

IBM PROJECT

A NOVEL METHOD FOR HANDWRITTEN DIGIT
RECOGNITION SYSTEM

SI NO	Table of Content
1	INTRODUCTION
1.1	PROJECT OVERVIEW
1.2	PURPOSE
2	LITERATURE SURVEY
2.1	EXISTING PROBLEM
2.2	REFERENCE
2.3	PROBLEM STATEMENT DEFINITION
3	IDEATION & PROPOSED SOLUTION
3.1	EMPATHY MAP CANVAS
3.2	IDEATION & BRAINSTROMING
3.3	PROPOSED SOLUTION
3.4	PROBLEM SOLUTION FIT
4	REQUIREMENT ANALYSIS
4.1	FUNCTIONAL REQUIREMENTS
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	REPORT FROM JIRA
7	CODING & SOLUTION
7.1	FEATURE 1
7.2	FEATURE 2
7.3	DATABASE SCHEMA

8	TESTING
9	RESULT
10	ADVANTAGE & DISADVANTAGE
11	CONCLUSION
12	FUTURE SCOPE
13	APPENDIX
13.1	SOURCE CODE
13.2	GITHUB & PROJECT DEMO LINK

INTRODUCTION

Presently, many people use images to depict and transfer information. It is also popular to extract important features from an image that provide valuable information. Image recognition is a growing domain in that field of research because of its wide variety of applications. In the comparatively young and emerging field of computer pattern recognition, one of the challenges lies in the accurate automated recognition of human handwriting. This is quite a challenging task taking into consideration the difference in handwriting from one person to another. Although, this variance in handwriting does not cause much problems to humans, it is tricky and quite a comprehensive task in teaching a generic handwriting to a computer. For the problem of handwritten digit classification through image pattern recognition, it is very important to analyse how information is represented in images. The image pixels are not the ones that contain information, but it is the features of images which have high level illustration. In the case of handwritten digit recognition, features relating to the structure of the digits should be first extracted from the style in which the digits are written. These features which are extracted can be used to recognize the handwritten digit. Most of the classification and regression machine learning algorithms that are present currently are shallow learning algorithms. It is difficult to represent complex functions effectively, and ability to generalize them is limited for the complicated classification problems.

Deep learning is a recently discovered algorithm consisting of a multilayer network of neurons. Applications of deep learning to a number of problems has brought a substantial difference in the number of recent studies ranging from classification of images and recognition of speech to classification of audio. It has changed the realm of machine learning, and making huge advancements in artificial intelligence. Deep Learning algorithms are best used because of their high efficiency in image recognition tasks, in this case, MNIST digit recognition.

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 digitalized 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 analyzed by the model and the detected result is returned on to UI

Purpose

Handwritten digit recognition using MNIST dataset is a major project made with the help of Neural Network. The core purpose of the Handwritten Digit Recognition is to basically detects the scanned images of handwritten digits and convert it to a relevant digital digit. This methodology gives the power to the number plate detection etc.

LITERATURE SURVEY

[1] M. M. Abu Ghosh and A. Y. Maghari

This paper shows the Approaches through Neural Networks . The three famous Neural Network approaches are deep neural network (DNN), deep belief network (DBN) and convolutional neural network (CNN). The three approaches are compared based on their accuracy and performance. Random and standard dataset of handwritten digits have been used for conducting the experiments. The results show that among the 3 NN approaches, DNN is the most accurate algorithm as it has an accuracy rate of 98.08%. However, the execution time of DNN is comparable with the two other algorithms. Every algorithm has an error rate of 1-2% because of the similarity in digit shapes, specially, with the digits (1,7), (3,5), (3,8), (8,5) and (6,9).

[2] Jürgen Schmidhuber

This paper reviews deep supervised learning (also recapitulating the history of backpropagation), unsupervised learning, reinforcement learning & evolutionary computation, and developing short programs encoding deep and large networks.

[3] Han, X. and Y. Li.

This paper explains about Convolutional neural networks which combines artificial neural networks and recent deep learning methods. This paper summarizes the latest development of convolutional neural networks and expounds the relative analysis of image recognition technology and explains how convolutional neural networks can be used in handwritten numeral recognition.

[4] Ciresan, D. et al

This paper explains how Small (often minimal) receptive fields of convolutional winner-take-all neurons gain massive network depth, leading to several sparsely connected neural layers. Many deep neural columns become well trained on inputs preprocessed in various ways; their predictions are averaged. On this sophisticated MNIST handwriting benchmark, the proposed method is the very first to achieve near-human performance. The paper has mentions to improve on a plethora of common image classification benchmarks.

[5] Niu n Xiao-Xiao and Y. Suen Ching

This paper describes the method involved to classify Arabic Handwritten Digits as there are deficient works accomplished on Arabic pattern digits because Arabic digits are more challenging than English patterns. Hence, the lack of research using Arabic digits endeavors the authors to work deeper by creating a challenge of Arabic Handwritten Digits recognition, which consists of more than 45,000 samples. As a challenging dataset is used for evaluation, a robust deep convolutional neural network is used for classification and superior results are achieved.

[6] Ahlawat S, Choudhary A, Nayyar A, Singh S, Yoon B

The aim of this paper is to find choices in Model design parameters like number of layers, stride size, receptive field, kernel size, padding and dilution for CNN-based handwritten digit recognition. Additionally, the paper aims to evaluate various SGD optimization algorithms to enhance the performance of handwritten digit recognition. The objective is to achieve comparable accuracy by using a pure CNN architecture without ensemble architecture, as ensemble architectures raises computational cost and features greater testing complexity. Thus, a CNN architecture is proposed to attain accuracy better than that of ensemble architectures that offers reduced operational complexity and cost. An appropriate combination of learning parameters is presented in designing a CNN that achieved 99.87% recognition accuracy rate on MNIST dataset.

[7] Fabien Lauer, Ching Y. Suen, Gérard Bloch

This article focuses on the problems of extraction of features and handwritten digit recognition. A feature tractor model which is trainable and based on the LeNet5 convolutional neural network architecture is introduced to solve the problem of feature extraction in a high level manner without much previous knowledge on the data being used. The classification task is performed by SVM to enhance the ability with which LeNet5 is able to generalize. In order to enhance the rate at which it recognizes, new samples for the purpose of training are generated subject to various transformations and distortions. Various experiments conducted on the standard MNIST dataset show that the approach used can perform better than both SVMs and LeNet5 while providing performances reminiscent of the simplest performance on the given dataset.

[8] L. Bottou et al

This paper compares the performance of several classifier algorithms on a customary set of handwritten digits. The paper not only considers the accuracy obtained, but also time taken to train, time consumed for recognition, the amount of memory required and carefully analyzing the patterns that should be rejected so that the rate at which the classifications of the remaining patterns made are false are less than a well defined threshold.

Existing problem

Mr.R.Anandaraj is 27 years old, who is working in the State bank of India, Hyderabad. His work is to collect the Handwritten Cheque from the customers and fill it in the bank database. For the last 2 years, the usage of Cheque got higher, and understanding the customer's handwriting takes much time which made his work harder.

1. Mr.R.Anandaraj needed much more time to complete a small set of cheque entry.

2. He thinks this problem will make his work uncertain, and someone will be replaced instead of him.

He wants to know a better solution for the this problem.

Problem Statement Definition

Who does the problem affect?	Person who does data entry in bank.
What are the boundaries of the problem?	Person who works in bank and does data entry.
What is the issue?	The understanding of customer's handwriting takes much time, so there comes the delay in data entry.
When does the issue occur?	During the peak hours of the bank.
Where does the issue occur?	In the places, where people does data entry from the handwritten data sheet or document.
Why is it important that we fix the problem?	To reduce the excess time taken for the understanding the customer's handwriting.
What is the solution for this issue?	An automated system is introduced to recognize the digit in the handwritten paper which reduce the excess time taken for the understanding the customer's handwriting.
What methodology used to solve the issue?	Deep learning techniques are used to recognize the digits and display it correctly to the users.

REFERENCES

- [1] M. M. Abu Ghosh and A. Y. Maghari, "A Comparative Study on Handwriting Digit Recognition Using Neural Networks," 2017 International Conference on Promising Electronic Technologies (ICPET), 2017, pp. 77-81, doi: 10.1109/ICPET.2017.20.
- [2] Jürgen Schmidhuber, "Deep learning in neural networks: An overview, Neural Networks", Volume 61, 2015, Pages 85-117, ISSN 0893-6080, <https://doi.org/10.1016/j.neunet.2014.09.003>.
- [3] Han, X. and Y. Li. "The Application of Convolution Neural Networks in Handwritten

Numeral Recognition.” International journal of database theory and application 8 (2015): 367-376.

[4] Ciresan, D. et al. “Multi-column deep neural networks for image classification.” 2012 IEEE Conference on Computer Vision and Pattern Recognition (2012): 3642-3649.

[5] Niu n Xiao-Xiao and Y. Suen Ching, "A novel hybrid CNN-SVM classifier for recognizing handwritten digits", Elsevier, vol. 45, no. 4, pp. 1318-1325, April 2012.

[6] Ahlawat S, Choudhary A, Nayyar A, Singh S, Yoon B. Improved Handwritten Digit Recognition Using Convolutional Neural Networks (CNN). Sensors. 2020; 20(12):3344. <https://doi.org/10.3390/s20123344>

[7] Fabien Lauer, Ching Y. Suen, Gérard Bloch, A trainable feature extractor for handwritten digit recognition, Pattern Recognition, Volume 40, Issue 6, 2007, Pages 1816-1824, ISSN 0031-3203, <https://doi.org/10.1016/j.patcog.2006.10.011>.

[8] L. Bottou et al., "Comparison of classifier methods: a case study in handwritten digit recognition," Proceedings of the 12th IAPR International Conference on Pattern Recognition, Vol. 3 - Conference C: Signal Processing (Cat. No.94CH3440-5), 1994, pp. 77-82 vol.2, doi: 10.1109/ICPR.1994.576879.

IDEATION & PROPOSED SOLUTION

Empathy Map Canvas



Ideation & Brainstorming

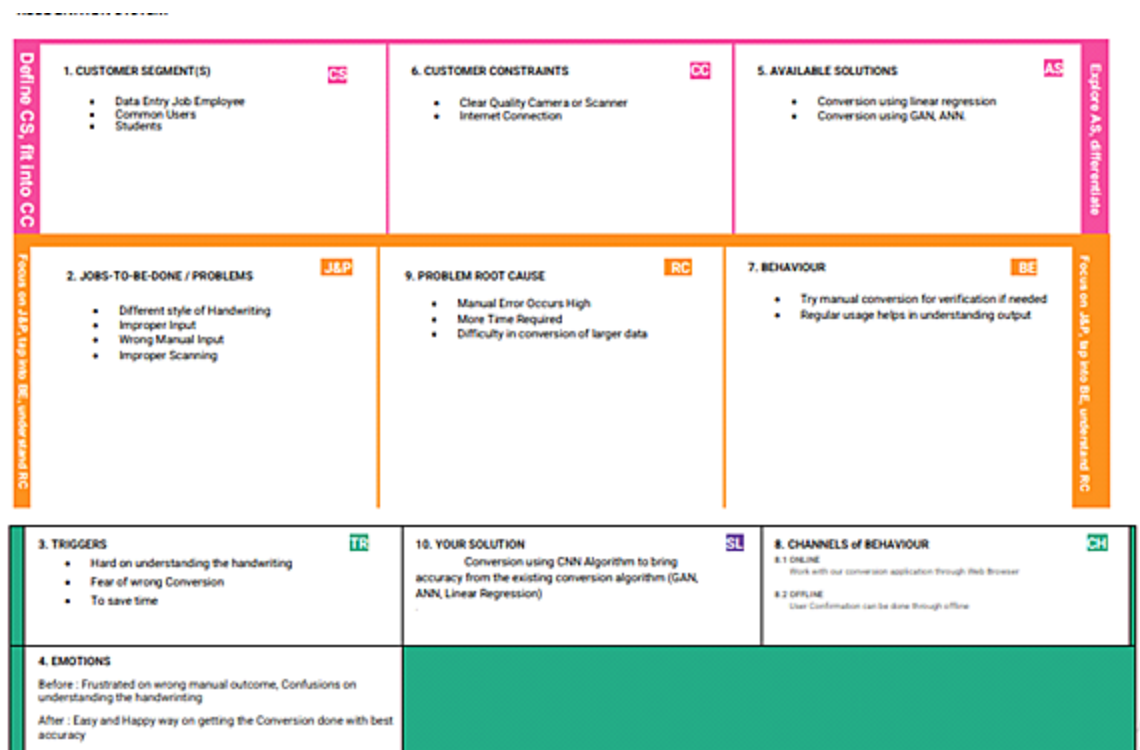


Proposed Solution

Project team shall fill the following information in proposed solution template.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Conversion of Hand Written raw Document Data (Digits) into equivalent Digital Data.
2.	Idea / Solution description	Recognizing the Hand Written Data and Converting it using CNN Algorithm
3.	Novelty / Uniqueness	Bring up more accuracy with CNN when compared to ANN or GAN Algorithm
4.	Social Impact / Customer Satisfaction	Easy and Accurate results with the CNN Algorithm rather than wasting time on manual conversion
5.	Business Model (Revenue Model)	Free of Cost Model

Problem Solution fit



REQUIREMENT ANALYSIS

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Sub Requirement (Story / Sub-Task)
FR-1	ImageData: 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 categorise 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 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.
FR-5	Modified National Institute of Standards and Technology dataset: The abbreviation MNIST stands for the MNIST dataset. It is a collection of 60,000 tiny square grayscale photographs, each measuring 28 by 28, comprising handwritten single digits between 0 and 9.

Non-functional Requirements:

Following are the non-functional requirements of the proposed solution.

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	1) 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. 1. The generative models are capable of segmentation driven by recognition. 2. The procedure uses a relatively.
NFR-3	Reliability	The samples are used by the neural network to automatically deduce rules for reading handwritten digits. 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.

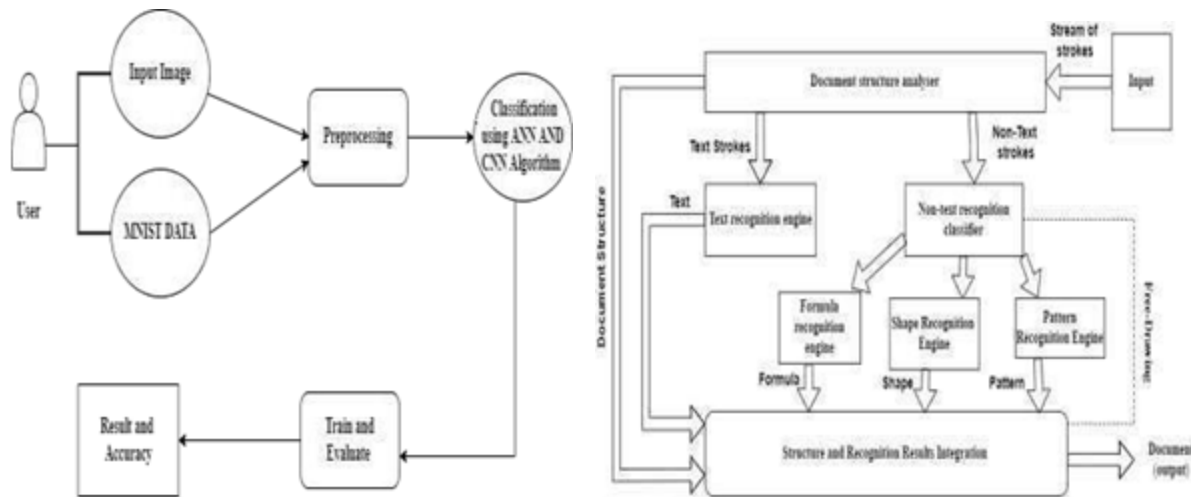
NFR-4	Accuracy	With typed text in high-quality photos, optical character recognition (OCR) technology offers accuracy rates of greater than 99%. However, variances in spacing, abnormalities in handwriting, and the variety of human writing styles result in less precise character identification.
-------	-----------------	---

PROJECT DESIGN

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.

Example: (Simplified) FLOW



User Type	Functional Requirement (Epic)	User Story Number	User Story/ Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Home	USN-1	As a user, I can view the guide and awareness to use this application.	I can view the awareness to use this application and its limitations.	Low	Sprint-1
		USN-2	As a user, I'm allowed to view the guided video to use the interface of this application.	I can gain knowledge to use this application by a practical method.	Low	Sprint-1

		USN-3	As a user, I can read the instructions to use this application.	I can read instructions also to use it in a userfriendly method.	Low	Sprint-2
	Recognize	USN-4	As a user, In this prediction page I get to choose the image.	I can choose the image from our local system and predict the output.	High	Sprint-2
	Predict	USN-6	As a user, I'm Allowed to upload and choose the image to be uploaded	I can upload and choose the image from the system storage and also in any virtual storage.	Medium	Sprint-3
		USN-7	As a user, I will train and 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 the result.	High	Sprint-4
		USN-8	As a user, I can access the MNIST data set	I can access the MNIST data set to produce the accurate result.	Medium	Sprint-3
Customer (Webuser)	Home	USN-9	As a user, I can view the guide to use the web app.	I can view the awareness of this application and its limitations.	Low	Sprint-1

	Recognize	USN-10	As a user,I can use the web application virtually anywhere.	I can use the application portably anywhere.	High	Sprint-1
		USN-11	As it is an open source, can use it cost freely.	I can use it without any payment to be paid for it to access.	Medium	Sprint-2
		USN-12	As it is a web application, it is installation free	I can use it without the installation of the application or any software.	Medium	Sprint-4

Solution & Technical Architecture

User Stories

Technical Architecture:

The architectural diagram of the model is as below and the Technology used is shown in Table 1

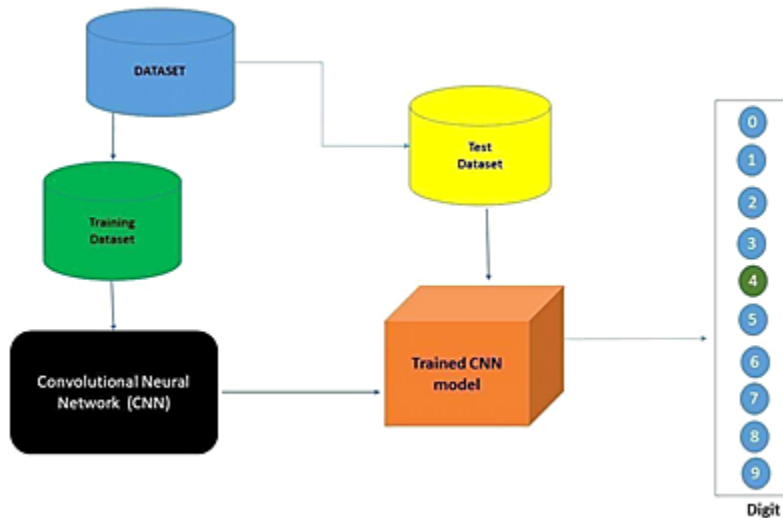


TABLE 1

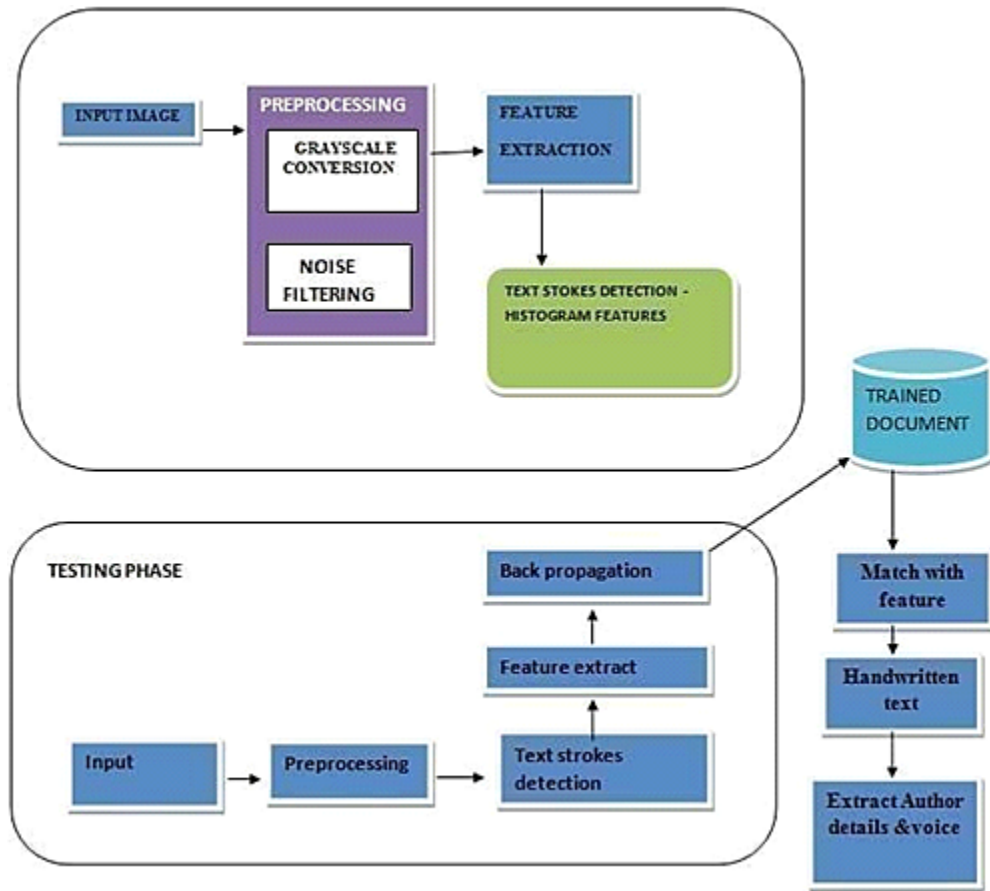


FIG. 1. BLOCK DIAGRAM

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	UserInterface	Howuser interacts withapplication e.g., Mobile Application	HTML,CSS, JavaScript / Angular JS / Node Red
2.	Application Logic-1	Logic for a process in the application	Java /Python
3.	Application Logic-2	Logic for a process in the application	IBMWatson STT service
4.	Application Logic-3	Logic for a process in the application	IBMWatson Assistant
5.	Database	DataType, Configurations etc.	MySQL, NoSQL, etc.

6.	CloudDatabase	Database Service on AI in cloud	IBM DB2
7.	FileStorage	Filestorage requirements	IBMBlock Storage or Other Storage Service or Local Filesystem
8.	External API-1	Purpose of External API used in the application	IBM Weather API,etc.
9.	Internet of Things Model	Purpose of AI Modelis for integrating the sensors with a user interface.	IBMAI Platform
10.	Machine Learning Model	Purpose of Machine Learning Model	ObjectRecognition Model
11.	Infrastructure (Server / AI)	Application Deployment on Local System / AI Local Server Configuration AI Server Configuration	Local,Kubernetes, etc.

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Deep leáning fíameworks can help you uploaddata and tíain a deep leáningmodel that would lead to accuíate and intuitive píedictive analysis.	Tensorflow, PyTorch
2.	Security Implementations	The system should automatically be able toauthenticate all userswith their uniqueusername and password	NA
3.	Scalable Architecture	Thesystem should be able to handle 10000 users accessing the siteat the same time	NA
4.	Availability	Information is restricted to each users limited	NA

		access	
--	--	--------	--

User Stories



PROJECT PLANNING & SCHEDULING

Sprint Planning & Estimation

Milestone	Function (Epic)	Milestone Story Number	Story / Task
Milestone -1	Registration	M1	As a user, I can register for the application by entering my email, password and confirming my password.
Milestone -2	Login	M2	As a user, I can log into the application by entering email & password.
Milestone -3	Prediction	M3	As a user, I can predict the digitalized output.
Milestone -4	Image Uploading	M4	As a user, I can able to upload images of the handwritten digit documents.
Milestone -5	Digit Recognition	M5	As a user, I can able to get the digital output of the digit from uploaded handwritten images.

Sprint Delivery Schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for application.	10	High	Thiyaneswaran S
Sprint-1	Log in	USN-2	As a user, I can log in into the application.	10	High	Anandaraj R

Sprint-2	Model creation	USN-3	As a user, I can use the model for predicting the handwritten digits.	10	High	Rajesh R
Sprint-2	Prediction	USN-4	As a user, I can predict the digitalized output.	10	High	Dhineshkumar S
Sprint-3	Upload Image of the handwritten document	USN-5	As a user, I can able to upload images of the handwritten digit documents.	20	Medium	Thiyaneswaran S, Anandaraj R
Sprint-4	Digit Recognition	USN-6	As a user, I can able to get the digital output of the digit from uploaded handwritten images.	20	Medium	Rajesh R, Dhineshkumar S

Velocity:

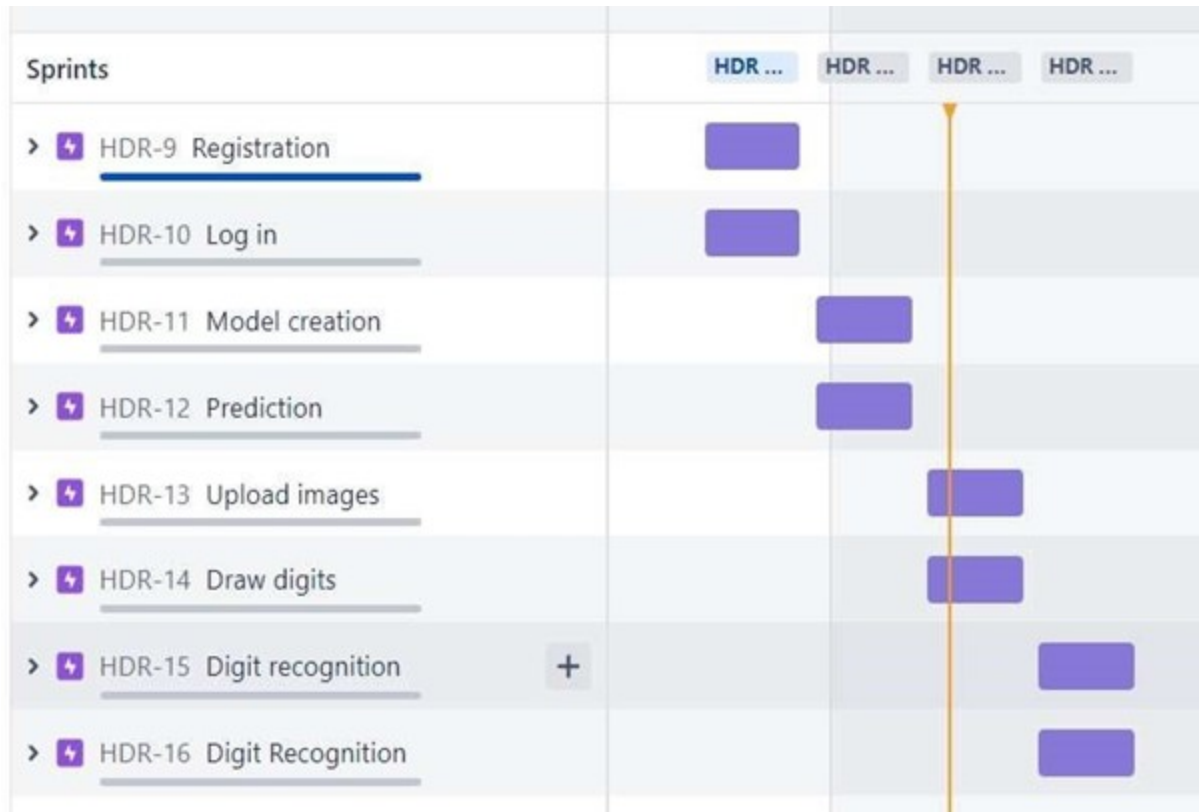
Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

$$AV = \frac{\text{sprint duration}}{\text{velocity}} = \frac{20}{10} = 2$$

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 charts can be applied to any project containing measurable progress over time.

Reports from JIRA



CODING AND SOLUTION

Feature 1

The feature that we have added is login credentials with the below code.

app.py

```
from flask
import (Flask,
render_templ
ate, request,
redirect,
url_for,
session)

from flask_session import Session
import dbworks as db
app = Flask(__name__, static_url_path='/static')
app.config["SESSION_PERMANENT"] = False app.config["SESSION_TYPE"] =
"filesystem"
Session(app)
@app.route('/')
@app.route('/Get
_Started') def
getStarted():
try:    del
session['user']
except:    pass
    return render_template('home.html')
@app.route('/Authentication', methods = ['POST','GET']) def
Authorize():
    if request.method == "POST":    if
request.form["type"] == "Login":
email = request.form["email"]
```



```

password = request.form["password"]
user = db.retreiveUsers(email)      if
user == []:

    return render_template('Authentication.html', msg="User Doesnot Exist")      if
user[0][3] != password:
    return render_template('Authentication.html', msg="Incorrect Password")      session['user']
= user;
    return redirect(url_for('uploadpage'))

elif request.form["type"] == "Signup":      username = request.form["username"]
email = request.form["email"]      password = request.form["password"]      if
password != request.form["cpassword"] and '' in [username,email,password]:
    return render_template('Authentication.html', msg="Invalid Data")
if db.retreiveUsers(username) != []:
    return render_template('Authentication.html', msg="User Already Exist")
ack = db.insertUser(username, email, password)      if not ack:
    return render_template('Authentication.html', msg="Invalid Data")      return
render_template('Authentication.html', msg="Successfully Signed In")
    return render_template('Authentication.html')

```

base.html

```

<!DOCTYPE html>

<html>

<head>

    <meta charset="utf-8">

        <meta name="viewport" content="width=device-width, initial-scale=1">

    <title>

        {%block title%}

        {%endblock%}

    </title>

    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/css/bootstrap.min.css"
integrity="sha384-

```

```

ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">

<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://cdn.jsdelivr.net/npm/popper.js@1.14.7/dist/umd/popper.min.js"
integrity="sha384-
UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1cHTMGa3JDZwrnQq4sF86dIHNDz0W1"
crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/js/bootstrap.min.js"
integrity="sha384-
JjSmVgyd0p3pXB1rRibZUAYyoly6OrQ6VrjIEaFf/nJGzlxFDsf4x0xIM+B07jRM"
crossorigin="anonymous"></script>

        <link href="{{url_for('static', filename = 'style.css')}}" rel="stylesheet">

<script type="text/javascript" src="{{url_for('static', filename = 'javascript.js')}}"></script>

        <style type="text/css">

                {% block style%}

                {% endblock %}

        </style>

</head>

<body onload=" {%block bodyelement%} {%endblock%} ">

        <nav class="nav-bar">

                <h1 class="title2" id="basetitle">Trucnket</h1>

                <div id="nav-bar-buttons">

                        <a href="{{url_for('getStarted')}}"><button class="btn nav-btn btn-outlineprimary" >Log
                        Out</button></a>

                </div>

        </nav>

{%block content1%}

{%endblock%}

{%block authentication%}

{%endblock%}

```

```
{%block uploadform%}
```

```
{%endblock%}
```

```
</body>
```

```
</html>
```

home.html

```
{%extends 'base.html'%}
```

```
{%block title%
```

```
    HomePage
```

```
{%endblock%}
```

```
{%block style%
```

```
.nav-bar{
```

```
    display : None;
```

```
}
```

```
{%endblock%}
```

```
{%block content1%
```

```
<div id="home-sector">
```

```
<div>
```

```
<div class="ball" id="blueball">
```

```
    <h1 id="title1">A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM</h1>
```

```
<a href="{{url_for('Authorize')}}"><button type="button" class="btn btn btn-primary"
```

```
    id="getstarted">Get Started</button></a>
```

```
</div>
```

```
<div class="ball" id="grayball"></div>
```

```
</div>
```

```
<div id="Aimage-box"><h1 class="title2">Trucnket</h1></div>
```

```
</div>
```

```
{%endblock%}
```

Authentication.html

```
{% extends 'base.html' %}

{% block title%}

    Authentication

{%endblock %} {%block style%} body{

    background-image: linear-gradient(to right, #045DE9 , #09C6F9);

}

.nav-bar{

    display:none;

}

#title2{

    color: white;

    text-shadow: 1px 1px 13px black;

}

{%endblock%}

{% block authentication %}

<div id="auth-sector">

<h1 class="title2">Trucnket</h1>

<div class="line"></div>

<div class="Authorizebox" id="Login">

    <div class="active whitescreen" id="loginscreen" onclick="swipeup()" >Login</div>

    <h1 class="auth-title">Login</h1><br>

    <form action="{{url_for('Authorize')}}" method="POST" id="loginform" class="authform">

        <input type="hidden" name="type" value="Login">

        <label>

            <input type="type" name="email" class="form-control"
```

```

placeholder="Username" required></label><br>
    <label>
        <input type="password" name="password" class="form-control"
placeholder="Password" required></label><br>
        <input type="submit" value="Submit" class="btn btn-success submit"> </form>
</div>
<div class="line1"></div>
<div class="Authorizebox" id="signup">
    <div class="whitescreen" id="signupscreen" onclick="swipeup()" >Signup</div> <h1
class="auth-title">Signup</h1><br>
    <form action="{{url_for('Authorize')}}" method="POST" id="signupform" class="auth-form">
        <div id="firsthalf">
            <input type="hidden" name="type" value="Signup">
            <label>
                <input type="text" name="username" class="form-control"
placeholder="Username" required></label><br>
                <label>
                    <input type="email" name="email" class="form-control"
placeholder="Email" required></label><br>
                    <button type="button" class="btn btn-success submit"
onclick="moveinputsignup()">Next</button>
        </div>
        <div id="passhalf">
            <label>
                <input type="password" name="password" class="form-control"
placeholder="Password" required></label><br>
                <label>
                    <input type="password" name="cpassword" class="form-
control" placeholder="Confirm Password" required></label><br>
                    <input type="submit" value="Submit" class="btn btn-success submit">

```

```

        </div>

    </form>

</div>

</div>

<span style="color:white">{{msg}}</span>

{% endblock %}

```

Feature 2

The feature that we have added is digit recognition using the below code.

App.py

```

from flask import (Flask,
render_template,
request,
redirect,
url_for,
session)
from flask_session import Session
from Module.predict import predict
import dbworks as db
app = Flask(__name__, static_url_path='/static')
app.config["SESSION_PERMANENT"] = False
app.config["SESSION_TYPE"] = "filesystem"
Session(app)
@app.route('/')
@app.route('/Get_Started')
def getStarted():
    try:
        del session['user']
    except:
        pass
    return render_template('home.html')
@app.route('/uploadpage',methods = ['POST','GET'])
def uploadpage():
    if not session['user']:
        return render_template('Authentication.html', msg="Login First")

```

```

if request.method == "POST":
    if request.form["type"]=="Upload":
        image = request.files['image']
        filesrc = './uploads/.predict.jpg'
        image.save(filesrc)
        best = predict(filesrc)
        if best == False:
            return render_template('debug.html',msg="Cannot able to Recognize, reupload with
correct image")
        user = session.get('user',None)
        user[0] = list(user[0])
        user[0][4] = ','+str(best)+user[0][4]
        db.updateHistory(user[0])
        return render_template('Uploadphoto.html', bodyele = int(best))
    return render_template('Uploadphoto.html', bodyele = '-')
@app.route('/DeleteAccount', methods=['POST','GET'])
def deleteaccount():
    if request.method == "POST":
        user = session['user'][0][1]
        db.deleteAccount(user)
        return render_template('Authentication.html')
    return render_template('deleteacc.html')
@app.route('/Authentication', methods = ['POST','GET'])
def Authorize():
    if request.method == "POST":
        if request.form["type"] == "Login":
            username = request.form["email"]
            password = request.form["password"]
            user = db.retreiveUsers(username)
            if user == []:
                return render_template('Authentication.html', msg="User Doesnot Exist")
            if user[0][3] != password:
                return render_template('Authentication.html', msg="Incorrect Password")
            session['user'] = user;
            return redirect(url_for('uploadpage'))
        elif request.form["type"] == "Signup":
            username = request.form["username"]
            email = request.form["email"]
            password = request.form["password"]
            if password != request.form["cpassword"] and " in [username,email,password]:
                return render_template('Authentication.html', msg="Invalid Data")

```

```

    if db.retreiveUsers(username) != []:
        return render_template('Authentication.html', msg="User Already Exist")
    ack = db.insertUser(username, email, password)
    if not ack:
        return render_template('Authentication.html', msg="Invalid Data")
    return render_template('Authentication.html', msg="Successfully Signed In")
return render_template('Authentication.html')

```

uploadphoto.html

```

{% extends 'base.html'%}
{%block title%}
A Novel Method For Handwritten Digit Recognition System
{%endblock%}
body{
    overflow:hidden;
}
{%block style%}
{%endblock%}
{%block uploadform%}
<div id="upload-sector">
    <div style="display: flex; align-items: center;">
<div class="uploadform">
    <form method="POST" action="{{url_for('uploadpage')}}" enctype="multipart/form-
data">

        <input type="hidden" name="type" value="Upload">
        <div class="custom-file">
            <label class="form-label">UPLOAD IMAGE</label><br>
            <input type="file" accept="image/*" name="image" required class="form-
control-sm upinputbox" id="formFileLg">
        </div>
        <br><br>
        <input type="submit" value="Submit" class="btn btn-success float-left select ">
    </form>
</div>
<div class="line" id="bline"></div>
</div>
<div id="resultbox">
    <div id="numbers">

```



```

        <span id="result">Result</span>
        <div class="numberbox" id="1"><span id="one"
class="number">{{bodyele}}</span></div>
    </div>
</div>
<div id="up-image">
</div>
</div>
{%endblock%}
{{msg}}

```

Base.html

```

<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>
        {%block title%}

        {%endblock%}
    </title>
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/css/bootstrap.min.css"
integrity="sha384-
ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">
        <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://cdn.jsdelivr.net/npm/popper.js@1.14.7/dist/umd/popper.min.js"
integrity="sha384-
UO2eT0CpHqdSJK6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1"
crossorigin="anonymous"></script>
        <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/js/bootstrap.min.js"
integrity="sha384-
JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM"

```

```

crossorigin="anonymous"></script>
    <link href="{ {url_for('static', filename = 'style.css') }}" rel="stylesheet">
    <script type="text/javascript" src="{ {url_for('static', filename =
'javascript.js') }}"></script>
    <style type="text/css">
        {% block style%}
        {% endblock %}
    </style>
</head>
<body onload=" {%block bodyelement%} {%endblock%} ">
    <nav class="nav-bar">
        <h1 class="title2" id="basetitle">Trucnket</h1>
        <div id="nav-bar-buttons">
            <a href="{ {url_for('deleteaccount') }}"><button class="red btn nav-btn
btn-outline-danger" >Delete Account</button></a>
            <a href="{ {url_for('getStarted') }}"><button class="blue btn nav-btn btn-
outline-primary" >Log Out</button></a>
        </div>
    </nav>
    {%block content1%}
    {%endblock%}
    {%block authentication%}
    {%endblock%}
    {%block uploadform%}
    {%endblock%}
</body>
</html>

```

Database Schema

dbworks.py

```

import sqlite3 as sql

con = sql.connect('database.db') def
insertUser(Username, Password,
Email):    try:

```

```

        con =
sql.connect('database.db')
cur = con.cursor()
        cur.execute("INSERT INTO USER (USERNAME,PASSWORD,EMAIL,HISTORY)
VALUES
        (?,?,,?)", (Username, Password,
Email, ""))    con.commit()
con.close()    return True    except:
return False
def
retrieveUsers
(Email):
try:
        con =
sql.connect('database.db')
cur = con.cursor()
cur.execute("SELECT * FROM
USER WHERE EMAIL = '"+
Email +"'");    user =
cur.fetchall()    con.close()
return user    except:
        return False

```

schema.sql

```

DROP TABLE IF EXISTS USERTABLE;
CREATE TABLE USER (
ID INTEGER PRIMARY KEY AUTOINCREMENT,
USERNAME TEXT NOT NULL,

```

PASSWORD TEXT NOT NULL,

EMAIL TEXT NOT NULL,

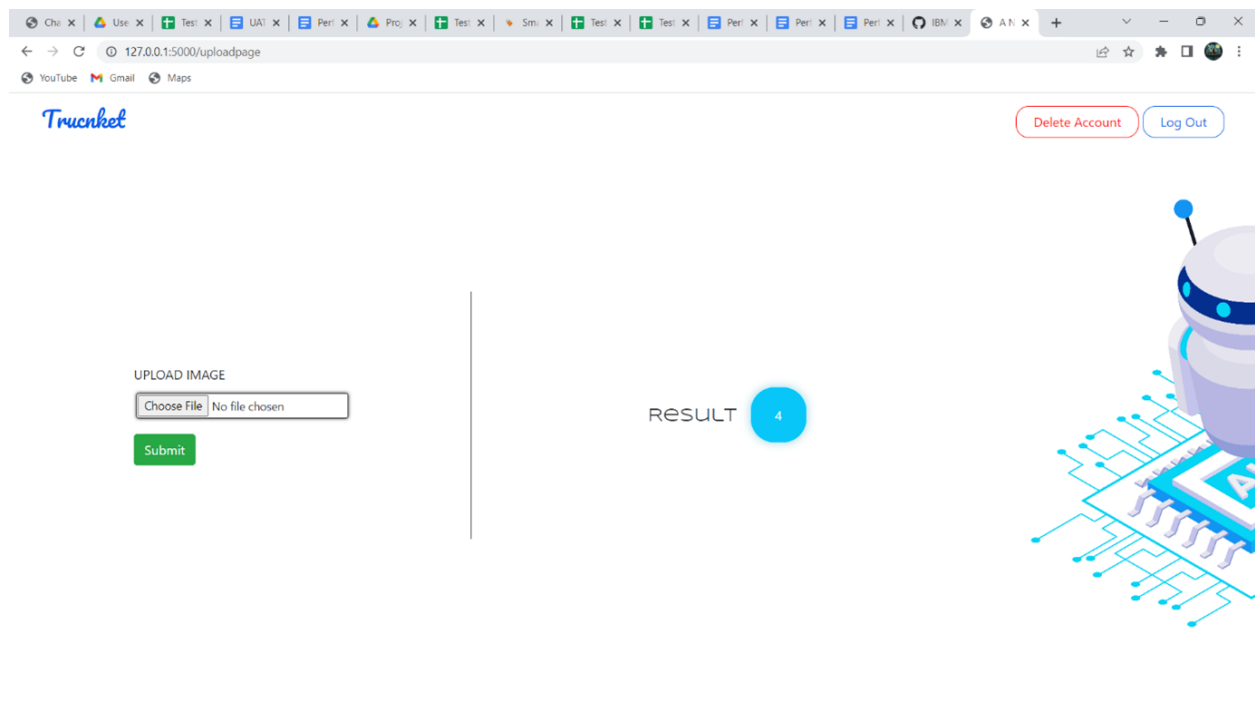
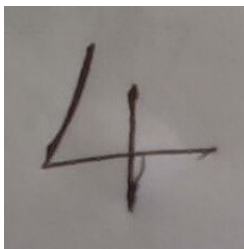
HISTORY TEXT

);

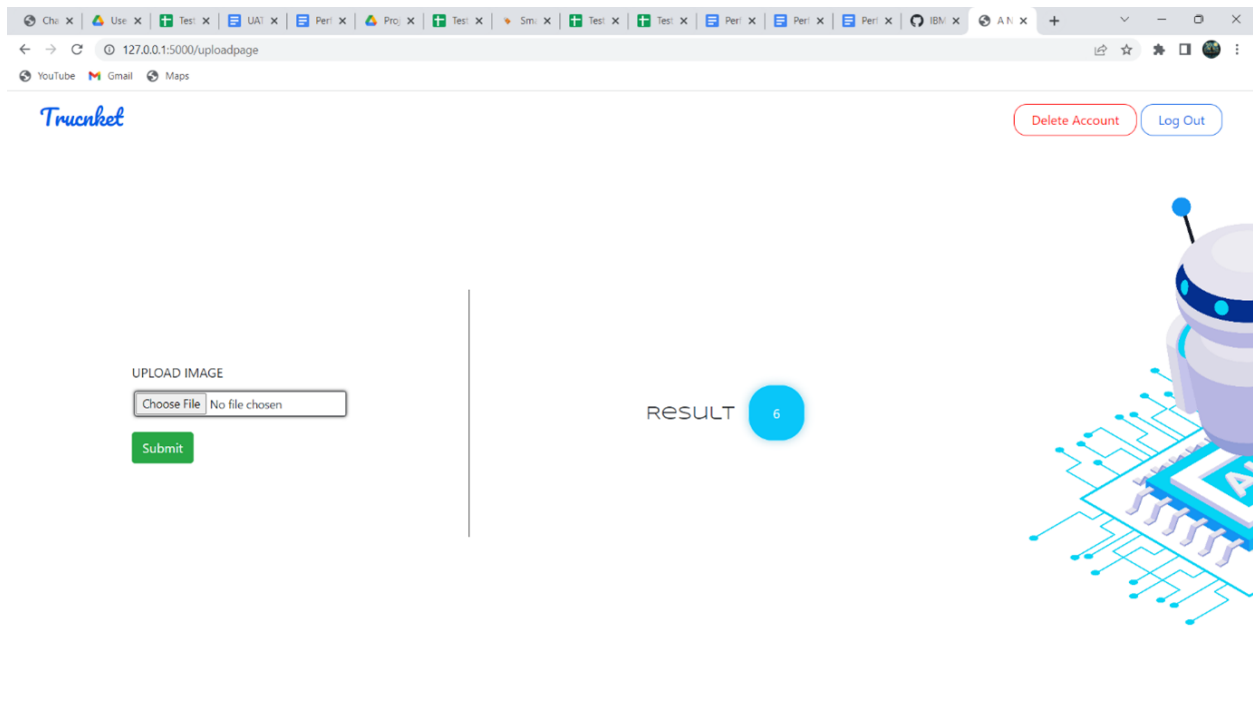
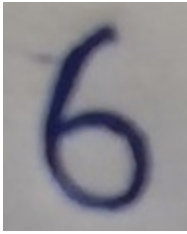
TESTING

Test Cases

Test case-1



Test case-2



RESULTS

Performance Metrics

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

```
Metrics(Test loss & Test Accuracy):
[0.08917615562677383, 0.972000002861023]
```

ADVANTAGES

- Interface that provided makes easy understanding to the user to how to find their need easily.
- Inbuild trained model helps to recognize the digits faster.
- Featured with Security to make data security in future.

DISADVANTAGES

- This model does not provide a hundred percent perfect result.
- Interface/UI is not much responsible to the devices (supports for devices larger than 1280px width)

CONCLUSION

Handwritten digit recognition has immense applications in the field of medical, banking, student management, and taxation process etc. Many classifiers like KNN, SVM, CNN are used to identify the digit from the handwritten image. as per the review, CNN is providing better performance than others. Stages of HDR using CNN classifier is discussed in this paper. MNIST dataset consist of handwritten numbers from 0-9 and it is a standard dataset used to find performance of classifiers. HDR consists of three different stages. First is preprocessing where dataset is converted into binary form and image processing has been applied on it. Second stage is segmentation where the image is converted into multiple segments. Third stage is feature extraction where features of image are identified. Last stage is classification where classifiers like KNN, SVM, CNN are used. Results of HDR is improved a lot by using CNN classifier but it can be improved further in terms of complexity, duration of execution and accuracy of results by making combination of classifiers or using some additional algorithm with it.

FUTURE SCOPE

1. Idea about bringing up history details that user have recognized image, details about logging in and logging out.
2. Creating a Responsive web application which can be accessed with any type of devices.

APPENDIX

SOURCE CODE

App.py

```
from flask import (Flask,
render_template,
request,
redirect,
url_for,
session)
from flask_session import Session
from Module.predict import predict
import dbworks as db
app = Flask(__name__, static_url_path='/static')
app.config["SESSION_PERMANENT"] = False
app.config["SESSION_TYPE"] = "filesystem"
Session(app)
@app.route('/')
@app.route('/Get_Started')
def getStarted():
    try:
        del session['user']
    except:
        pass
    return render_template('home.html')
@app.route('/uploadpage',methods = ['POST','GET'])
def uploadpage():
    if not session['user']:
        return render_template('Authentication.html', msg="Login First")
    if request.method == "POST":
        if request.form["type"]=="Upload":
            image = request.files['image']
```

```

        filesrc = './uploads/predict.jpg'
        image.save(filesrc)
        best = predict(filesrc)
        if best == False:
            return render_template('debug.html',msg="Cannot able to Recognize, reupload with correct
image")
        user = session.get('user',None)
        user[0] = list(user[0])
        user[0][4] = ','+str(best)+user[0][4]
        db.updateHistory(user[0])
        return render_template('Uploadphoto.html', bodyele = int(best))
    return render_template('Uploadphoto.html', bodyele = '-')
@app.route('/DeleteAccount', methods=['POST','GET'])
def deleteaccount():
    if request.method == "POST":
        user = session['user'][0][1]
        db.deleteAccount(user)
        return render_template('Authentication.html')
    return render_template('deleteacc.html')
@app.route('/Authentication', methods = ['POST','GET'])
def Authorize():
    if request.method == "POST":
        if request.form["type"] == "Login":
            username = request.form["email"]
            password = request.form["password"]
            user = db.retreiveUsers(username)
            if user == []:
                return render_template('Authentication.html', msg="User Doesnot Exist")
            if user[0][3] != password:
                return render_template('Authentication.html', msg="Incorrect Password")
            session['user'] = user;
            return redirect(url_for('uploadpage'))
        elif request.form["type"] == "Signup":

```



```

username = request.form["username"]
email = request.form["email"]
password = request.form["password"]
if password != request.form["cpassword"] and " " in [username,email,password]:
    return render_template('Authentication.html', msg="Invalid Data")
if db.retreiveUsers(username) != []:
    return render_template('Authentication.html', msg="User Already Exist")
ack = db.insertUser(username, email, password)
if not ack:
    return render_template('Authentication.html', msg="Invalid Data")
return render_template('Authentication.html', msg="Successfully Signed In")
return render_template('Authentication.html')

```

home.html

```

{%extends 'base.html'%}
{%block title%}
    HomePage
{%endblock%}
{%block style%}
.nav-bar{
    display : None;
}
{%endblock%}
{%block content1%}
<div id="home-sector">
<div>
<div class="ball" id="blueball">
    <h1 id="title1">A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION
SYSTEM</h1>
<a href="{{url_for('Authorize')}}"><button type="button" class="btn btn-btn-primary"
id="getstarted">Get Started</button></a>
</div>
<div class="ball" id="grayball"></div>

```

```

</div>
<div id="Aimage-box"><h1 class="title2">Trucnket</h1></div>
</div>
{%endblock%}

```

Base.html

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>
    {%block title%}
    {%endblock%}
  </title>

  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/css/bootstrap.min.css" integrity="sha384-
ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">

  <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://cdn.jsdelivr.net/npm/popper.js@1.14.7/dist/umd/popper.min.js"
integrity="sha384-
UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1"
crossorigin="anonymous"></script>

  <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/js/bootstrap.min.js"
integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM"
crossorigin="anonymous"></script>

  <link href="{ {url_for('static', filename = 'style.css')}}" rel="stylesheet">
  <script type="text/javascript" src="{ {url_for('static', filename = 'javascript.js')}}"></script>
  <style type="text/css">
    {% block style%}

```

```

        {% endblock %}
    </style>
</head>
<body onload=" {%block bodyelement%} {%endblock%} ">
    <nav class="nav-bar">
        <h1 class="title2" id="basetitle">Trucnket</h1>
        <div id="nav-bar-buttons">
            <a href="{{url_for('deleteaccount')}}"><button class="red btn nav-btn btn-outline-
danger" >Delete Account</button></a>
            <a href="{{url_for('getStarted')}}"><button class="blue btn nav-btn btn-outline-
primary" >Log Out</button></a>
        </div>
    </nav>
    {%block content1%}
    {%endblock%}
    {%block authentication%}
    {%endblock%}
    {%block uploadform%}
    {%endblock%}
</body>
</html>

```

Uploadphoto.html

```

{% extends 'base.html'%}
{%block title%}
A Novel Method For Handwritten Digit Recognition System
{%endblock%}
body{
    overflow:hidden;
}
{%block style%}
{%endblock%}
{%block uploadform%}

```

```

<div id="upload-sector">
    <div style="display: flex; align-items: center;">
<div class="uploadform">
    <form method="POST" action="{{url_for('uploadpage')}}" enctype="multipart/form-data">
        <input type="hidden" name="type" value="Upload">
        <div class="custom-file">
            <label class="form-label">UPLOAD IMAGE</label><br>
                <input type="file" accept="image/*" name="image" required class="form-control-sm
upinputbox" id="formFileLg">
        </div>
        <br><br>
        <input type="submit" value="Submit" class="btn btn-success float-left select ">
    </form>
</div>
<div class="line" id="bline"></div>
</div>
<div id="resultbox">
    <div id="numbers">
        <span id="result">Result</span>
        <div class="numberbox" id="1"><span id="one"
class="number">{{bodyele}}</span></div>
    </div> </div>
<div id="up-image">
</div> </div>
{%endblock%}
{{msg}}

```

GITHUB LINK

<https://github.com/IBM-EPBL/IBM-Project-42631-1660671210>

DEMO LINK

<https://drive.google.com/file/d/1593w09gkrdLudJoNck7E3g16lmgI5Ww2/view?usp=sha>

ring