

HANDWRITTEN DIGIT RECOGNITION

TABLE OF CONTENTS

| S.NO | TITLE | PAGE |
|------|-----------------------------------|------|
| 1 | Introduction | 3 |
| 1.1 | Project Overview | 3 |
| 1.2 | Purpose | 3 |
| 2 | Literature Survey | 4 |
| 2.1 | Existing problem | 4 |
| 2.2 | References | 4 |
| 2.3 | Problem Statement | 5 |
| 3 | Ideation & Proposed Solution | 6 |
| 3.1 | Empathy Map canvas | 6 |
| 3.2 | Ideation & Brainstorming | 6 |
| 3.3 | Proposed Solution | 7 |
| 3.4 | Problem Solution Fit | 8 |
| 4 | Requirement Analysis | 8 |
| 4.1 | Functional Requirement | 8 |
| 4.2 | Non-functional Requirement | 9 |
| 5 | Project Design | 9 |
| 5.1 | Data Flow Diagrams | 9 |
| 5.2 | Solution & Technical Architecture | 10 |
| 5.3 | User Stories | 13 |
| 6 | Project Planning & Scheduling | 14 |

| | | |
|-----|------------------------------|----|
| 6.1 | Sprint Planning & Estimation | 14 |
| 6.2 | Sprint Delivery Schedule | 15 |
| 6.3 | Reports from JIRA | 15 |
| 7 | Coding & Solutioning | 16 |
| 7.1 | Feature-1 | 16 |
| 7.2 | Feature-2 | 16 |
| 8 | Testing | 16 |
| 8.1 | Test Cases | 16 |
| 8.2 | User Acceptance Testing | 17 |
| 9 | Results | 18 |
| 9.1 | Performance Metrics | 18 |
| 10 | Advantages & Disadvantages | 18 |
| 11 | Conclusion | 19 |
| 12 | Future Scope | 19 |
| 13 | Appendix | 20 |
| | Source Code | 20 |
| | GitHub & Project Demo Link | 28 |

1. INTRODUCTION

1.1. PROJECT OVERVIEW

Handwritten digit recognition is the process to provide the ability to machines to recognize human handwritten digits. It is not an easy task for the machine because handwritten digits are not perfect, vary from person-to-person, and can be made with many different flavors. The rapid growth in the need for digitizing handwritten data and the availability of massive processing power demands improvement in recognition accuracy. Hence a highly proficient algorithm is required when dealing with handwriting recognition. A neural network is a model inspired by how the brain works. It consists of multiple layers having many activations, this activation resembles neurons of our brain. A neural network tries to learn a set of parameters in a set of data which could help to recognize the underlying relationships. Neural networks can adapt to changing input; so the network generates the best possible result without needing to redesign the output criteria. The MNIST data set is widely used for this recognition process and it has 70000 handwritten digits. Those images are split as train set and test set images. Artificial neural networks are used 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 analysed by the model and the detected result is returned on to the UI.

1.2. PURPOSE

The main aim is to use the neural network approach for recognizing handwritten digits. The Convolution Neural Network has become the center of all deep learning strategies. Optical character recognition (OCR) is a part of image processing that leads to excerpting text from images. Recognizing handwritten digits is part of OCR. Recognizing the numbers is an important and remarkable subject. In this way, since the handwritten digits are not of same size, thickness, position, various difficulties are faced in determining the problem of recognizing handwritten digits. The unlikeness and structure of the compositional styles of many entities further influences the example and presence of the numbers. This is the strategy for perceiving and organizing the written characters. Its applications are such as programmed bank checks, health, post offices, for education, etc.

2. LITERATURE SURVEY

2.1. EXISTING PROBLEM

Because of the progress in science and technology everything is being digitised to reduce human effort. It takes a lot of time and effort on the side of manual workers when sorting through mails by postal codes. It is not an easy task to handle data by human workers. There is also the possibility of human error while processing huge amounts of data. Therefore, digitizing these will help reduce time and labour. The labour cost will also be reduced with the help of machines. There are also the activities of processing bank checks and tax documentations. Large piles of records and archives can be arranged and sorted well, easing the stress and workload from manual labourers. The problem with handwriting is that every individual has a different style of writing. There is a differing thickness of stroke, style and general uniqueness that just brings a level of hardness in identifying the handwriting. The machine must be capable of picking up the digits correctly with a good accuracy rate. Hence a highly proficient algorithm is required when dealing with handwriting recognition. Handwritten digit recognition can be done using deep learning methods effectively.

2.2. REFERENCES

- [1] Handwritten Digit Recognition Using Various Machine Learning Algorithms and Models Pranita Patil, Bhupinder Kaur.
- [2] A comparison of three classification algorithms for the identification of handwritten digits R.G Jadhav.
- [3] Development of a high precision handwritten digit recognition detector based on a Convolution Neural Network Sonia Flora, Anju kakkad
- [4] A Novel Handwritten Digit Classification System Based on Convolutional Neural Network Approach, Ali Abdullah Yahya, 18 September 2021
- [5] Own Handwritten Digit recognition using MLP and CNN in tensorflow Deepti D.Nikumbh, Rupali S. Kale.
- [6]. Handwritten digit recognition using deep learning, International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), vol. 6, no. 7, A. Dutta and A. Dutta, July 2017.
- [7]. HANDWRITTEN DIGIT RECOGNITION USING OPENCV AND CNN K. Swetha, Y. Hithaishi.
- [8]. Handwritten Digits Recognition, Project Report, Gaurav Jain, Jason Ko, University of Toronto, 11/21/2008.

- [9]. Handwritten recognition using SVM, KNN and neural network, arXiv preprint arXiv:1702.00723. Hamid, Norhidayu Abdul, and NilamNur Amir Sjarif, 2017.
- [10]. Digit Classification using the MNIST Dataset, M. Wu and Z. Zhang, Handwritten, 2010.
- [11]. Handwritten digit classification using support vector machines, R.G.Mihalyi, 2011.

2.3. PROBLEM STATEMENT DEFINITION

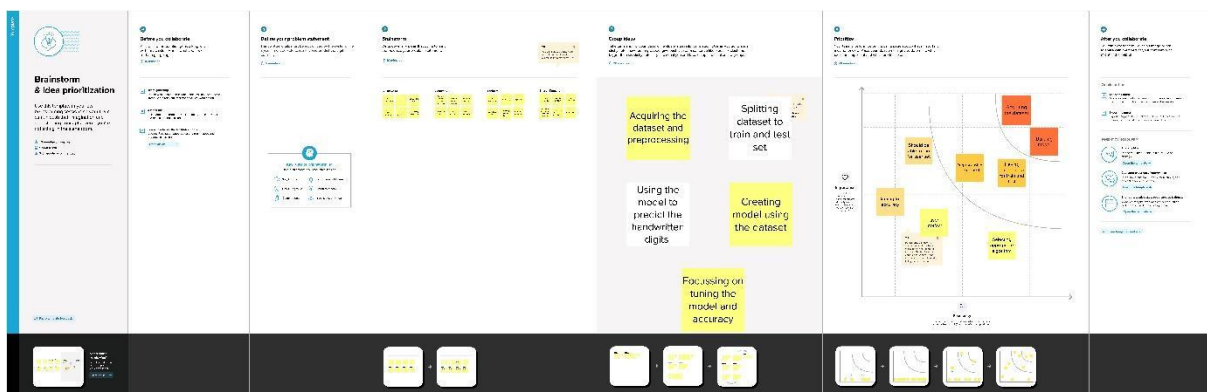
Handwriting recognition is one of the compelling research works going on because every individual in this world has their own style of writing. Since the style of handwriting changes with every individual, it is a challenging task in identifying the characters correctly. The thickness of stroke, style carries uniqueness with different people depending on them. 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. The MNIST data set is widely used for this recognition process and it has 70000 handwritten digits. Artificial neural networks are used to train these images and build a deep learning model. The Convolutional Neural Networks (CNN) is a deep learning algorithm that is highly suitable for image recognition and those tasks involving processing of pixel data. Convolutional neural networks (CNNs) are very effective in perceiving the structure of handwritten characters/words in ways that help in automatic extraction of distinct features and make CNN the most suitable approach for solving handwriting recognition problems. Our aim in the proposed work is to deploy the CNN model effectively and produce a good result with better accuracy. The main objective was to actualize a pattern characterization method to perceive the handwritten digits provided in the MINIST dataset of images of handwritten digits (0-9). Web application is created where the user can upload an image of a handwritten digit. This image is analysed by the model and the detected result is returned on to the UI.

3. IDEATION AND PROPOSED SOLUTION

3.1. EMPATHY MAP CANVAS



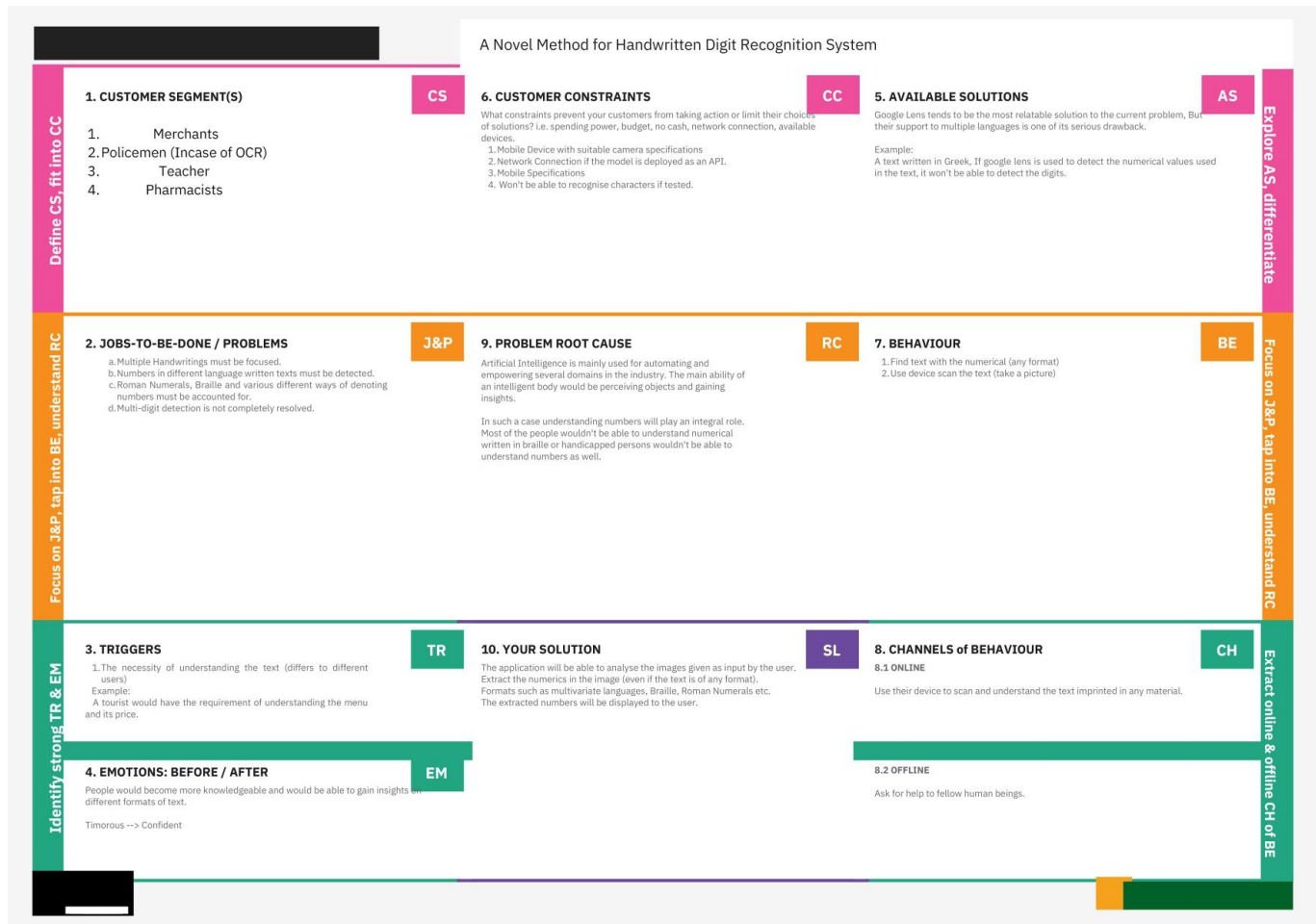
3.2. IDEATION AND BRAINSTORMING



3.3. PROPOSED SOLUTION

| S. No | Parameter | Description |
|-------|---------------------------------------|---|
| 1. | Problem Statement | To develop a system that will identify the handwritten digit correctly |
| 2. | Idea / Solution Description | <ul style="list-style-type: none"> a. Predict the digit using deep learning algorithms b. Ensure the correct prediction of digit |
| 3. | Novelty / Uniqueness | <ul style="list-style-type: none"> a. Predict the digits instantly b. Recognizing digits irrespective of the varying handwriting styles |
| 4. | Social impact / Customer Satisfaction | Serves workers at postal offices and bank, where it can be used for mail sorting and check processing. Also helpful in data form entry. It will reduce the work load from workers and hence reduce stress |
| 5. | Business Model | <ul style="list-style-type: none"> a. Collaboration with postal office and banks. And other corporations using database b. Saves time and cost with manual labor c. Bank check programming d. Tax documentation |
| 6. | Scalability of the Solution | <ul style="list-style-type: none"> a. Fine tuning the model aiming to produce more accurate results b. Making it independent with less human intervention |

3.4. PROBLEM SOLUTION FIT



4.0. REQUIREMENT ANALYSIS

4.1. FUNCTIONAL REQUIREMENTS

Following are the functional requirements of the proposed solution

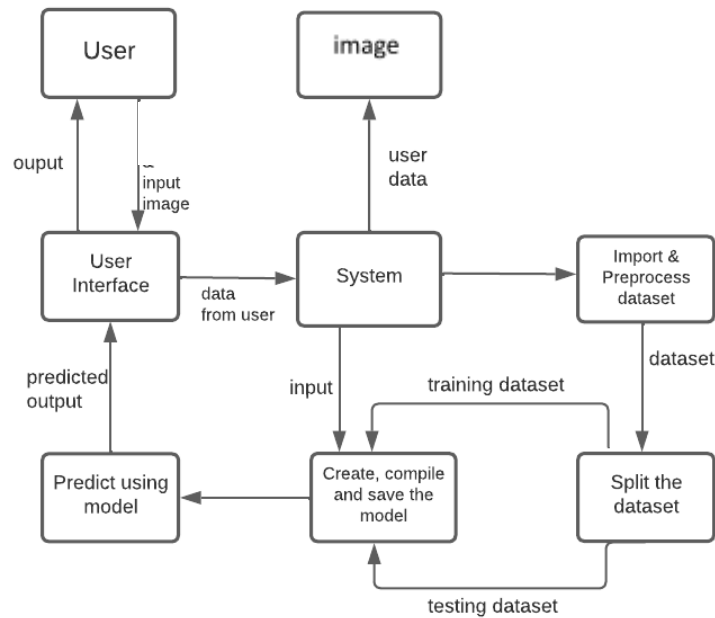
| FR No. | Functional Requirement (Epic) | Sub Requirement (Story / Sub-Task) |
|--------|-------------------------------|--|
| FR-1 | User Registration | Registration through Form and Google Auth |
| FR-2 | User Confirmation | Confirmation via Email |
| FR-3 | Image Input Form | A flask wtf form with image upload as fields |
| FR-4 | Result | Result is displayed as a popup and an email notification is sent to the user |
| FR-5 | User Details | The user can view/update user details. |
| FR-6 | ChatBot | A chatbot for assistance. |

4.2. NON-FUNCTIONAL REQUIREMENTS

| FR No. | Non-Functional Requirement | Description |
|--------|----------------------------|--|
| NFR-1 | Usability | Responsive and interactive UI allows the user to access the app with ease |
| NFR-2 | Security | The user data is encrypted and stored securely. By using authentication mechanisms user accounts are secured |
| NFR-3 | Reliability | Data Loss wouldn't occur since the usage of IBM Cloud. |
| NFR-4 | Performance | The prediction model Performance will be dynamic and improves over time. The user interface requires less heavy APIs and libraries to render |
| NFR-5 | Availability | Will be available throughout the time since the application is deployed in IBM Cloud. |
| NFR-6 | Scalability | The application is scalable and can cater users based on traffic |

5. PROJECT DESIGN

5.1. DATA FLOW DIAGRAM



1. User Interface
2. Input from the user
3. System loads the dataset
4. Splitting into training and testing
5. CNN modelling
6. Output prediction
7. Display the output

5.2. SOLUTION AND TECHNICAL ARCHITECTURE

Solution Architecture

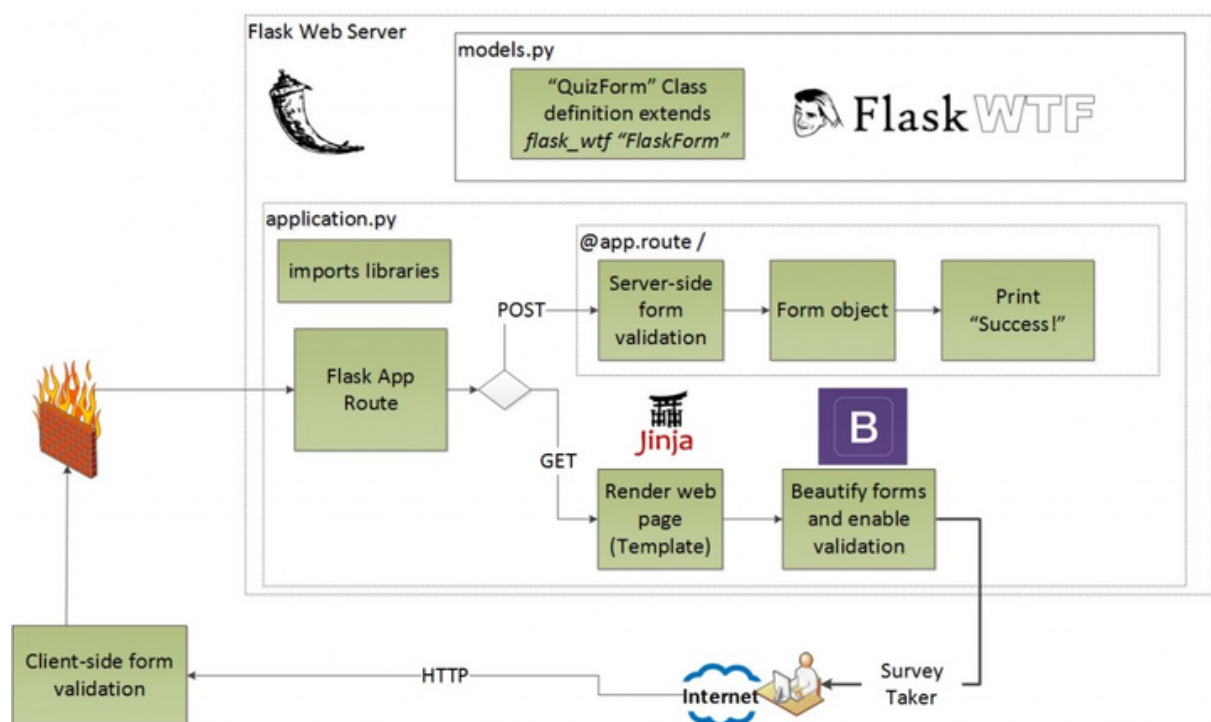
Deep learning has evolved into an extensive field within research and development. In image analysis, **convolutional neural networks (CNN)** have been particularly successful. Modified National Institute of Standards and Technology (MNIST) is a large set of computer vision dataset which is extensively used for training and testing different systems. It was created from the two special datasets of National Institute of Standards and Technology (NIST) which holds binary images of handwritten digits. The training set contains handwritten digits from 250 people, among them 50% training dataset

was employees from the Census Bureau and the rest of it was from high school students. However, it is often attributed as the first datasets among other datasets to prove the effectiveness of the neural networks.

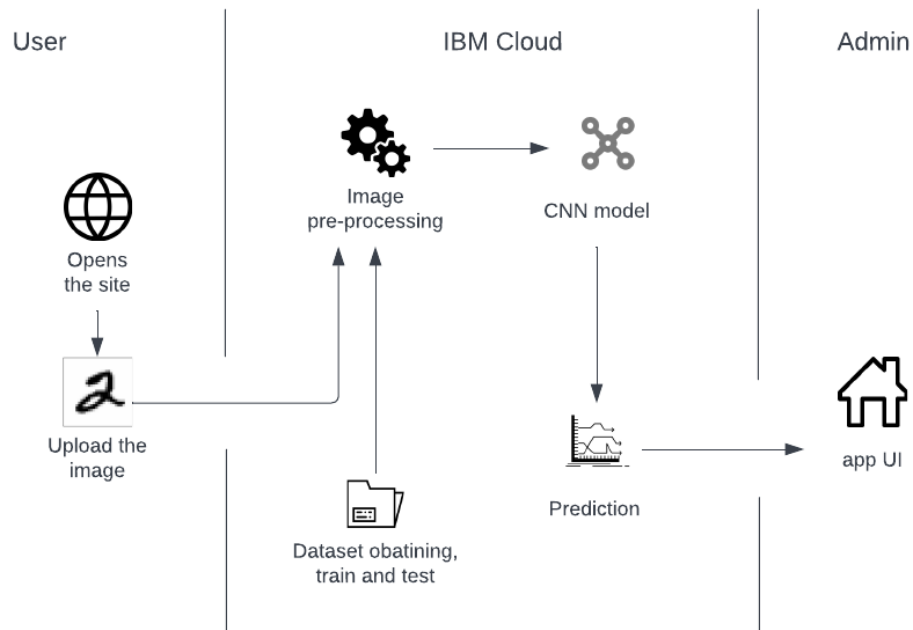
The major steps involved in this.

1. Load the dataset
2. Splitting into training and testing
3. CNN modelling
 - 3.1. Convolution
 - 3.2. Pooling
 - 3.3. Fully connected
4. Output prediction

Architecture Diagram



Technical architecture



Components & Technologies:

| S.No | Component | Description | Technology |
|------|------------------------|--|--|
| 1. | User Interface | User Interface consist of multiple pages 1. Home Page 2. Sign In Page 3. Sign Up Page 4. User Page | HTML, CSS, JavaScript |
| 2. | Application Logic-1 | Auth Logic: 1. Checking whether the user is present 2. Password requirements 3. email id requirements | Python |
| 3. | Application Logic-2 | Image Processing Logic 1. Initial Image cropping 2. Adding filters for smoothening (Gaussian Blur) | Python |
| 4. | Application Logic-3 | Automatic Chatbot | IBM Watson Assistant |
| 5. | Database | 2 tables (user table, image mapping table) | MySQL |
| 6. | Cloud Database | Database Service on Cloud | IBM DB2, IBM Cloudant etc. |
| 7. | File Storage | File storage requirements | IBM Block 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. | Machine Learning Model | Tensorflow model (.h5 format) developed using VGG16 net. | Object Recognition Model, etc. |

Application Characteristics:

| S.No | Characteristics | Description | Technology |
|------|--------------------------|---|-------------------------------|
| 1. | Open-Source Frameworks | List the open-source frameworks used | OpenCV, Tensorflow |
| 2. | Security Implementations | List all the security / access controls implemented, use of firewalls etc. | Flask Login, Web Authn |
| 3. | Scalable Architecture | Kubernetes is open-source orchestration software that provides an API to control how and where those containers will run | Kubernetes (Dockerization) |
| 4. | Availability | Nginx is a web server that can also be used as a reverse proxy, load balancer, mail proxy and HTTP cache. | Nginx and Docker |
| 5. | Performance | Latency - time for each request. Rate - No. of requests per minute etc. No of failures - If there is a failure, how many etc. | Scout APM, Sentry.io, Postman |

5.3. USER STORIES

| 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 the application by entering my email, password, and confirming my password. | 2 | High | Ashwin Jaison Jacob |
| Sprint-1 | | USN-2 | As a user, I will receive confirmation email once I have registered for the application | 2 | High | Ashwin Jaison Jacob |
| Sprint-2 | | USN-3 | As a user, I can register for the application through Facebook | 2 | Low | Aldrin Joe Shakthi M |
| Sprint-1 | | USN-4 | As a user, I can register for the application through Gmail | 2 | Medium | Ashwin Aldrin Joe |
| Sprint-1 | Login | USN-5 | As a user, I can log into the application by entering email & password | 1 | High | Ashwin Aldrin Joe Jaison Jacob Shakthi M |

6. PROJECT PLANNING & SCHEDULING

6.1. SPRINT PLANNING AND ESTIMATION

| 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 the application by entering my email, password, and confirming my password. | 2 | High | Ashwin Jaison Jacob |
| Sprint-1 | | USN-2 | As a user, I will receive confirmation email once I have registered for the application | 2 | High | Ashwin Jaison Jacob |
| Sprint-2 | | USN-3 | As a user, I can register for the application through Facebook | 2 | Low | Aldrin Joe Shakthi M |
| Sprint-1 | | USN-4 | As a user, I can register for the application through Gmail | 2 | Medium | Ashwin Aldrin Joe |
| Sprint-1 | Login | USN-5 | As a user, I can log into the application by entering email & password | 1 | High | Ashwin Aldrin Joe Jaison Jacob Shakthi M |

6.2. SPRINT DELIVERY SCHEDULE

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

6.3. REPORTS FROM JIRA

Velocity:

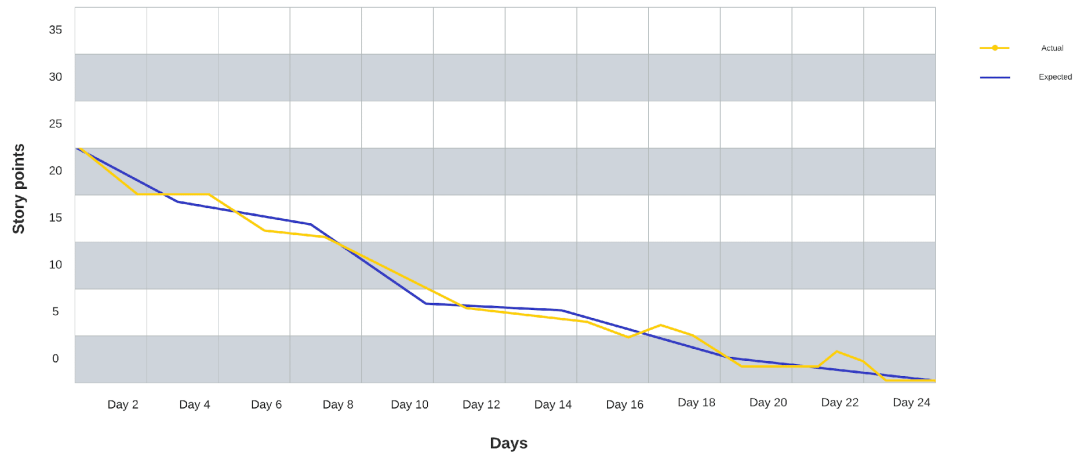
Imagine we have a 6-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

$$AV = \text{sprint duration} / \text{velocity}$$

$$AV = 20/6 = 3.33$$

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



7. CODING AND SOLUTIONING

7.1. FEATURE-1 MODEL BUILDING

ML depends heavily on data, without data, it is impossible for a machine to learn. It is the most crucial aspect that makes algorithm training possible. In Machine Learning projects, we need a training data set. It is the actual data set used to train the model for performing various actions. TensorFlow already has a MNIST Data set so there is no need to explicitly download or create Dataset. The MNIST dataset contains ten classes: Digits from 0-9. Each digit is taken as a class. The required libraries are imported which are required for the model to run. The dataset for this model is imported from the Keras module. The data is split into train and test. Using the training dataset, the model is trained and the testing dataset is used to predict the results. Basically, the pixel values range from 0-255. The value of each image is stored in `y_train`. The model is built with convolutional, pooling and dense layers. The created model is then compiled and saved.

7.2. FEATURE-2 WEB APP

HTML, CSS and JavaScript are used to create the web pages for the front end. An html page that takes in image files as input using form and submits to the back end is created. A flask app is created using python flask, where it receives the image files from the templates, html pages and the prediction operation is done over this image. Later the predicted output is sent to the result page.

8. TESTING

8.1 TEST CASES

| | | | | | | | | | |
|------------------|------------|------------|--|--|---|---|---|-------------|------|
| LoginPage_TC_001 | Functional | Login page | Verify user is able to login to application with valid credentials | | 1. Enter URL and click go 2. Click on Sign In 3. Enter Valid username/email in Email text box 4. Enter valid password in password text box 5. Click on login button | Username: d@y.com password: arhuin2000 | Application should go to User Page | Passed | Pass |
| LoginPage_TC_002 | Functional | Login page | Verify user is able to login to application with Invalid credentials | | 1. Enter URL and click go 2. Click on Sign In 3. Enter Invalid username/email in Email text box 4. Enter Invalid password in password text box 5. Click on login button | Username: rdard@gmail.com password: Torting1234567890123456789 | Application should show 'Incorrect email or password' validation message. | As Expected | Pass |
| LoginPage_TC_003 | Functional | Login page | Verify user is able to login to application with invalid credentials | | 1. Enter URL and click go 2. Click on Sign In 3. Enter Invalid username/email in Email text box 4. Enter valid password in password text box 5. Click on login button | Username: apple2@gmail.com password: arhuin2000 | Application should go to Home page with error | As Expected | Pass |
| LoginPage_TC_004 | Functional | Login page | Verify user is able to login to application with invalid credentials | | 1. Enter URL and click go 2. Click on Sign In 3. Enter Valid username/email in Email text box 4. Enter Invalid password in password text box 5. Click on login button | Username: d@y.com password: arh2000 | Application should go to Home page with error | As Expected | Pass |
| Maths_TC_001 | Functional | User page | Verify whether the canvas is working | | 1. Enter User Page 2. Try Drawing Digit in canvas | Draw the digit 8 | The drawing must be visible in the canvas | As Expected | Pass |
| | | | | | 1. Enter URL and click go 2. Click on Sign In | Draw the digit 9 | Application should display 9 | | |

8.2 USER ACCEPTANCE TEST

Defect Analysis

This report shows the 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 | 0 | 4 | 2 | 3 | 9 |
| Duplicate | 1 | 0 | 3 | 0 | 4 |
| External | 2 | 3 | 0 | 1 | 6 |
| Fixed | 20 | 0 | 1 | 0 | 21 |
| Not Reproduced | 0 | 0 | 1 | 0 | 1 |
| Skipped | 0 | 0 | 0 | 2 | 2 |
| Won't Fix | 2 | 0 | 0 | 0 | 2 |
| Totals | 25 | 7 | 7 | 7 | 45 |

Test Case Analysis

This report shows the number of test cases that have passed, failed, and untested

| Section | Total Cases | Not Tested | Fail | Pass |
|---------------------|-------------|------------|------|------|
| Login Page | 10 | 3 | 0 | 7 |
| User Interface | 3 | 0 | 1 | 2 |
| Model Endpoints | 2 | 1 | 0 | 1 |
| Exception Reporting | 9 | 0 | 0 | 9 |
| Final Report Output | 4 | 0 | 0 | 4 |
| Version Control | 3 | 0 | 0 | 3 |

9. RESULTS

9.1 PERFORMANCE METRICS

| S.No. | Parameter | Values | Screenshot | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--------------------------------|--------------------|---|---|--------------|--------------|---------|-----------------|-------------------|-----|------------------------------|-------------------|---|-------------------|--------------------|------|--------------------------------|------------------|---|-------------------|-------------|---|---------------|-------------|-------|-----------------|------------|-------|-----------------|------------|-----|
| 1. | Model Summary | | <div>Model: "sequential"</div> <table><thead><tr><th>Layer (type)</th><th>Output Shape</th><th>Param #</th></tr></thead><tbody><tr><td>conv2d (Conv2D)</td><td>(None, 28, 28, 6)</td><td>156</td></tr><tr><td>max_pooling2d (MaxPooling2D)</td><td>(None, 14, 14, 6)</td><td>0</td></tr><tr><td>conv2d_1 (Conv2D)</td><td>(None, 10, 10, 16)</td><td>2416</td></tr><tr><td>max_pooling2d_1 (MaxPooling2D)</td><td>(None, 5, 5, 16)</td><td>0</td></tr><tr><td>flatten (Flatten)</td><td>(None, 400)</td><td>0</td></tr><tr><td>dense (Dense)</td><td>(None, 120)</td><td>48120</td></tr><tr><td>dense_1 (Dense)</td><td>(None, 84)</td><td>10164</td></tr><tr><td>dense_2 (Dense)</td><td>(None, 10)</td><td>850</td></tr></tbody></table> <div>Total params: 61,706 Trainable params: 61,706 Non-trainable params: 0</div> | Layer (type) | Output Shape | Param # | conv2d (Conv2D) | (None, 28, 28, 6) | 156 | max_pooling2d (MaxPooling2D) | (None, 14, 14, 6) | 0 | conv2d_1 (Conv2D) | (None, 10, 10, 16) | 2416 | max_pooling2d_1 (MaxPooling2D) | (None, 5, 5, 16) | 0 | flatten (Flatten) | (None, 400) | 0 | dense (Dense) | (None, 120) | 48120 | dense_1 (Dense) | (None, 84) | 10164 | dense_2 (Dense) | (None, 10) | 850 |
| Layer (type) | Output Shape | Param # | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| conv2d (Conv2D) | (None, 28, 28, 6) | 156 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| max_pooling2d (MaxPooling2D) | (None, 14, 14, 6) | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| conv2d_1 (Conv2D) | (None, 10, 10, 16) | 2416 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| max_pooling2d_1 (MaxPooling2D) | (None, 5, 5, 16) | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| flatten (Flatten) | (None, 400) | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dense (Dense) | (None, 120) | 48120 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dense_1 (Dense) | (None, 84) | 10164 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dense_2 (Dense) | (None, 10) | 850 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2. | Accuracy | <div>Training Accuracy 99.2 %</div> <div>Validation Accuracy 99.8 %</div> <div>Epochs Trained 200</div> | <div>Epoch 4/200 625/625 [=====] - 5s 9ms/step - loss: 0.0443 - accuracy: 0.9855 - val_loss: 7.8187 - val_accuracy: 0.9860</div> <div>Epoch 5/200 625/625 [=====] - 5s 9ms/step - loss: 0.0363 - accuracy: 0.9884 - val_loss: 7.9836 - val_accuracy: 0.9895</div> <div>Epoch 6/200 625/625 [=====] - 6s 10ms/step - loss: 0.0298 - accuracy: 0.9902 - val_loss: 8.5668 - val_accuracy: 0.9865</div> <div>Epoch 7/200 625/625 [=====] - 5s 9ms/step - loss: 0.0251 - accuracy: 0.9918 - val_loss: 8.2603 - val_accuracy: 0.9845</div> <div>Epoch 8/200 625/625 [=====] - 5s 9ms/step - loss: 0.0219 - accuracy: 0.9929 - val_loss: 6.6184 - val_accuracy: 0.9910</div> <div>Epoch 9/200 625/625 [=====] - 6s 9ms/step - loss: 0.0197 - accuracy: 0.9933 - val_loss: 7.5502 - val_accuracy: 0.9875</div> <div>Epoch 10/200 625/625 [=====] - 6s 9ms/step - loss: 0.0180 - accuracy: 0.9941 - val_loss: 12.2241 - val_accuracy: 0.9855</div> | | | | | | | | | | | | | | | | | | | | | | | | | | | |

10. ADVANTAGES & DISADVANTAGES

ADVANTAGES:

- It saves times for arranging and sorting huge amount of data
- Only requires far less physical space than the storage of the physical copies.
- Recognising multiple digits on a single frame using sequential model in Keras.
- Data storage, for an example, there are many files, contracts and some personal records that contains some handwritten digits.
- It reduces human effort and labour cost
- This can be used for sorting through mail by postal code

DISADVANTAGES

- The system build is complex and holds difficulty
- The handwriting of every individual varies which proves to be a challenge for the system to predict
- Possible unemployment of labour that is typical of technology growth
- The accuracy is not guarantees and there are risk of errors

11. CONCLUSION

Handwritten digit recognition has immense applications in the field of medical, banking, student management, and taxation process etc. Many classifiers like KNN, SVM, and CNN are used to identify the digit from the handwritten image. Here we've used CNN for implementation. Convolutional Neural Network gets trained from the real-time data and makes the model very simple by reducing the number of variables and gives relevant accuracy. MNIST dataset consist of handwritten numbers from 0-9 and it is a standard dataset used to find performance of classifiers.

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. More accurate results can be established with more convolution layers and more number of hidden neurons. It can completely abolish the need for typing. Digit recognition is an excellent prototype problem for learning about neural networks and it gives a great way to develop more advanced techniques of deep learning.

12. FUTURE SCOPE

In future, different architectures of CNN, namely, hybrid CNN, viz., CNN-RNN and CNN-HMM models, and domain-specific recognition systems, can be investigated. Evolutionary algorithms can be explored for optimizing CNN learning parameters, namely, the number of layers, learning rate and kernel sizes of convolutional filters. The future development of the applications based on algorithms of deep and machine learning is practically boundless.

In the future, we can work on a denser or hybrid algorithm than the current set of algorithms with more manifold data to achieve the solutions to many problems. In future, the application of these algorithms lies from the public to high-level authorities, as from the differentiation of the algorithms above and with future development we can attain high-level functioning applications which can be used in the classified or government agencies as well as for the common people. Currently only the digits are recognized. In future the all the characters in all the language can be predicted with high accuracy rate.

13. APPENDIX

Source code

The necessary libraries are imported.

```
import keras
import tensorflow
from keras.datasets import mnist
from keras.layers import Dense, Dropout, Flatten
from keras.layers import Conv2D, MaxPooling2D
from keras import backend as K
from tensorflow.keras.utils import to_categorical
import matplotlib.pyplot as plt
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D
from tensorflow.keras.layers import MaxPooling2D
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Flatten
from tensorflow.keras.optimizers import SGD
```

The MNIST dataset is downloaded from the keras library and the data is analyzed.

```
# the data, split between train and test sets
(x_train,y_train),(x_test,y_test)=mnist.load_data()
print(x_train.shape,y_train.shape)
print(x_test.shape,y_test.shape)
x_train[0]
```

The data is pre-processed and reshaped

```
#Preprocess the data
num_classes=10
x_train=x_train.reshape(x_train.shape[0],28,28,1)
x_test=x_test.reshape(x_test.shape[0],28,28,1)
input_shape = (28,28,1)
```

Applying one-hot encoding. The class vectors are converted to binary class matrices.

```
#Convert class vectors to binary class matrices
y_train=keras.utils.to_categorical(y_train,num_classes)
y_test=keras.utils.to_categorical(y_test,num_classes)
x_train=x_train.astype('float32')
x_test=x_test.astype('float32')
x_train=x_train/255
x_test=x_test/255
print('x_train shape:',x_train.shape)
print(x_train.shape[0],'train samples')
```

The CNN model is created. The activation function is Rectified linear unit(ReLU). The pooling layers, dense layers are added and flattened.

```
#Create the Model
batch_size=128
num_classes=10
epochs=20
model = Sequential()
model.add(Conv2D(32,
kernel_size=(3,3),activation='relu',input_shape=input_shape))
model.add(Conv2D(64,(3,3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.25))
model.add(Flatten())
model.add(Dense(64,activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(num_classes,activation='softmax'))
```

The model is then compiled

```
model.compile(loss=keras.losses.categorical_crossentropy,
optimizer=keras.optimizers.
Adadelta(),metrics=['accuracy'])
```

The model is trained

```
hist = model.fit(x_train,
y_train,batch_size=20,epochs=5,verbose=1,validation_data=(x_test, y_test))
```

Observing the metrics and testing the model

```
metrics = model.evaluate(x_test, y_test, verbose=0)
print("Metrics(Loss and Accuracy):")
print(metrics)
prediction = model.predict(x_test[:4])
print(prediction)
```

The model is saved and then tested. A sample image is given in to test the saved model. The image is reshaped and then predicted.

```
model.save('digit_classifier.h5')
from keras.utils.image_utils import img_to_array
from tensorflow.keras.models import load_model
model = load_model('/content/digit_classifier.h5')
from PIL import Image
import numpy as np

img = Image.open('/content/sample.png').convert("L")
```

```

img = img.resize((28,28))
im2arr = np.array(img)
im2arr = im2arr.reshape(1,28,28,1)

#display the image
import matplotlib.pyplot as plt
plt.imshow(img)

#predict the image
y_predict = model.predict(im2arr)
print(np.argmax(y_predict[0]))

```

The pages to display the home and recognise page with navigation bar.

HDR front end.html

```

<html>
  <head>
    <style>
      body {
        background-image:
url('https://cdn.pixabay.com/photo/2020/09/23/03/54/background-5594879_1280.jp
g');
        margin: 0;
        font-family: 'Times New Roman', Times, serif, Helvetica,
sans-serif;
      }

      .topnav {
        overflow: hidden;
        background-color: rgb(255, 255, 255);
      }

      .topnav a {
        float: left;
        color: #480557;
        text-align: center;
        padding: 14px 16px;
        text-decoration: none;
        font-size: 17px;
      }

      .topnav a:hover {
        background-color: rgb(57, 55, 55);
        color: rgb(250, 248, 248);
      }

      .topnav a.active {
        background-color: #f8e406;
        color: rgb(19, 19, 19);
      }
    p{

```



```

        text-align: center;
        background-color: rgb(8, 0, 0);
        margin-left: 25%;
        margin-right: 25%;
        margin-top: 5%;
        font-family: 'Times New Roman', Times, serif;
        color: aliceblue;
        font-size: large;
    }
</style>
</head>
<body>

    <div class="topnav">
        <a class="active" href="#home">Home</a>
        <a href="recognise.html">Recognise</a>
    </div>
    <p style="font-size: larger">Handwritten Digit Recognition</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. 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>
    </body>
</html>

```

The recognise page where the user can upload the image for prediction

recognise.html

```

<html>
    <head>
        <meta charset="UTF-8">
        <meta http-equiv="X-UA-Compatible" content="IE=edge" />
        <title>Digit Recognition</title>
        <style>
            body {
                background-image:
url('https://img.freepik.com/premium-vector/falling-colorful-messy-numbers-mat
h-study-concept-with-flying-digits-fascinating-back-school-mathematics-banner-
white-background-falling-numbers-vector-illustration_174187-2929.jpg?w=2000');
                margin: 0;
            }
        </style>
    </head>
    <body>
        <div class="topnav">
            <a class="active" href="#home">Home</a>
            <a href="recognise.html">Recognise</a>
        </div>
        <p style="font-size: larger">Handwritten Digit Recognition</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. 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>
    </body>
</html>

```

```

        font-family: 'Times New Roman', Times, serif, Helvetica,
sans-serif;
        height: 100%;
        width: 100%;
    }
    h1 {
        display: block;
        font-size: 3.5em;
        margin-top: 5.4em;
        margin-bottom: 0em;
        margin-left: 50%;
        margin-right: 0;
        font-weight: bold;
    }

    .button {
        border: #e5b9f3;
        color: rgb(56, 1, 69);
        padding: 15px 32px;
        text-align: center;
        text-decoration: none;
        display: inline-block;
        font-size: 16px;
        margin: 4px 2px;
        cursor: pointer;
    }

    .button1 {
        background-color: #b6e6f0;
        margin-top: 5.4em;
        margin-left: 56%;
        margin-right: 0;
        border: black;
    }
    .button2 {
        background-color: #b6e6f0;
        margin-top: 5.5em;
        margin-left: 55%;
        margin-right: 0;
        border: black;
    }
}
</style>
<script>
    function view() {
        frame.src=URL.createObjectURL(event.target.files[0]);
    }

    $('#submit').click(function(){
        $.ajax({
            url: 'app.py',
            type: 'POST',

```

```

    })
  })

  var input = document.getElementById('image');
  var infoArea = document.getElementById('frame');

  input.addEventListener('change', showFileName);

  function showFileName(event) {
    var input = event.srcElement;
    var fileName = input.files[0].name;
    infoArea.textContent = 'File name: ' + fileName;
  }

</script>
<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+965Dz00rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6
jizo" crossorigin="anonymous"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"
integrity="sha384-U02eT0CpHqdSJJQ6hJty5KVphtPhzWj9W01c1HTMGa3JDZwrnQq4sF86dIHND
z0W1" crossorigin="anonymous"></script>
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"
integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy60rQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B0
7jRM" crossorigin="anonymous"></script>
<script
src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@latest"></script>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.0/jquery.min.js"></script>

</head>
<body>
  <h1 style = "color: rgb(2, 60, 0)">Digit Recognition
  <br>
  <br>
  <form action="/predict" method="POST" enctype="multipart/form-data">

    <input id="image" type="file" name="image" accept="image/png,
image/jpeg" onchange="view()"><br><br>
    <img id="frame" type="submit" src="" width="100px" height="100px"/>

    <input type="submit" value="Recognise" class="button button2"/>
  </h1>
</form>

```

```
</body>
</html>
```

The page where the predicted output is displayed

predict.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge" />
  <title>Prediction</title>
</head>

<style>
  body{
    background-image:
url('https://img.freepik.com/premium-vector/falling-colorful-messy-numbers-mat
h-study-concept-with-flying-digits-fascinating-back-school-mathematics-banner-
white-background-falling-numbers-vector-illustration_174187-2929.jpg?w=2000');
    background-repeat: no-repeat;
    height: 100%;
    width: 100%
  }
  .button{
    border:#e5b9f3;
    color: rgb(56, 1, 69);
    padding: 15px 32px;
    text-align: center;
    text-decoration: none;
    display: inline-block;
    font-size: 16px;
    margin-left:45%;
    cursor: pointer;
  }

  #pred{
    text-align: center;
    font-family: 'Times New Roman', Times, serif;
    font-size: 40px;
    margin: 0 auto;
    padding: 3% 5%;
    padding-top: 15%;
    color: rgb(0, 10, 80);
  }
}
```

```

</style>

<body>

    <p id="pred">PREDICTION : {{ num }}</p>
    <p>
        <a href="recognise.html">
            <button data-inline="True" class="button">Back</button>
        </a>
    </p>
</body>
</html>

```

The flask app.py python code to calculate the prediction value from processing the image uploaded by the user

app.py

```

import os
import numpy as np
from flask import Flask, render_template, request
import tensorflow as tf
from PIL import Image
from werkzeug.utils import secure_filename

UPLOAD_FOLDER = 'C:\Users\AKSHAYA\Pictures\static\images'

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

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

model = tf.keras.models.load_model("digit_classifier.h5")

@app.route('/predict', methods = ['GET', 'POST'])
def upload_image_file():
    if request.method == 'POST':
        imagefile = request.files['image']
        filename = secure_filename(imagefile.filename)
        imagefile.save(os.path.join(app.config['UPLOAD_FOLDER'], filename))
        path_img = os.path.join(UPLOAD_FOLDER, filename)
        img = Image.open(path_img).convert("L")
        img = img.resize((28,28))
        im2arr = np.array(img)
        im2arr = im2arr.reshape(1,28,28,1)

```

```
y_pred = model.predict(im2arr)

return render_template('predict.html', num = str(y_pred))

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

Github

<https://github.com/IBM-EPBL/IBM-Project-37884-1660361840>

Demo link

<https://github.com/IBM-EPBL/IBM-Project-37884-1660361840/tree/main/Final%20Deliverables/demo%20video>