very popular for computer vision tasks like image classification, object detection, image segmentation and a lot more. Image classification is one of the most needed techniques in today's era, it is used in various domains like healthcare, business, and a lot more.

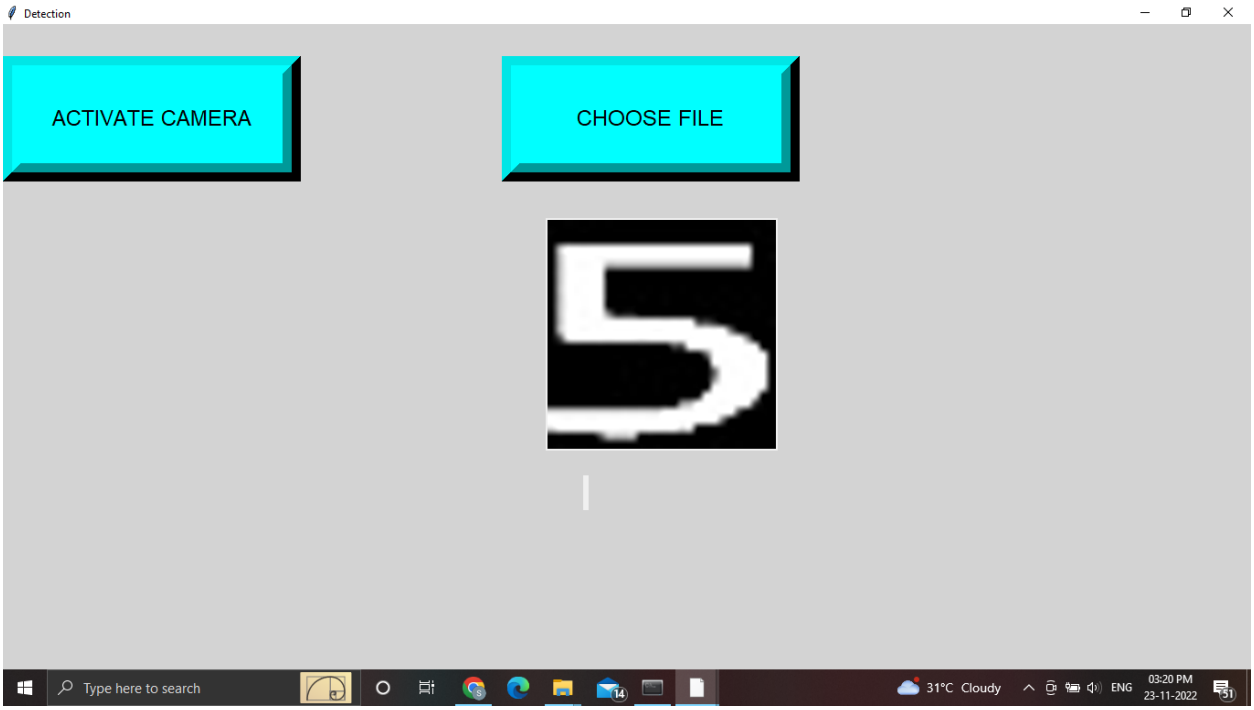
Tensor flow: TensorFlow is an open-source machine learning library for research and production. TensorFlow offers APIs for beginners and experts to develop for desktop, mobile, web, and cloud. See the sections below to get started. By scanning the numerical digit and converting it into png format using the python3 command in the terminal we can get text output and sound output.

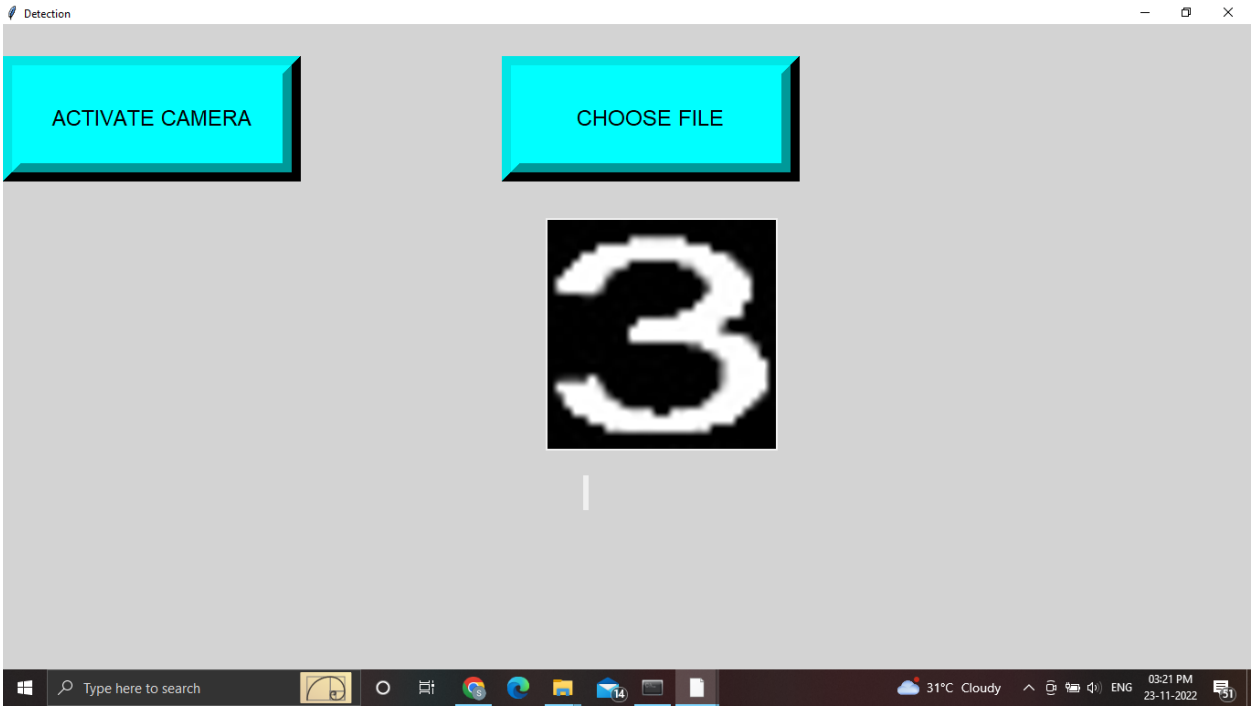
Data & Labels



0
1
2
3
4
5
6
7
8
9

OUTPUT:


Detection
ACTIVATE CAMERA
CHOOSE FILE
Type here to search
31°C Cloudy
ENG
03:20 PM
23-11-2022


Detection
ACTIVATE CAMERA
CHOOSE FILE
Type here to search
31°C Cloudy
ENG
03:21 PM
23-11-2022

10. ADVANTAGES & DISADVANTAGES

ADVANTAGES:

1. Image Segmentation:
   Image segmentation is the process of partitioning an image into multiple segments. Image classification is the process of predicting a specific class, or label, for something that is defined by a set of data points
2. Character Recognition:
   The handwritten character image is converted into printed text by sending the image to the model. In this model, each character is recognized after preprocessing and they are converted into readable text.
3. Less Complicated.
4. Easy to process
5. Accuracy is more
6. User friendly
7. Consistent Response and Availability

DISADVANTAGES:

1. Provides inappropriate result when handwritten image is not clear
2. Technology issues
3. Data should be some what clear for prediction

11.CONCLUSION

Handwriting recognition is very useful software that really helps to safe and keeps data and documents well. But at time it also has its disadvantage such as that sometimes if fails to read certain people's handwriting and due to this many people do not prefer to use the handwriting recognition software that much. Even though handwriting recognition has its disadvantages but still it is growing rapidly in the technology world. Handwriting recognition is used when there are certain people who prefer writing on the screens rather than writing it on a paper. In this project, Handwritten Digital Recognition is used In-depth learning strategies have been developed. Many widely used machine learning algorithms, RFC and CNN trained and tested on the same data into find comparisons between dividers. Use these are deeper learning methods, the higher the level of accuracy can be found. Compared to other research methods, this method focuses on which category works best for developing more than 99% separation accuracy models. Use Keras as backend and Tensorflow as software, CNN The model is able to provide about 98.72% accuracy. In the first one test, CNN provides 98.72% accuracy, while KNN provides 96.67% accuracy.

12.FUTURE SCOPE

Handwriting recognition is undoubtedly one of the most challenging areas of the pattern recognition. The goal of the project is to classify numeric samples which are mostly saved as digital images. Several pattern recognitions approaches have been applied to both online and off line handwriting recognition on the basis of unique patterns. The process of recognition consists of several steps such features extraction and recognition with voice alert. Python has a special toolbox, called neural network toolbox which makes the implementation less difficult but the knowledge of theory is needed. We can train these networks with preferred parameters. Artificial Neural Network approach for character recognition is now gaining importance because of CNN's high fault tolerance and parallel architecture. Comparing the performance of the proposed system with the eminent results of existing techniques, it is concluded that the new CNN based proposed system has achieved comparable performance to most of the existing techniques as well as it has exceeded the performance of others. Hence our results are comparable to state-of-the-art and this research will contribute positively to the research effort in the field of handwritten digit recognition. In future, other feature extraction methods and classification schemes will be considered for the digit recognition system.

13.APPENDIX

SOURCE CODE

```
import tensorflow as tf
tf.executing_eagerly()
#tf.enable_eager_execution()
from tensorflow.keras.preprocessing.image import ImageDataGenerator
train_datagen = ImageDataGenerator(rescale=1./255, width_shift_range=0.1,
height_shift_range=0.1)

# Load training data
train_generator = train_datagen.flow_from_directory('Data', target_size=(28,28),
batch_size=1, class_mode='categorical')

# Load validation data
validation_generator = train_datagen.flow_from_directory('Data', target_size=(28,28),
batch_size=1, class_mode='categorical')
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Flatten, MaxPooling2D, Dropout, Conv2D
from tensorflow.keras import optimizers
# CNN model
model = Sequential()
model.add(Conv2D(32, (5,5), input_shape=(28, 28, 3), activation='relu', padding='same'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.4))
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dense(36, activation='softmax'))
model.compile(loss='categorical_crossentropy',
optimizer=optimizers.Adam(lr=0.00001), metrics=['accuracy'])
# training the model
batch_size = 1
model.fit_generator(train_generator, validation_data = validation_generator, epochs =
10)
model.save("DEV.model")
```

GITHUB LINK: https: //github.com/IBM-EPBL/IBM-Project-21378-1659778933

DEMO VIDEO LINK:
https://drive.google.com/drive/u/0/folders/18myLvUvbELOdkWZKIyjOvw-3HBmwAdwj