A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM
PROJECT REPORT

Submitted by
Team ID: PNT2022TMID36178

SAKTHIVELAN.T	110519104022
SWATHI.H	110519104028
SUNIL KUMAR.S	110519104027
JOYCE SUDHANDHIRA.C	110519104011

In Partial Fulfillment for The Award of the Degree

OF

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

GOJAN SCHOOL OF BUSINESS AND TECHNOLOGY
REDHILLS.
ANNA UNIVERSITY: CHENNAI – 600 025

NOVEMBER-2022

TABLE OF CONTENT

PROJECT REPORT

A Novel Method for Handwritten Digit Recognition System

1.INTRODUCTION:

1.1 Project Overview

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

1.2 Purpose

1. To provide the ability of the machines to recognize human handwritten digits.
2. It will automatically recognize patterns and regularities in data.
3. It allows us to predict what will come next.
4. It basically detects the scanned images of handwritten digits.
5. It includes in postal mails sorting, bank check processing, form data entry, etc.
6. It is the ability of a computer to recognize the human handwritten digits from different sources like images, papers, touch screen etc. And classify them into ten predefined classes (0-9).

2. LITERATURE SURVEY:

2.1 Existing problem

- Handwritten recognition tends to have problem when it comes to accuracy. People can struggle to read others handwriting. The issue is that, there's a wide range of handwriting-good and bad.
- This make it tricky for programmers to provide enough example of how every character might look. Plus, sometimes characters look very similar, making it hard for a computer to recognise accurately.
- Joined-up handwriting is another challenge for computers. When your letters all connect, it makes it hard for computers to recognise the individual characters.

Consider, for instance an ‘r’ and an ‘n’. Joined-up, these letters could be mistaken for an ‘m’.

- In the case of handwriting recognition from photos, there are also awkward angles to consider. The angle the photo is taken could obscure the character, making it harder for the computer to identify.

2.2 References

[1] A Comparison of Feature and Pixel-based Methods for Recognizing Handwritten Bang la Digits. Olarik Surinta, Lambert Schomaker and Marco Wiering. Institute of Artificial Intelligence and Cognitive Engineering, University of Groningen Nijenborgh 9, Groningen, The Netherlands

[2] Handwritten Arabic Numeral Recognition using Deep Learning Neural Networks Akm Ashiquzzaman and Abdul Kawsar Tushar. Computer Science and Engineering Department, University of Asia Pacific, Dhaka, Bangladesh

[3] An Improved Feature Extraction Method for Individual Offline Handwritten Digit Recognition.Wang Qinghui1, Yang Aiping , and Dai Wenzhan. Department of Automatic Control, Zhejiang Sci-Tech University, Hangzhou, 310018.Zhejiang University of Finance & Economics Hangzhou, 310018. Department of Electron & Electric, Longyan University, Longyan.

[4] Moment-based Image Normalization for Handwritten Text Recognition. Michał Kozielski, Jens Forster, Hermann Ney Human Language Technology and Pattern Recognition Group, Chair of Computer Science 6 RWTH Aachen University, D-52056 Aachen, Germany.

[5] Combined mRMR-MLPSVM Scheme for High Accuracy and Low Cost Handwritten Digits Recognition. Mohammad Hassan Shammakhi, Electrical Engineering Department Amirkabir University of Technology Ali Mirzaei Electrical Engineering Department Amirkabir University of Technology. Parviz Khavari Electrical Engineering Department Amirkabir University of Technology. Vahid Pourahmadi Electrical Engineering Department Amirkabir University of Technology.

[6] Improved Handwritten Digit Recognition Using Convolutional Neural Networks (CNN) Year: 2020 Authors: Savita Ahlawat , Amit Choudhary , Anand Nayyar , Saurabh Singh and Byungun Yoon. Computer Science and Engineering Department.

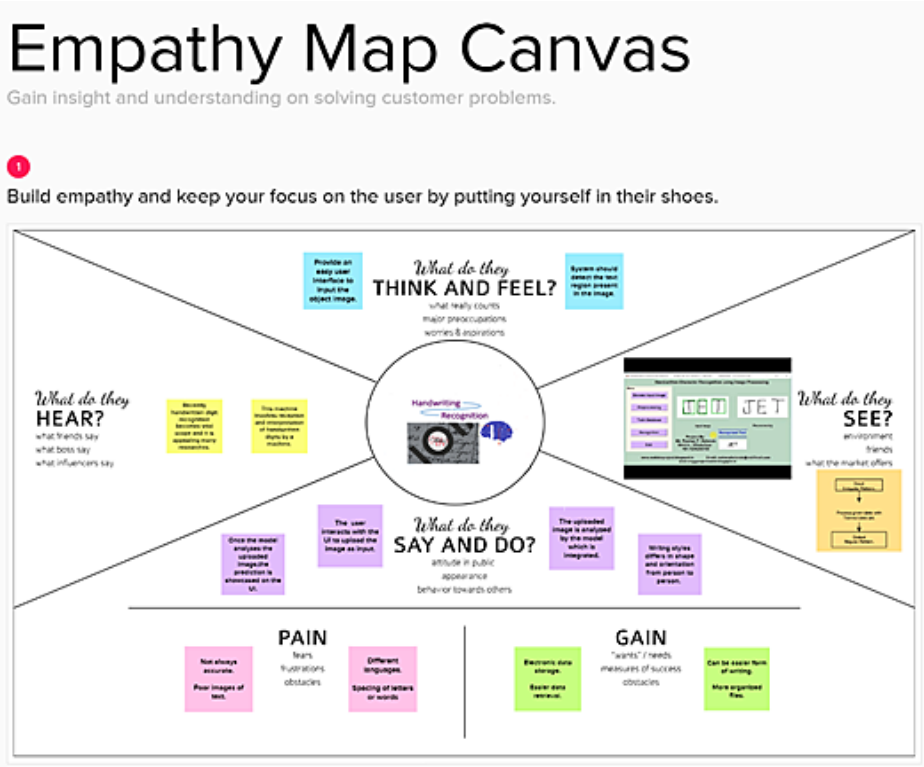
[7] A Novel Handwritten Digit Classification System Based on Convolutional Neural Network Approach Year: 2021 Authors: Ali Abdullah Yahya, Jieqing Tan, Min Hu. Electrical and Electronic Engineering Department.

2.3 Problem Statement Definition

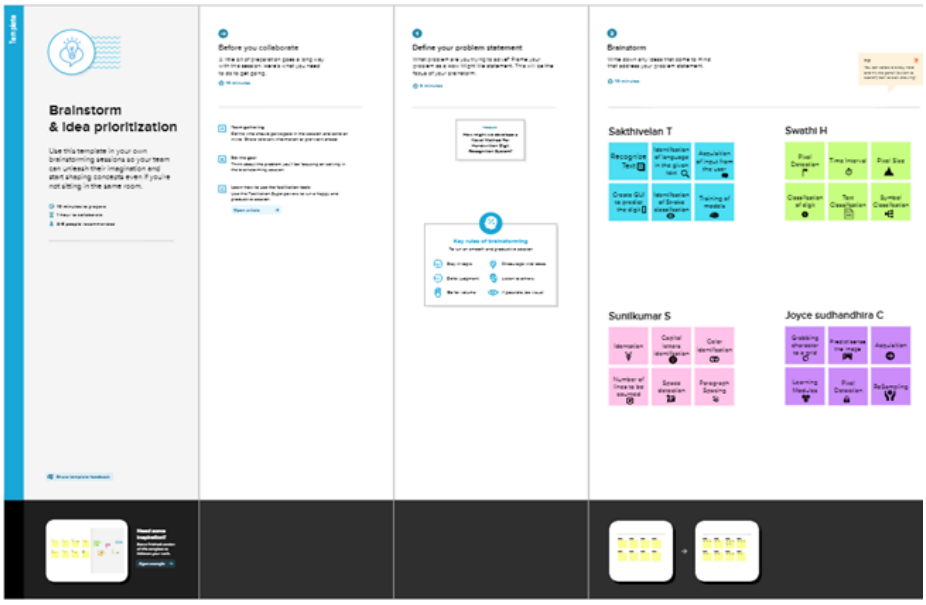
Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	Professor	Convert The Handwritten Notes Into Digital Text.	When written by hand, digits are not necessarily always of typically the same size, thickness, orientation and validated to margins	They differ coming from writing of personal to individual.	Exasperate
PS-2	Student.	Convert the Handwritten Assignment into Digital Text.	While converting, it change in text size, alignment and errors.	They differ with other individuals Text.	Thwarting

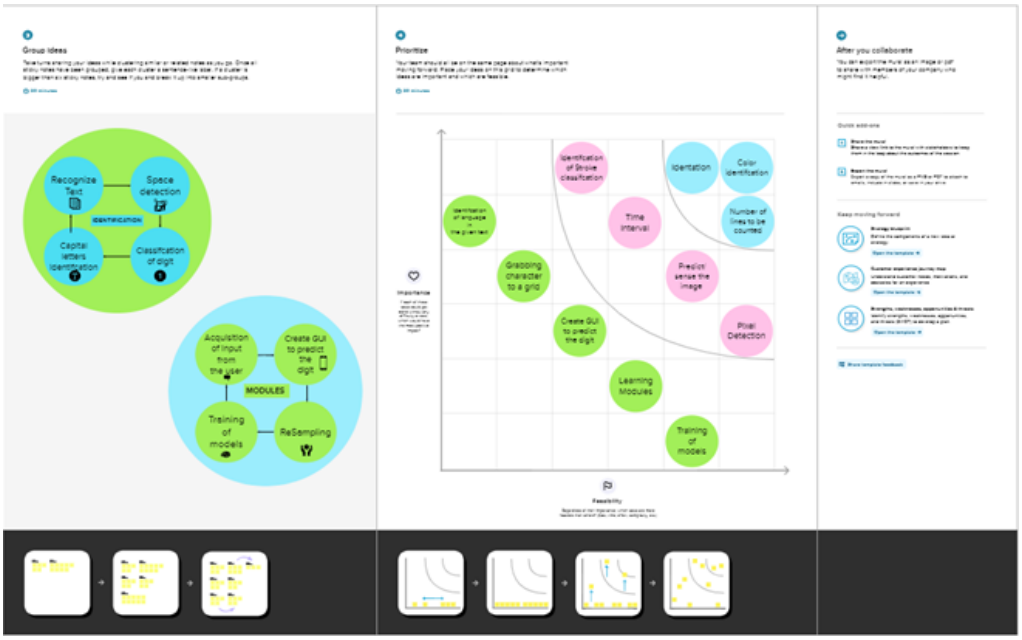
3. IDEATION & PROPOSED SOLUTION:

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming



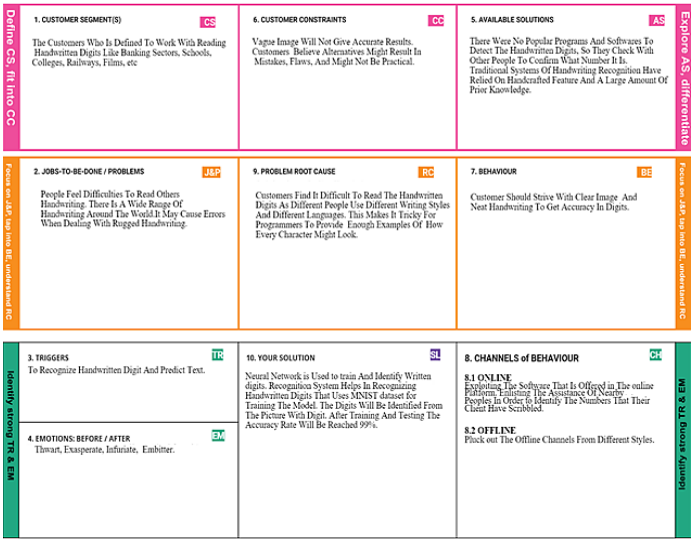


3.3 Proposed Solution

S.No	Parameter	Description
1.	Problem Statement (Problem to be solved)	**Statement**-The handwritten digit recognition is the capability of computer applications to recognize the human handwritten digits. **Description**: It is a hard task for the machine because handwritten digits are not perfect and can be made with many different shapes and sizes.
2.	Idea / Solution description	-It is the capability of a computer to fete the mortal handwritten integers from different sources like images, papers, touch defences. -It allows user to translate all those signature and notes into electronic words in a text document format and this data only requires far less physical space than the storage of the physical copies.

3.	Novelty / Uniqueness	Accurately recognize the digits rather than recognizing all the characters like OCR.
4.	Social Impact / Customer Satisfaction	1. Artificial Intelligence developed the app called Handwritten Digit Recognizer. 2. It converts the written word into digital approximations and utilizes complex algorithms to identify characters before churning out a digital approximation.
5.	Business Model (Revenue Model)	**1**.This system can be integrated with traffic surveillance cameras to recognize the vehicle's number plates for effective traffic management. **2**.Can be integrated with Postal system to identify and recognize the pin-code details easily.
6.	Scalability of the Solution	**1**.Ability to recognise digits in more noisy environments. **2**.There is no limit in the number of digits it can be recognized.

3.**4 Problem Solution fit**



4. REQUIREMENT ANALYSIS:

4.1 Functional requirements

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	**UserRegistration**.	Handwritten digit recognition refers to a computer's capacity to identify human handwritten digits from a variety 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	**UserConfirmation**.	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		**DigitClassifierModel**: To train a convolutional network to predict the digit from an 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 virtual storage, 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.

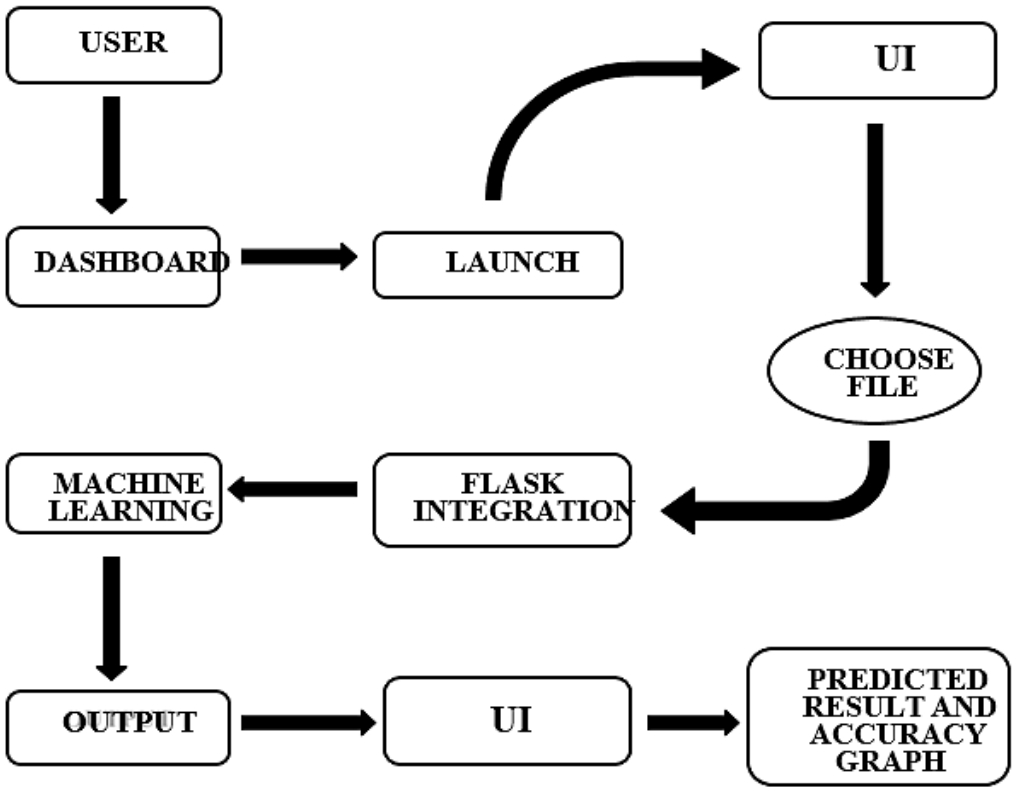
4.2 Non-Functional requirements

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	One of the very significant 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**	The system generates a thorough description of the instantiation parameters, which might reveal information like the writing style, in addition to a categorization of the digit. The generative models are capable of segmentation driven by recognition.
NFR-3	**Reliability**	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 Deep Learning/CNN, SVM, Gaussian Naive Bayes, KNN, Decision Trees, Random Forests, etc., can be used to recognise handwritten numbers.
NFR-4	**Performance**	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 less precise character identification
NFR-5	**Accuracy**	These features are based on shape analysis of the digit image and extract slant or slope information. They are effective in obtaining good recognition of accuracy

5. PROJECT DESIGN:

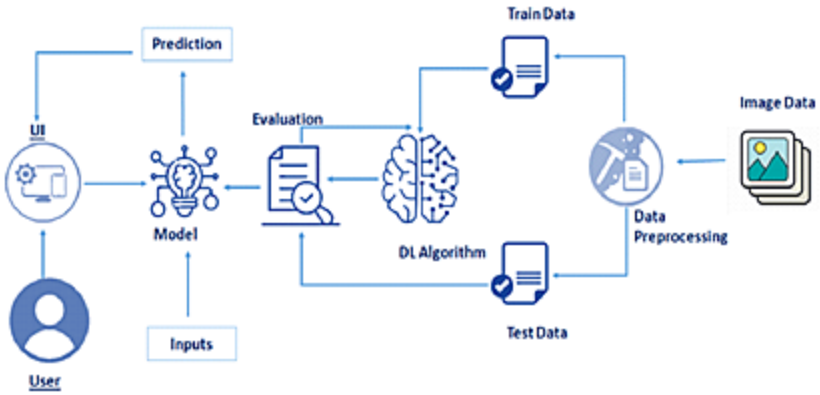
5.1 Data Flow Diagrams

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

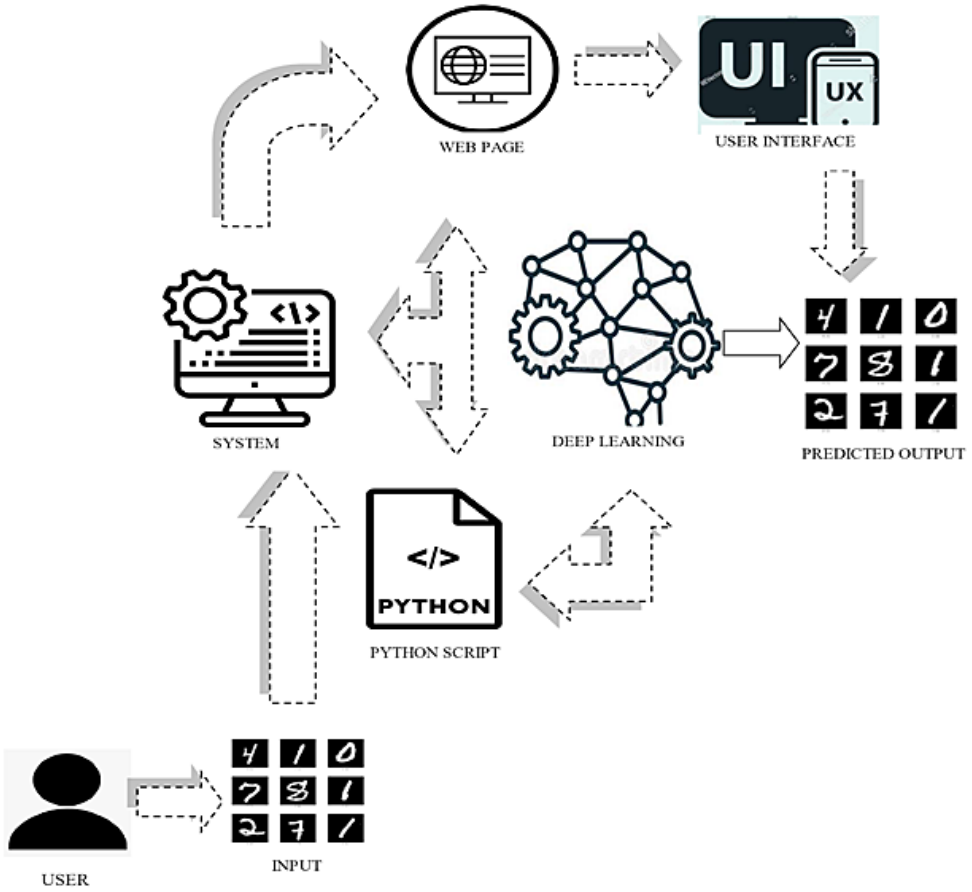


5.2 Solution & Technical Architecture

Solution Architecture:



Technology Architecture:



5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer	Dashboard	USN-1	As a user, they can see the information regarding the prediction of handwritten digit recognition.	I can see the information regarding digit recognition.	High	Sprint 1
	Launch	USN-2	On clicking the launch button, it will redirect the user to a page where the images to be predicted can be uploaded.	I can see the launch button.	High	Sprint 1
	Upload	USN-3	Users can select the image from the local storage.	I can upload the image.	High	Sprint 2
	Predict	USN-4	Once the image is uploaded, it will predict the respective image.		High	Sprint 3
	Display	USN-5	The predicted image will be displayed with the accuracy chart	I can see the result with accuracy.	High	Sprint 4

6. PROJECT PLANNING & SCHEDULING:

6.1 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.
Sprint-1	Login	USN-2	As a user, I can log into the application by entering email & password.
Sprint-2	Upload Image of digital document	USN-3	As a user, I can able to input the images of digital documents to the application.
Sprint-2	Prediction	USN-4	As a user, I can predict the word.
Sprint-3	Upload Image of Handwritten document	USN-5	As a user, I can able to input the images of the handwritten documents or images to the application.
Sprint-3	Recognize text	USN-6	As a user, I can able to choose the font of the text to be displayed.

Sprint-4	Recognize digit	USN-7	As a user I can able to get the recognised digit as output from the images of digital documents or images.
Sprint-4	Recognize digit	USN-8	As a user I can able to get the recognised digit as output from the images of handwritten documents or images.

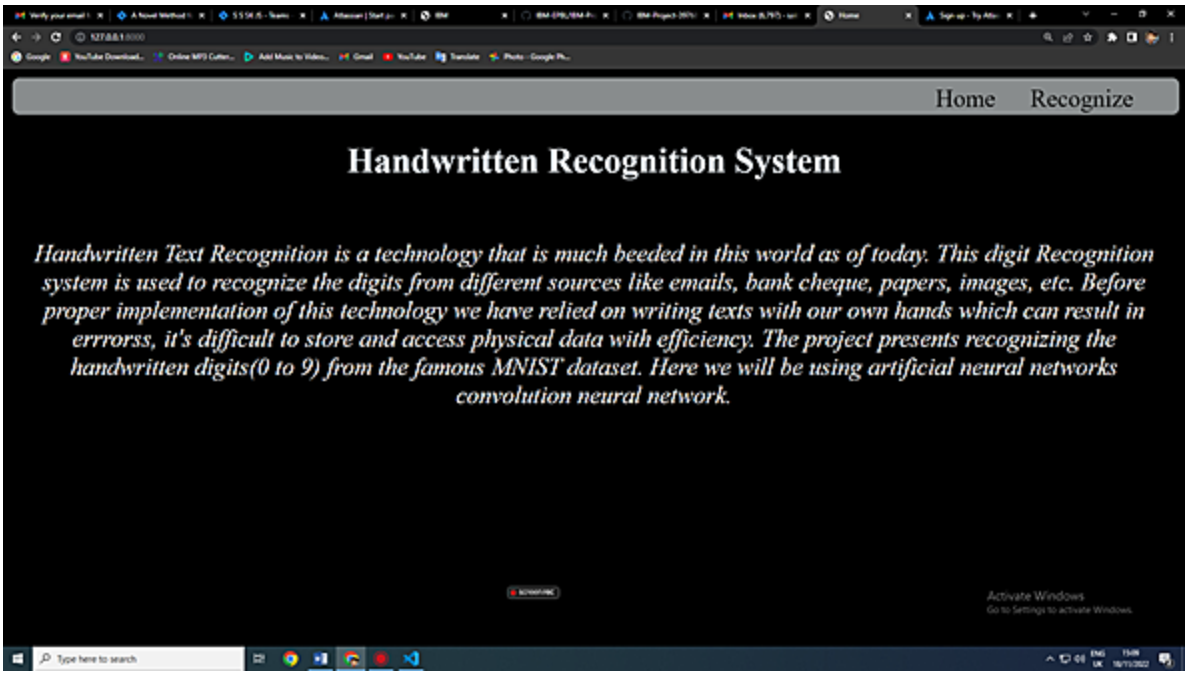
6.2 Sprint Delivery Schedule

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Sprint Release Date (Actual)
Sprint-1	2	6 Days	24 Oct 2022	29 Oct 2022	29 Oct 2022
Sprint-2	2	6 Days	31 Oct 2022	05 Nov 2022	05 Nov 2022
Sprint-3	2	6 Days	07 Nov 2022	12 Nov 2022	12 Nov 2022
Sprint-4	2	6 Days	09 Nov 2022	18 Nov 2022	15 Nov 2022

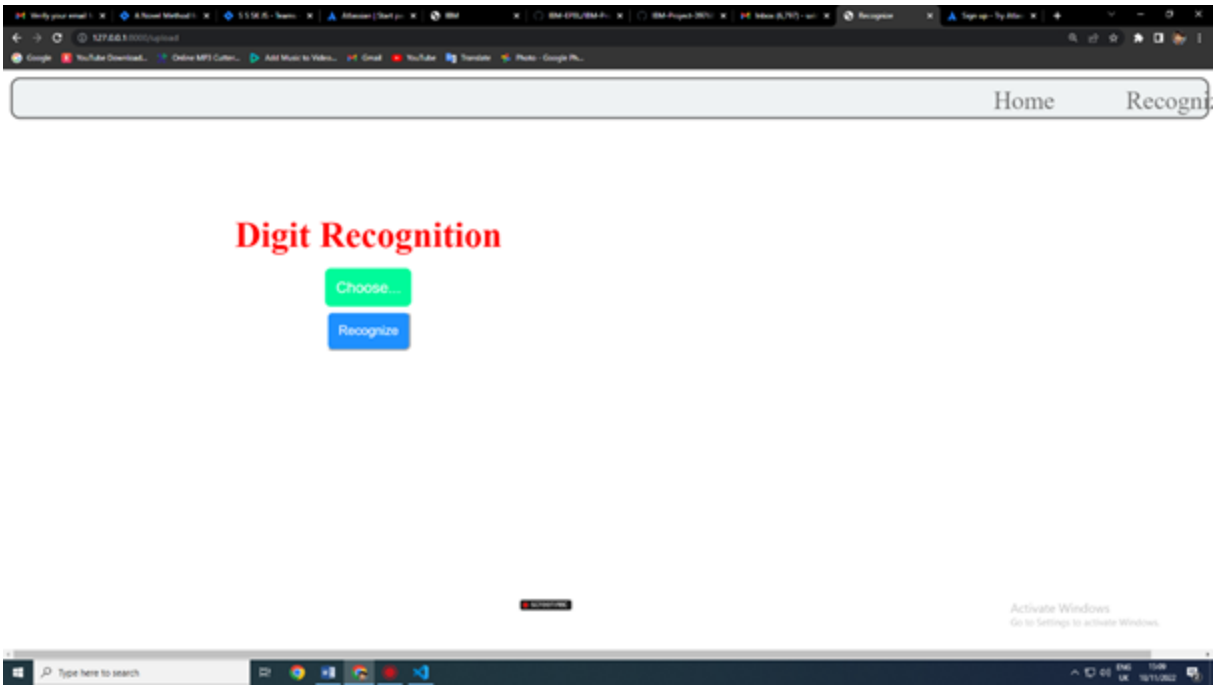
6.3 Reports from JIRA

7. CODING & SOLUTIONING:

7.1 Feature 1:



7.2 Feature 2:



8. TESTING:

8.1 Test Cases

Thisreport shows the number of test cases that have passed, failed,and untested.

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	6	0	0	6
Client Application	51	0	2	49
Security	2	0	0	2
Outsource Shipping	3	0	0	3
Exception Reporting	9	0	1	8
Final ReportOutput	4	0	0	4
Version Control	2	0	0	2

8.2 User Acceptance Testing

The purpose of this document is to briefly explain the test coverageand open issuesof the [Product Name] project at the time of the release to User Acceptance Testing (UAT).

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

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	3	1	2	16
Duplicate	1	0	2	0	3
External	1	2	0	1	4
Fixed	10	1	3	12	26
Not Reproduced	0	0	1	0	1
Skipped	0	0	0	1	1
Won't Fix	0	0	1	1	2

9. RESULTS:

9.1 Performance Metrics

We here are printing the metrics which lists out the Test loss and Test accuracy

- Loss value implies how poorly or well a model behaves after each iteration of optimization.
- An accuracy metric is used to measure the algorithm's performance in an interpretable way.

10. ADVANTAGES & DISADVANTAGES:

ADVANTAGES

- Electronic data storage.
- More organized files.
- Easier data retrieval.
- Historical preservation.
- Can be easier form of writing.
- Verification methods.

DISADVANTAGES

- Not always accurate.
- Unique style of writing.
- Spacing of letters or words.
- Poor images of text.
- Different languages.
- Modern handwriting compared to historical.

11. CONCLUSION:

This paper proposes a novel algorithm for recognizing in handwritten with the help of CNN. Our proposed CNN model uses several convolutional layers, with dropout used as a regularization Layer. Then the output is fed into a fully connected layer with activation to obtain prediction for each class. Our proposed novel method achieves better accuracy. We showed that the use of moments improves significantly the recognition performance in handwriting recognition and outperforms other pre-processing approaches. The proposed scheme is also evaluated by using MNIST datasets which leads to 96.1 and 98.14 accuracy respectively. It obtained 96.8% accuracy with 90% of the training data.

12. FUTURE SCOPE:

- Make the system more font independent.
- Increase the number of nodes and layers in ANN.
- Try different recognition algorithm such HMM (Hidden Markov Model).
- Improve the separation of touching characters.
- Improve the efficiency of the feature extraction method.
- Improve the system to identify any other characters.

13. APPENDIX:

Source Code:

Flask:

App.py

```
from flask import Flask, request, render_template
from PIL import Image
import numpy as np
from requests import post
import tensorflow as tf
from tensorflow import keras
from keras.models import load_model
from keras.utils import np_utils
import matplotlib.pyplot as plt
from io import BytesIO
import base64
```

```
app = Flask('__main__')

model = load_model('models/mnistCNN.h5')


def generate(y_pred):
    n_o_c = 10
    y_pred = np_utils.to_categorical(y_pred,n_o_c)
    x = np.array([0,1,2,3,4,5,6,7,8,9])
    y = y_pred.astype(int)
    y = y[0]
    ch = np.where(y ==1)
    y[ch] = 10
    data = sub(x,y)
    return data

def sub(x,y):
    plt.bar(x,y, color = 'red')
    b = BytesIO()
    plt.savefig(b, format='png')
    data = base64.b64encode(b.getbuffer()).decode()
    return data

def show(img):
    img = img
    data = base64.b64encode(img.getbuffer()).decode()
    return data

@app.route('/')
def home():
    return render_template('index.html')

@app.route('/upload')
def upload():
    plt.clf()
```

```
    return render_template('web.html',show = 'hidden')

@app.route('/predict', methods = ['POST', 'GET'])
def predict():
    if request.method =='POST':
        img = request.files['file']
        data = show(img)
        img = Image.open(img.stream).convert('L')
        img = img.resize((28,28))
        im2arr = np.array(img)
        im2arr = im2arr.reshape(1,28,28,1)
        y_pred = model.predict(im2arr)
        y_pred = np.argmax(y_pred,axis=1)
        pred = str(y_pred)
        output = "Recognized digit is :"+ pred
        bar = generate(y_pred)
        return render_template('web.html',output = output,bar = bar,data = data,button =
'hidden' )

if __name__ == '__main__':
    app.run(host='0.0.0.0',port=8000,debug=True)
```

Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Home</title>
    <link rel="stylesheet" href="{{url_for('static',filename = 'style.css')}}">
</head>
<body>
    <div id="label"><div id="label2"><span><a href="">Home</a></span><span><a
href="/upload">Recognize</a></label></span></div></div>
```

```
   <div><h1>Handwritten Recognition System</h1></div>
   <div><p>Handwritten Text Recognition is a technology that is much beeded in this
world as of today. This digit Recognition system
      is used to recognize the digits from different sources like emails, bank cheque,
papers, images, etc. Before proper
      implementation of this technology we have relied on writing texts with our own
hands which can result in errrorss, it's
      difficult to store and access physical data with efficiency. The project presents
recognizing the handwritten digits(0 to 9)
      from the famous MNIST dataset. Here we will be using artificial neural networks
convolution neural network.</p></div>

</body>
</html>
```

Web.html

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <link rel="stylesheet" href="{{ url_for('static', filename = 'style2.css') }}">
   <title>Recognize</title>
   <title>Recognize</title>
</head>
<body>
   <div id="label"><div id="label2"><span><a href="/">Home</a></span><span><a
href="/upload">Recognize</a></label></span></div></div>
   <div id="parent">
      <div id="p1">
         <h1>Digit Recognition</h1>
         <form action="/predict" method="post" enctype="multipart/form-data">
            <label id="imglb" for="file">Choose...</label>
            <input type="file" name="file" id="file" hidden><br>
```

```
            <input type="submit" id="submit" name="submit"
value="Recognize"{{button}}><br>
            <img src="{{ data}}" id="imgpic" alt=""
srcset=""{{show}}>
        </form>
      </div>
      <div id="child">
        {{ output }}
        <img src="{{ bar }}" id="img_show" alt=""
srcset=""{{show}}>
      </div>
  </div>
</body>
</html>
```

CSS:

Style.css

```
span{
  background-color: ivory;
}
a{
  color: black;
  text-decoration-line: none;
}
#label{
  opacity: 0.6;
  height: 40px;
  width: 100%;
  border: 2px solid ;
  border-radius: 10px ;
  background-color: rgb(227, 234, 235);
}
#label2{
  padding-top: .2cm;
  /*padding-right: ;*/
  padding-bottom: .5cm;
  padding-left: 1000px;
```

```
    font-size: .7cm;
}
span{
    padding-right: 1cm;
    background-color: transparent;
}
body{
    background-color: black;
}
h1{
    text-align: center;
    color: aliceblue;
    font-size: 1cm;
}
p{
    padding-top: 1cm;
    font-size: .7cm;
    text-align: center;
    color: white;
    font-style: italic;
}
```

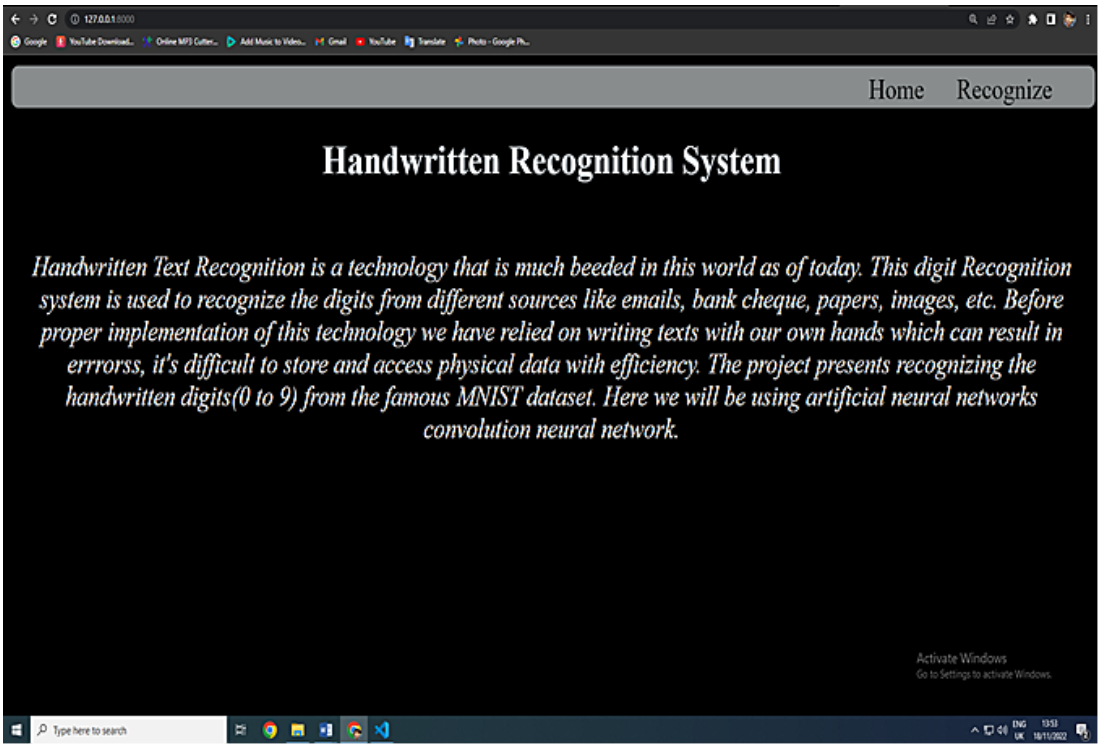
Style1.css

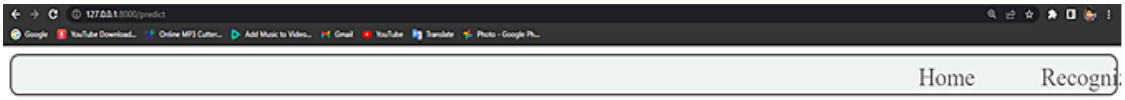
```
#label{
    opacity: 0.6;
    height: 40px;
    width: 100%;
    border: 2px solid ;
    border-radius: 10px ;
    background-color: rgb(227, 234, 235);
}
#label2{
    padding-top: .2cm;
    /*padding-right: ;*/
    padding-bottom: .5cm;
    padding-left: 1000px;
    font-size: .7cm;
```

```
}
#p1{
    text-align: center;
    margin-top: 2cm;
    margin-left: 5cm;
    color: black;
    background-color: white;
    display: inline-block;
    height: auto;
    width: 30%;
}
#child{
    text-align: center;
    margin-top: 2cm;
    margin-left: 5cm;
    color: black;
    background-color: white;
    display: inline-block;
    height: auto;
    width: 30%;

}
a{
    color: black;
    margin: 1cm;
    text-decoration: none;

}
a.hover{
    color: grey;
}
#imglb{
    background-color: mediumspringgreen;
    color: white;
    padding: 0.3cm;
    font-family: sans-serif;
```

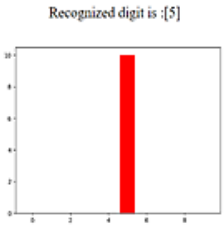
```
    text-align: center;
    border-radius: 0.4rem;
    cursor: pointer;
    margin: .4cm;
}
h1{
    color: red;
    font-size: 1cm;
}
#submit{
    background-color: dodgerblue;
    color: white;
    padding: 0.3cm;
    border-color: white;
    border-radius: 0.4rem;
    cursor: pointer;
    margin: .4cm;
}
#img_show{
    height: 6cm;
    width: 8cm;
}
#imgpic{
    height: 3cm;
    width: 3cm;

}
```

OUTPUT:


Home
Recognize
Handwritten Recognition System
Handwritten Text Recognition is a technology that is much beeded in this world as of today. This digit Recognition system is used to recognize the digits from different sources like emails, bank cheque, papers, images, etc. Before proper implementation of this technology we have relied on writing texts with our own hands which can result in errrorss, it's difficult to store and access physical data with efficiency. The project presents recognizing the handwritten digits(0 to 9) from the famous MNIST dataset. Here we will be using artificial neural networks convolution neural network.
Activate Windows
Type here to search


Home
Recogni


Digit Recognition
Choose...


Recognized digit is :[5]


Activate Windows
Type here to search

GitHub & Project Demo Link

GitHub link :- https://github.com/IBM-EPBL/IBM-Project-39768-1660533378

Project_Demo_Link:- https://github.com/IBM-EPBL/IBM-Project-39768-1660533378/blob/main/Final%20Deliverables/A%20Novel%20Method%20For%20Handwritten%20Digit%20Recognition%20System/Project%20Demo.mp4

Youtube link -