A NOVEL METHOD FOR HANDWRITTEN DIGITAL RECOGNITION SYSTEM

A PROJECT REPORT

Submitted by

TEAM ID : PNT2022TMID49347

UMESH S	**923119106010**
KIRUTHIKA T	**923119106002**
KOWSALYA M	**923119106004**
MARUTHUPANDIYAN V	**923119106006**
SNEKA S	**923119106007**
SUVIN KUMAR R	**923119106008**

in partial fulfilment for the award of the degree

of

BATCHLOR OF ENGINEERING

in

ELECTRONICS AND COMMUNICATION ENGINEERING

ARULMURUGAN COLLEGE OF ENGINEERING
THENNILAI, KARUR-639206

ANNAUNIVERSITY :: CHENNAI 600 025

NOVEMBER 2022

TABLE OF CONTENTS

CHAPTER 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 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 analysed by the model and the detected result is returned on to UI

1.2 PURPOSE

The most common use case in today's mobile world is handwriting recognition as a direct input to a touchscreen through a stylus or finger. This is useful as it allows the user to quickly jot down numbers and names for contacts as compared to inputting the same information via the on-screen keyboard.

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 flavours. Digit recognition systems are capable of recognizing the digits from different sources like emails, bank cheque, papers, images, etc.

CHAPTER 2

LITERATURE SURVEY

2.1 Existing problem

In this paper Handwritten digit recognition has recently been of very interest among the researchers because of the evolution of various Machine Learning, Deep Learning and Computer Vision algorithms. In this report, We compare the results of some of the most widely used Machine Learning Algorithms like CNN- convolutional neural networks and with Deep Learning algorithms like multilayer CNN using Keras with Theano and Tensorflow. MNIST is a dataset which is widely used for handwritten digit recognition. The dataset consist of 60,000 training images and 10,000 test images.The artificial neural networks can all most mimic the human brain and are a key ingredient in image processing field.For example Convolution Neural networks with back propagation for image processing.The applications where these handwritten digit recognition can be used are Banking sector where it can be used to maintain the security pin numbers, it can be also used for blind peoples by using sound output.

2.2 References

PAPER 1

TITLE: A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION WITH NEURAL NETWORKS

AUTHOR : Malothu Nagu

YEAR : 2011

This paper Character recognition plays an important role in the modern world. It can solve more complex problems and makes humans' jobs easier. An example is handwritten character recognition. This is a system widely used in the world to recognize zip code or postal code for mail sorting. There are different techniques that can be used to recognize handwritten characters. Two techniques researched in this paper are Pattern Recognition and Artificial Neural

Network (ANN). Both techniques are defined and different methods for each technique are also discussed. Bayesian Decision theory, Nearest Neighbour rule, and Linear Classification or Discrimination is types of methods for Pattern Recognition. Shape recognition, Chinese Character and Handwritten Digit recognition using Neural Network to recognize them. Neural Network is used to train and identify written digits. After training and testing, the accuracy rate reached 99%.This accuracy rate is very high

PAPER 2

TITLE: A NOVEL HANDWRITTEN CHARACTER RECOGNITION SYSTEM USING GRADIENT BASED FEATURES AND RUN LENGTH COUNT

AUTHOR : G.Raju, Bindu S Moni

YEAR : 2014

In this paper, we propose a novel handwritten character recognition system using a combination of gradient-based features and run length count (GBF–RLC). The performance of the proposed method has been tested on Malayalam script, a South Indian language. The gradient of the image is the intensity at each point, giving the direction of the largest possible increase from light to dark and the rate of change in that direction. RLC is the count of a contiguous group of 1’s encountered in a left to right/top to bottom scan of a character image or block of an image. Classification was carried out with a Simplified Quadratic Classifier (SQDF) and Multi Layer Perception (MLP). A database containing 19,800 isolated handwritten characters pertaining to 44 classes was used for the study. The feature vector is augmented by including aspect ratio, position of centroid and ratio of pixels on the vertical halves of a character image. The recognition accuracy of 99.78% was achieved with minimum computational and storage requirements

PAPER 3

TITLE: A MULTI-LANGUAGE HANDWRITTEN DIGITS RECOGNITION DATASET

AUTHOR :Weiwei Jiang

YEAR : 2015

In this paper, we contribute a multi-language handwritten digit recognition dataset named MNIST-MIX, which is the largest dataset of the same type in terms of both languages and data samples. With the same data format as MNIST, MNIST-MIX can be seamlessly applied in existing studies for handwritten digit recognition. By introducing digits from 10 different languages, MNIST-MIX becomes a more challenging dataset and its imbalanced classification requires a better design of models. We also present the results of applying a LeNet model which is pre-trained on MNIST as the baseline.

PAPER 4

TITLE: AUTOMATIC HANDWRITTEN DIGIT RECOGNITION ON DOCUMENT IMAGES USING MACHINE LEARNING METHODS

AUTHOR :Akkireddy Challa

YEAR : 2019

The main purpose of this thesis is to build an automatic handwritten digit recognition method for the recognition of connected handwritten digit strings. To accomplish the recognition task, first, the digits were segmented into individual digits. Then, a digit recognition module is employed to classify each segmented digit completing the handwritten digit string recognition task. In this study, different machine learning methods, which are SVM, ANN and CNN architectures are used to achieve high performance on the digit string recognition problem. In these methods, images of digit strings are trained with the SVM, ANN and CNN model with HOG feature vectors and Deep learning methods structure by sliding a fixed size window through the images labelling each sub-image as a part of a digit or not. After the completion of the segmentation, to achieve the complete recognition of handwritten digits.

PAPER 5

TITLE: A NOVEL METHOD FOR HAND WRITTEN DIGITAL RECOGNITION USING DEEP LEARNING

AUTHOR : Rohini.M, D.Surendran

YEAR : 2019

In this paper Handwritten digit recognition has recently been of very interest among the researchers because of the evolution of various Machine Learning, Deep Learning and Computer Vision algorithms. In this report, We compare the results of some of the most widely used Machine Learning Algorithms like CNN- convolutional neural networks and with Deep Learning algorithms like multilayer CNN using Keras with Theano and Tensorflow. MNIST is a dataset which is widely used for handwritten digit recognition. The dataset consist of 60,000 training images and 10,000 test images.The artificial neural networks can all most mimic the human brain and are a key ingredient in image processing field.For example Convolution Neural networks with back propagation for image processing.The applications where these handwritten digit recognition can be used are Banking sector where it can be used to maintain the security pin numbers, it can be also used for blind peoples by using sound output

PAPER 6

TITLE: A NOVEL HANDWRITTEN DIGIT CLASSIFICATION SYSTEM BASED ON CONVOLUTIONAL NEURAL NETWORK APPROACH

AUTHOR : Ali Abdullah Yahya , Jieqing Tan

YEAR : 2021

This paper contains an enormous number of CNN classification algorithms that have been proposed in the literature. Nevertheless, in these algorithms, appropriate filter size selection, data preparation, limitations in datasets, and noise have not been taken into consideration. As a consequence, most of the algorithms have failed to make a noticeable

improvement in classification accuracy. To address the shortcomings of these algorithms, our paper presents the following contributions: Firstly, after taking the domain knowledge into consideration, the size of the effective receptive field (ERF) is calculated. Calculating the size of the ERF helps us to select a typical filter size which leads to enhancing the classification accuracy of our CNN. Secondly, unnecessary data leads to misleading results and this, in turn, negatively affects classification accuracy. To guarantee the dataset is free from any redundant or irrelevant variables to the target variable, data preparation is applied before implementing the data classification mission. Thirdly, to decrease the errors of training and validation, and avoid the limitation of datasets, data augmentation has been proposed. Fourthly, to simulate the real-world natural influences that can affect image quality, we propose to add an additive white Gaussian noise with $\sigma = 0.5$ to the MNIST dataset. As a result, our CNN algorithm achieves state-of-the-art results in handwritten digit recognition, with a recognition accuracy of 99.98%, and 99.40% with 50% noise

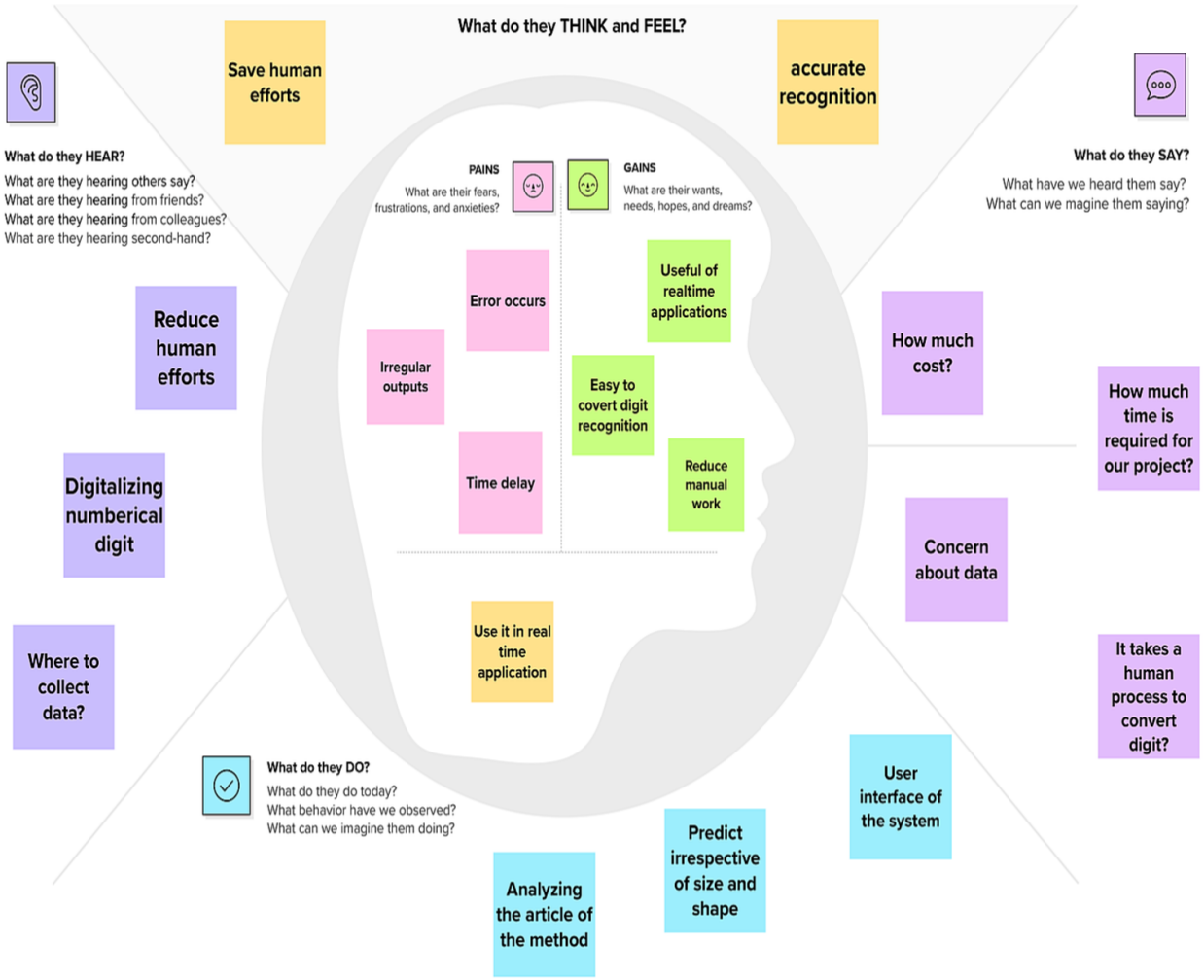
2.3 Problem Statement Definition

Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	A bank employee	Recognize the digit written on cheque	I can't recognize it	The digits are not written properly	Confused
PS-2	A student	recognize the last date for paying the exam fees which is written on the board	I can't find the correct date	The shapes of the digits are a little bit different.	Sad that I can't recognize the date
PS-3	A placement officer	recognize students DOB details and update to database	I can't able to understand some digits	Some digits are scribbled	Tensed that I can't able to recognize the digits

CHAPTER 3

IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS



3.2 IDEATION AND BRAINSTORMING

Template

Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

10 minutes to prepare
1 hour to collaborate
2-8 people recommended

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

A **Team gathering**
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B **Set the goal**
Think about the problem you'll be focusing on solving in the brainstorming session.

C **Learn how to use the facilitation tools**
Use the Facilitation Superpowers to run a happy and productive session.

Open article →

1

Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

5 minutes

PROBLEM

A Novel Method for Handwritten Digit Recognition

Handwriting recognition is one of the compelling research works goin g 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

Key rules of brainstorming

To run an smooth and productive session

- Stay in topic.
- Encourage wild ideas.
- Defer judgment.
- Listen to others.
- Go for volume.
- If possible, be visual.

2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

Umesh

- The hand writing must be legible
- Only the numeric character should be darken in the image
- Feedback should be enabled
- High accuracy

Sneka

- Enhanced dataset shoud be used
- Convert the text digits into electrical form
- DBN algorithm can be used
- Digit as output

Maruthupandiyan

- DNN algorithm can be used
- Variation in character style
- Mainly used in number plate detection
- Concern about data

Kowsalya

- Low time consumption
- Image as output
- Digits of different style recognised
- User interface of the system

Suvinkumar

- CNN algorithm can be used
- Accuracy should be displayed
- Display accuracy of detection
- Analysing the article of the method

Kiruthika

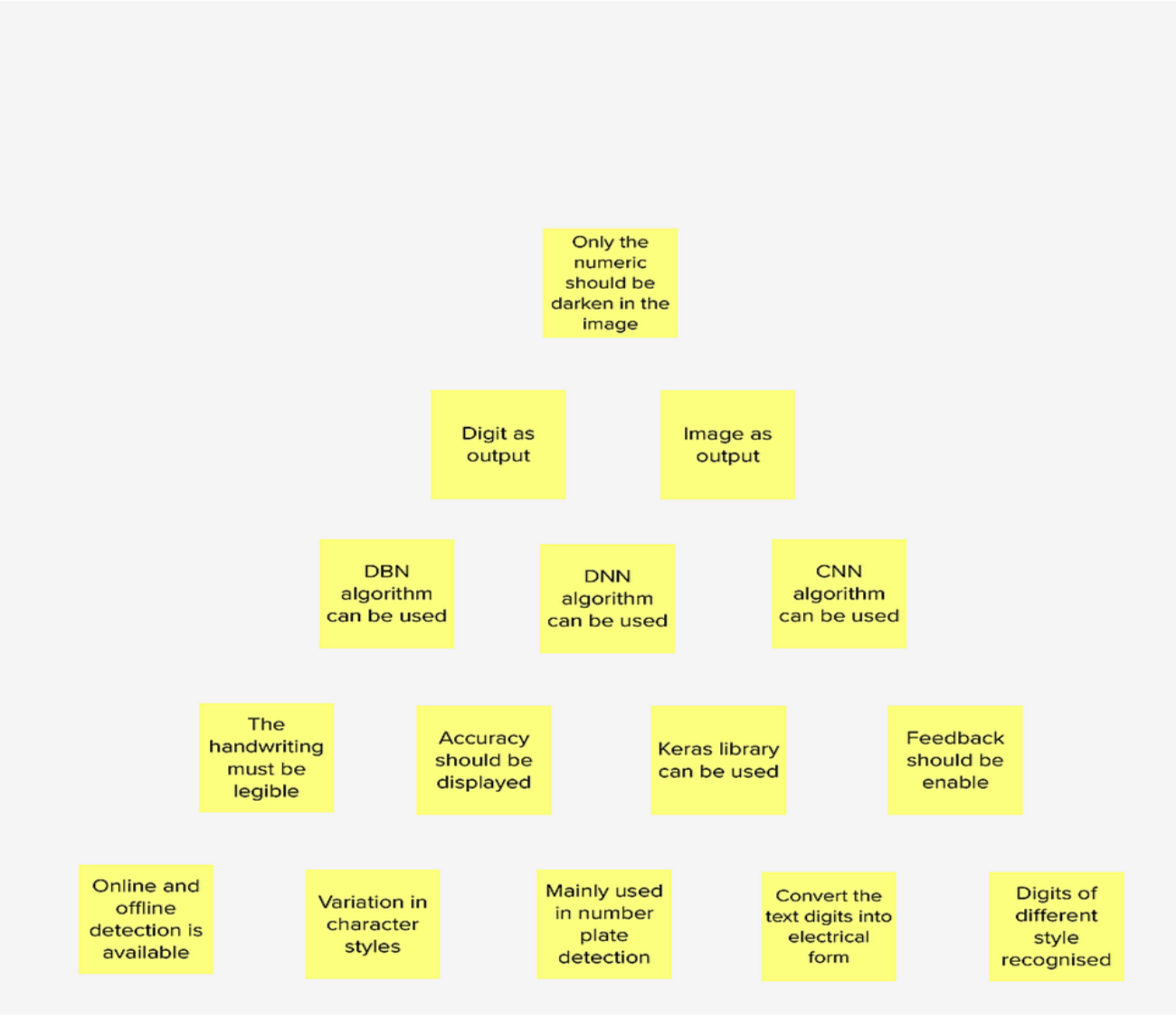
- Online and offline detection is available
- Keras library can be used
- The number must be dark enough
- Reduce human efforts

3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

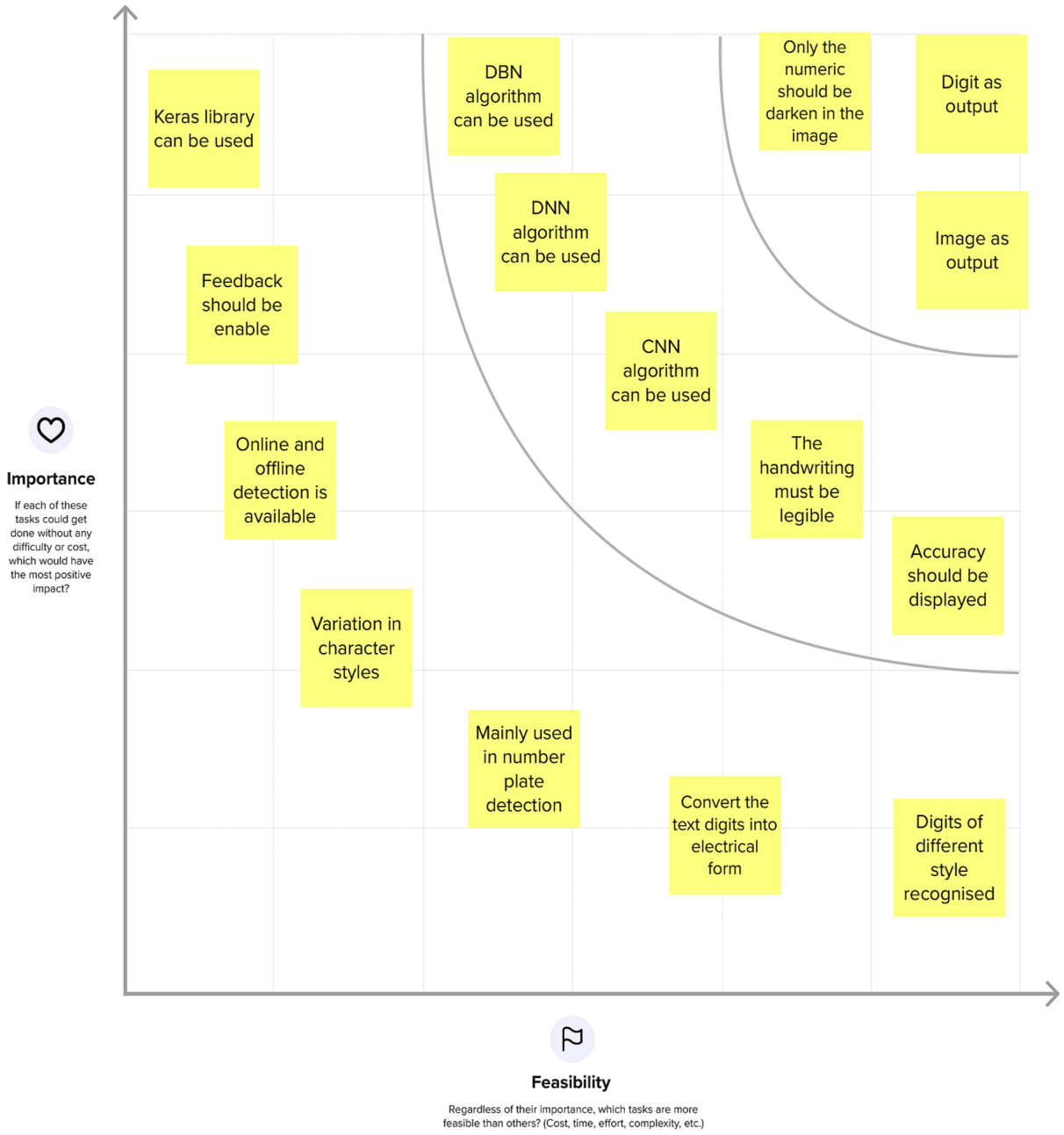


4

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes



3.3 PROPOSED SOLUTION

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	● The handwritten digit recognition is the capability of computer applications to recognize the human handwritten digits. ● It is a hard task for the machine because handwritten digits are not perfect and can be made with many different shapes and sizes. The handwritten digit recognition system is a way to tackle. ● It is easy for the human to perform task accurately by practicing it repeatedly and memorizing it for the next time. ● Human brain can process and analyse images easily. Also, recognize the different element present in the image ● It is a hard task for the machine because handwritten digits are not perfect and can be made with many different shapes and sizes ● This problem which uses the image of a digit and recognizes the digit present in the image ● In this competition, the goal is to correctly identify digits from a dataset of tens of thousands of handwritten images and experiment with different algorithms to learn what works well and how techniques compare
2.	Idea / Solution description	● The algorithm used is Convolution Neural Network(CNN). This will prepare the trained model which will be used to classify the digits present in the test data. Thus, we can classify the digits present in the images as: Class 0,1,2,3,4,5,6,7,8,9. ● MNIST is a dataset which is widely used for handwritten digit recognition. The dataset consist of 60,000 training images and 10,000 test images ● The artificial neural networks can all most mimic the human brain and are a key ingredient in image

		processing field.
3.	Novelty / Uniqueness	● This project introduces an operative strategy for dealing with novelty in the handwritten visual recognition domain. A perfect transcription agent would be able to distinguish known and unknown characters in a picture, as well as determine any aesthetic variations that may occur inside or between texts. The existence of novelty has shown to be a major stumbling block for even the most robust machine learning-based algorithms for these activities. ● Novelty in handwritten papers might include, among other things, a change in the writer, character properties, writing attributes, or overall document appearance. Instead of examining each element separately, we believe that an integrated agent capable of processing known characters and novelties concurrently is a superior technique. The handwritten digit recognition problem can be seen as a subtask of the optical character recognition (OCR) problem.
4.	Social Impact / Customer Satisfaction	● There are many benefits associated with the handwriting recognition system. In addition to reading postal addresses and bank check amounts, it is also useful for reading forms. Furthermore, it's used in fraud detection because it makes it easy to compare two texts and determine which one is a copy. As a result, this system fulfils customers' expectations, as it is a novel method for recognizing handwritten digits, ensuring high accuracy for the model and meeting all customer expectations. Users will save a lot of time and effort if the system provides various synonyms for the words recognized. Due to the fact that the users in rural areas will be using their own regional language, this proposed system should be able to detect those digits as well. As the system is being used in socially crowded places such as banks to check amounts, it should be

		fast and reliable. As it is designed to solve real-world problems, it should be highly reliable and trustworthy in every way, and users throughout the world should be able to use it effectively.
5.	Business Model (Revenue Model)	● The applications where these handwritten digit recognition can be used are Banking sector where it can be used to maintain the security pin numbers, it can be also used for blind peoples by using sound output. ● Some of the research areas include signature verification, bank check processing, postal address interpretation from envelopes etc
6.	Scalability of the Solution	● One of the approaches to make the handwritten digit recognition system scalable is to make use of cloud-native methods. For example, one of the cloud solutions for making AI scalable is IBM Cloud. IBM Cloud Build helps run and manage AI models, optimize decisions at scale across any cloud. The advantage of using cloud to make solutions scalable is that we can deploy our AI application on the specific cloud environment that best supports our business needs. We can take advantage of built-in security capabilities and AI model monitoring. We can Automate AI lifecycles with ModelOps pipelines, deploy and run models through one click integration and also prepare and build models visually and programmatically. Looking at these advantages, we can drive better business outcomes by optimizing our decisions and also make our solution scalable using cloud

3.4 PROBLEM SOLUTION FIT\

CHAPTER 4

REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	The product essentially converts handwritten digits to digital form.	The user is first asked to draw a number on the canvas and the model that is built is then utilized to compare the data to provide an output in digitized form.
FR-2	Recognizing the handwritten digit and displaying	Recognizing the handwritten digit and displaying
FR-3	Import the dataset file directly to the program from a command that will download the dataset from its website.save the dataset file in the same the directory as the program	Installing packages and applications.
FR-4	Build a neural network with a number of nodes in the input layer equal to the number of pixels in the arrays	Nil.
FR-5	Activating the neural network.	Packages -tensorflow.

4.2 NON-FUNCTIONAL REQUIREMENTS

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	System design should be easily understood and user friendly to users. Furthermore, users of all skill levels should be able to navigate it without problems.
NFR-2	Security	The system should automatically be able to authenticate all users with their unique username and password.
NFR-3	Reliability	It should be user-friendly.
NFR-4	Performance	Should reduce the delay in information when hundreds of requests are given.
NFR-5	Availability	Information is restricted to each user's limited access.
NFR-6	Scalability	The system should be able to handle 5000 users accessing the site at the same time.

CHAPTER 5

PROJECT DESIGN

5.1 DATA FLOW DIAGRAM

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.

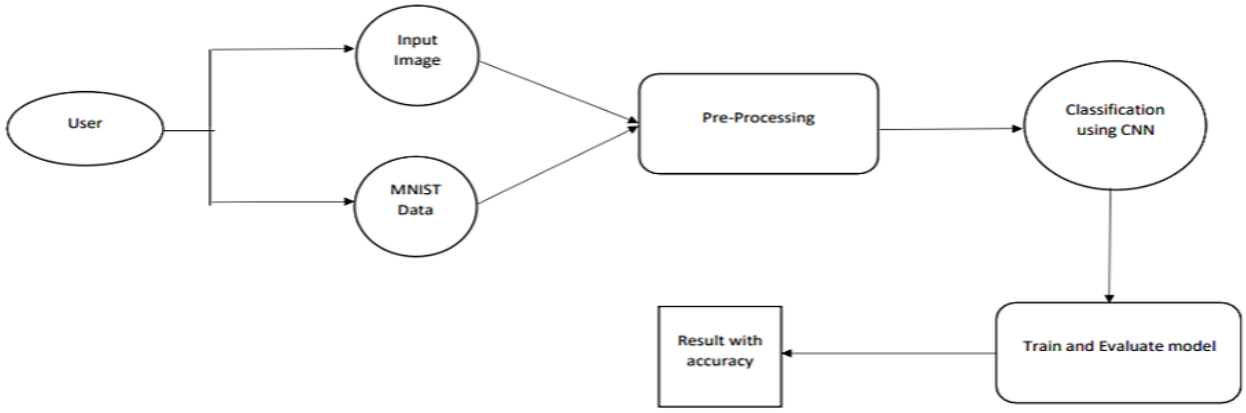


Fig.5.1Data Flow Diagrams

5.2 SOLUTION AND TECHNICAL ARCHITECTURE

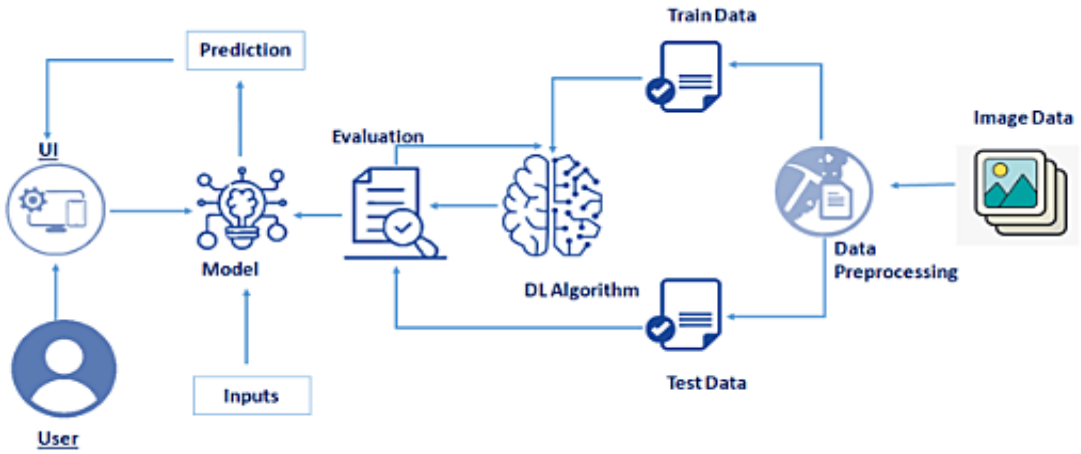


Fig.5.2 Architecture of the Handwritten digit recognition system

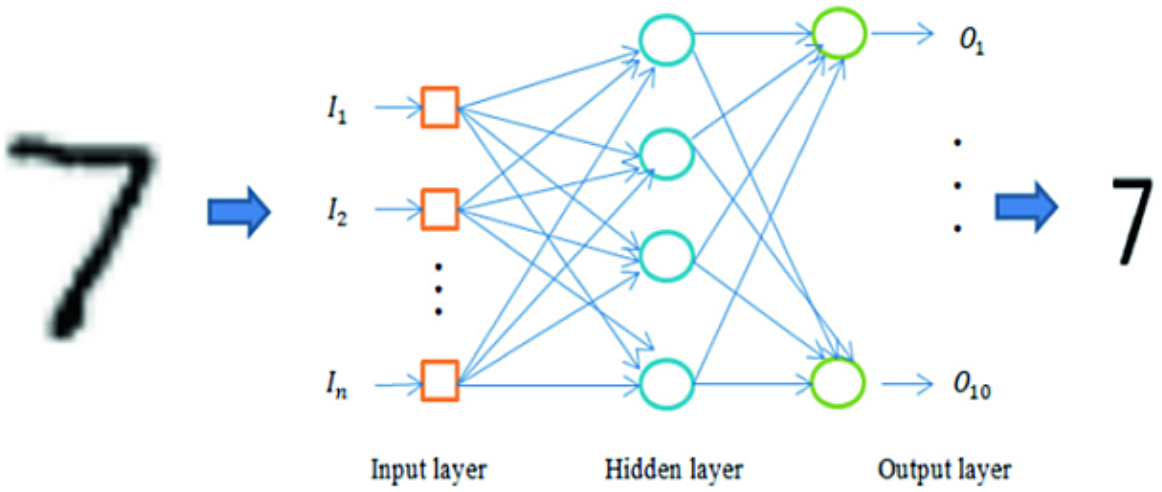


Fig.5.3 A Novel Approach for Handwritten Digit Recognition Using Multilayer Perception Neural Network

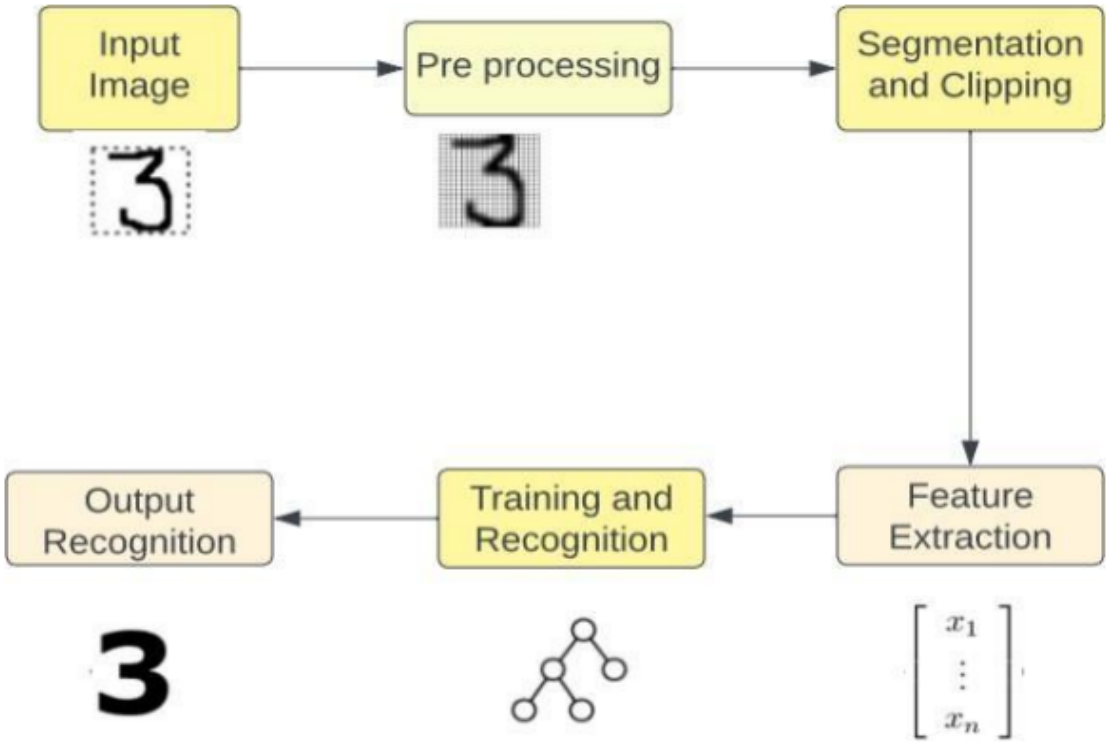


Fig.5.4 Model Architecture of the Deep Learning model used for Handwritten digit recognition

5.3 USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for a unique email id and password.	I can download the registration	High	Sprint-1
		USN-2	As a user, I will receive a confirmation email once I have registered for the application.	I can receive a confirmation email and click confirm.	Low	Sprint-2
		USN-3	As a user, I can sign in to an application.	I can sign in to the application	Medium	Sprint-2
	Upload	USN-4	As a user I can upload my handwritten digits images to be recognised from the device	I can choose any image from my device.	High	Sprint-3
	Recognize	USN-5	As a user, I will train and test the input to get the maximum accuracy of output	I am able to train and test the application until it gets maximum accuracy of the result	High	Sprint-4
Customer (Web user)	Registration	USN-6	As a user, I can register for a unique email id and password.	I can download the registration	High	Sprint-1

		USN-7	As a user, I will receive a confirmation email once I have registered for the application.	I can receive a confirmation email and click confirm.	Low	Sprint-2
		USN-8	As a user, I can sign in to an application.	I can sign in to the application	Medium	Sprint-2
	Home	USN-9	As a user, I can view the instructions and process of the application.	I can view the instructions and process of the application.	Low	Sprint-2
	Upload	USN-10	As a user I can upload my handwritten digits images to be recognised from the device	I can choose any image from my device.	High	Sprint-3
	Recognize	USN-11	As a user, I will train and test the input to get the maximum accuracy of output	I am able to train and test the application until it gets maximum accuracy of the result	High	Sprint-4
	Help	USN-9	As a user i can contact to administration	I can access the ask help to administrator	Low	Sprint-2

CHAPTER 6

PROJECT PLANNING AND 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	R.Suvin Kumar
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	V.Maruthu pandiyan
Sprint-2		USN-3	As a user, I can register for the application through Facebook	2	Low	T.Kiruthika
Sprint-1		USN-4	As a user, I can register for the application through Gmail	2	Medium	S.Umesh
Sprint-1	Login	USN-5	As a user, I can log into the application by entering email & password	1	High	S.Sneka
Sprint-3	Dashboard	USN-6	As a user, I can upload image using camera	3	Medium	V.Maruthu pandiyan

Sprint-3		USN-7	As a user, I can upload image from gallery	3	Medium	S.Sneka
Sprint-4		USN-8	As a user, I can download to the recognized digit	5	High	S.Umesh
Sprint-2	Help	USN-9	As a user i can contact to administration	1	Low	M.Kowsalya

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	6	6 Days	24 Oct 2022	29 Oct 2022	6	29 Oct 2022
Sprint-2	3	6 Days	31 Oct 2022	05 Nov 2022	3	05 Nov 2022
Sprint-3	6	6 Days	07 Nov 2022	12 Nov 2022	6	12 Nov 2022
Sprint-4	5	6 Days	14 Nov 2022	19 Nov 2022	5	19 Nov 2022

Velocity:

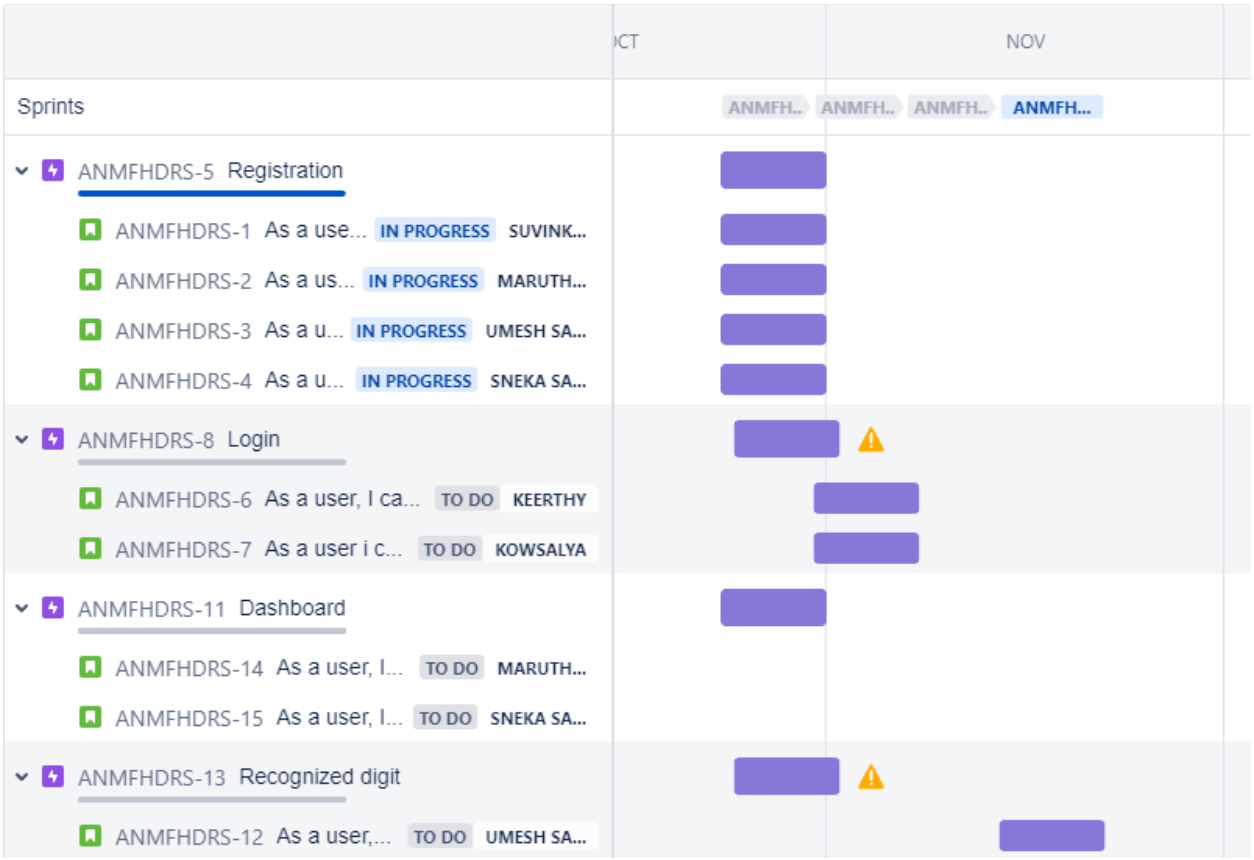
Imagine we have a 6-day sprint duration, and the velocity of the team is 20 (points per sprint). Let’s calculate the team’s average velocity (AV) per iteration unit (story points per day)

Average Velocity = Sprint Duration/Velocity = 06/06 = 1

Burn down 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.

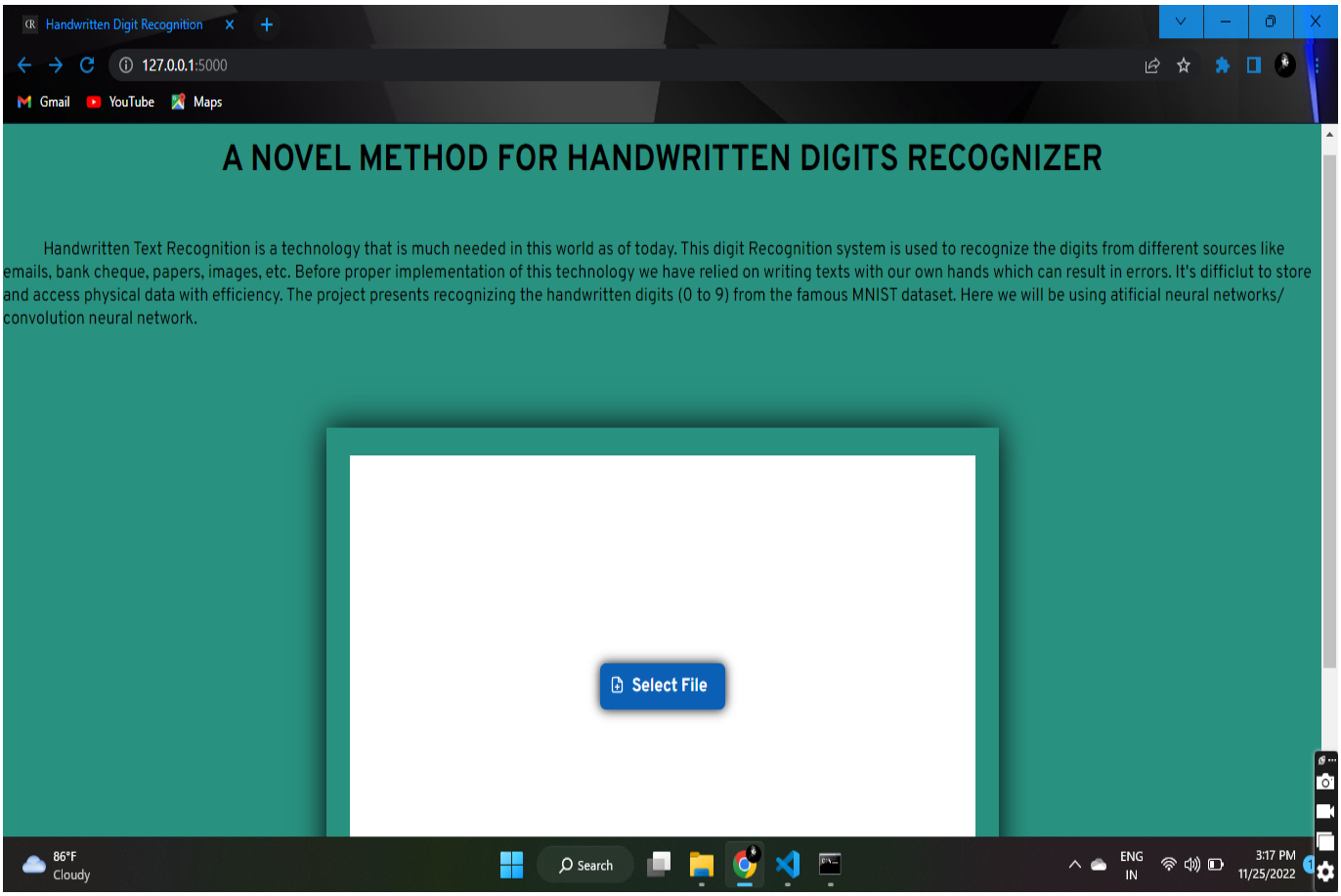
6.3 REPORT FROM JIRA



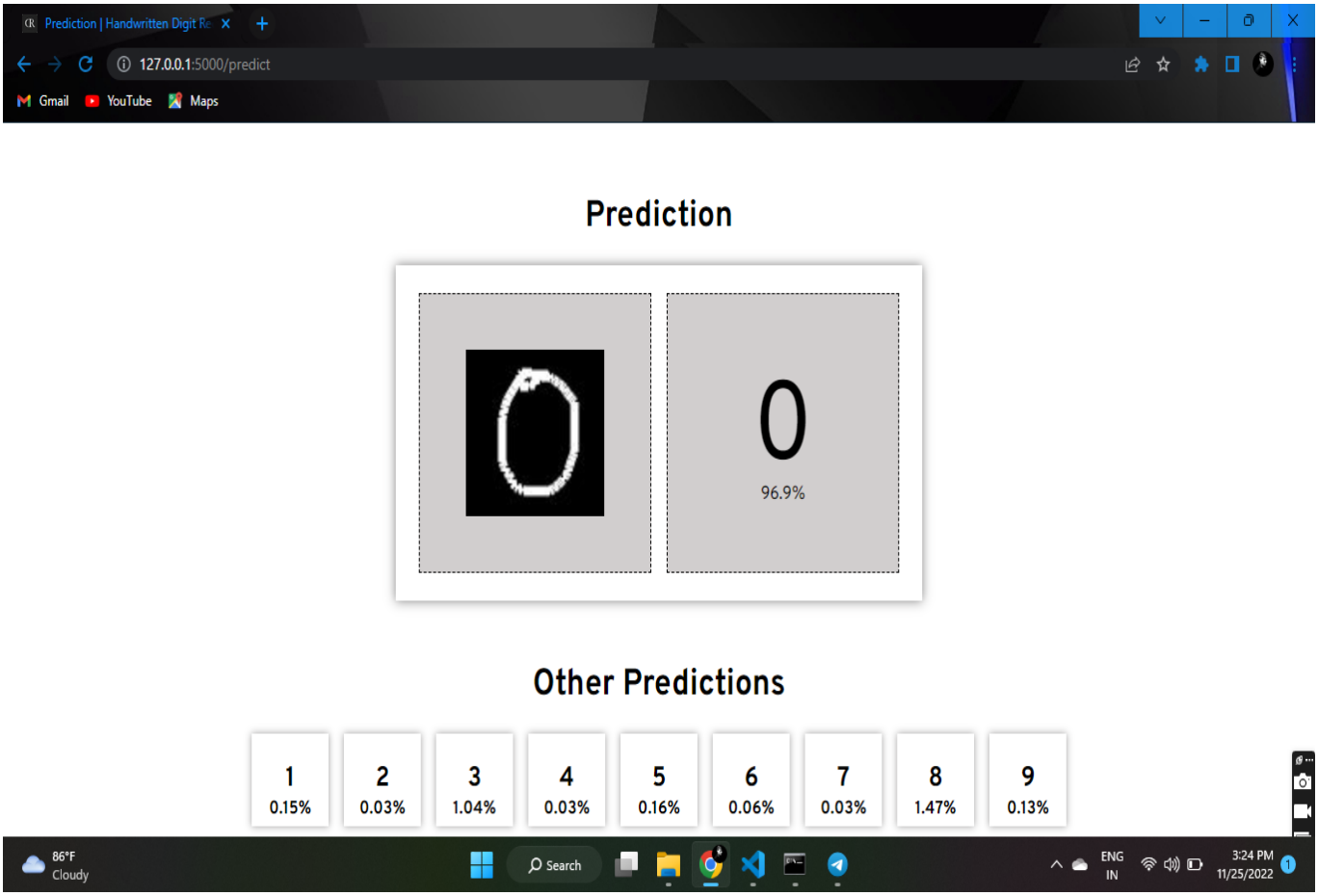
CHAPTER 7

CODING & SOLUTIONING

7.1 FEATURE



7.2 FEATURE


Prediction | Handwritten Digit Re
127.0.0.1:5000/predict
Gmail
YouTube
Maps
Prediction
0
96.9%
Other Predictions
1
0.15%
2
0.03%
3
1.04%
4
0.03%
5
0.16%
6
0.06%
7
0.03%
8
1.47%
9
0.13%
86°F
Cloudy
Search
ENG
IN
3:24 PM
11/25/2022

CHAPTER 8

TESTING

8.1 TEST CASES

TEST CASE ID	FEATURE TYPE	COMPONENT	TEST SCENARIO	EXPECTED RESULT	ACTUAL RESULT	STATUS
TC_001	UI	Home Page	Verify UI elements in the Home Page	The Home Page must be displayed properly	Working as expected	PASS
TC_002	UI	Home Page	Verify whether the page is responsive	The Home Page must display in the same way in all devices	The UI is displayed correctly only on the desktop screens	FAIL
TC_003	Functional	Home Page	Check if user could navigate to the next page	The button in the Home Page is directing to next page	Working as expected	PASS
TC_004	Functional	Backend	Check if all the routes are working properly	All the routes should properly work	Working as expected	PASS
TC_005	Functional	Model	Check if the model can handle various image sizes	The model should rescale the image and predict the results	Working as expected	PASS
TC_006	Functional	Model	Check if the model predicts the digit	The model should predict the number	Working as expected	PASS

TC_007	Functional	Model	Check if the model can handle complex input image	The model should predict the number in the complex image	The model fails to identify the digit since the model is not built to handle such data	FAIL
TC_008	Functional	Prediction Page	Reports error if files are not uploaded	Prediction Page pops out error page if file is not uploaded	Working as expected	PASS
TC_009	UI	Prediction Page	Verify UI elements in the Prediction Page	The Prediction page must be displayed properly	Working as expected	PASS
TC_010	UI	Prediction Page	Check if the result is displayed properly	The result should be displayed properly	Working as expected	PASS

8.2 USER ACCEPTANCE TESTING

Acceptance Testing is a level of the software testing where a system is tested for acceptability. The purpose of this test is to evaluate the system's compliance with the business requirements and assess whether it is acceptable for delivery. Formal testing with respect to user needs, requirements, and business processes conducted to determine whether or not a system satisfies the acceptance criteria and to enable the user, customers or other authorized entity to determine whether or not to accept the system. In this application, the customer's acceptance is been monitored and it is been put into usage.

8.2.1 TEST CASE ANALYSIS

SECTION	TOTAL CASES	NOT TESTED	FAIL	PASS
Client Application	5	0	1	4
Security	1	0	0	1
Performance	3	0	1	2
Exception Reporting	1	0	0	1

CHAPTER 9

RESULTS

9.1 PERFORMANCE METRICS

Script: locust.py

Request Statistics

Method	Name	# Requests	# Fails	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	RPS	Failures/s
GET	/	1043	0	13	4	290	1079	1.9	0.0
GET	/predict	1005	0	39648	385	59814	2670	1.8	0.0
	Aggregated	**2048**	**0**	**19462**	**4**	**59814**	**1859**	**3.7**	**0.0**

Response Time Statistics

Method	Name	50%ile (ms)	60%ile (ms)	70%ile (ms)	80%ile (ms)	90%ile (ms)	95%ile (ms)	99%ile (ms)	100%ile (ms)
GET	/	10	11	13	15	19	22	62	290
GET	/predict	44000	46000	47000	48000	50000	52000	55000	60000
	Aggregated	**36**	**36000**	**43000**	**45000**	**48000**	**50000**	**54000**	**60000**

CHAPTER 10

ADVANTAGES AND DISADVANTAGES

10.1 ADVANTAGES

- Reduces manual work
- Can recognize the digits more accurately than humans
- Application is capable of handling a lot of data
- Application can be used by bank officials, postal officers, traffic police, etc.

10.2 DISADVANTAGES

- All the data must be in digital format
- Requires a high-performance server for faster predictions
- Prone to occasional errors
- Cannot handle complex data

CHAPTER 11

CONCLUSION

This project demonstrates a web application build using HTML, CSS, JS, Flask and few other technologies. Each time the user uploads an image for recognizing, the image is pre-processed before feeding it into the model. After pre-processing, the image is fed into the CNN model for recognizing. In the CNN model, the pre-processed image passes into various layers and finally the model recognizes the digit. The output is being rendered into the web application and shown to the user. This application can be used in various domains for recognizing the digits. For example, by the police to track the vehicle number, postal officer to identify the zip codes or bank officials to recognize the digits on bank leaf.

CHAPTER 12

FUTURE SCOPE

- Add support to detect multiple digits
- Add support to different languages to help users all over the world
- Add support to detect digits from multiple images
- Improve model to convert the textual output into audio format

CHAPTER 13

APPENDIX

13.1 SOURCE CODE

app.py

app.py predict.css main.css recognizer.cpython-310.pyc

app.py > ...

```
1   from flask import Flask,render_template,request
2   from recognizer import recognize
3
4   app=Flask(__name__)
5
6   @app.route('/')
7   def main():
8       return render_template("home.html")
9
10
11  @app.route('/predict',methods=['POST'])
12  def predict():
13      if request.method=='POST':
14        image = request.files.get('photo', '')
15        best, others, img_name = recognize(image)
16        return render_template("predict.html", best=best, others=others, img_name=img_name)
17
18
19  if __name__=="__main__":
20      app.run()
21
```

home.html

templates > home.html > html > head > meta

```
<html>
    <head>
        <meta name="viewport" content="width=device-width, initial-scale=1.0" />
        <title>Handwritten Digit Recognition</title>
        <link rel="icon" type="image/svg" sizes="32x32" href="{{url_for('static',filename='images/icon.svg')}}"
        <link rel="stylesheet" href="{{url_for('static',filename='css/main.css')}}" />
        <script src="https://unpkg.com/feather-icons"></script>
        <script defer src="{{url_for('static',filename='js/script.js')}}"></script>
    </head>

    <body>
<h1 class="heading__main">A NOVEL METHOD FOR HANDWRITTEN DIGITS RECOGNIZER</h1>
                <p><br>           Handwritten Text Recogr
        <div class="container">

            <div class="upload-container">
                <div class="form-wrapper">
                    <form class="upload" action="/predict" method="post" enctype="multipart/form-data">
                        <label id="label" for="upload-image"><i data-feather="file-plus"></i>Select File</label>
                        <input type="file" name="photo" id="upload-image" hidden />
                        <button type="submit" id="up_btn"></button>
                    </form>
                    <img id="loading" src="{{url_for('static',filename='images/loading.gif')}}">
                </div>
            </div>
        </div>
    </body>
</html>

```

predict.html

templates > predict.html > html

```
<html>
    <head>
        <title>Prediction | Handwritten Digit Recognition</title>
        <link rel="stylesheet" href="{{url_for('static',filename='css/predict.css')}}" />
        <link rel="icon" type="image/svg" sizes="32x32" href="{{url_for('static',filename='images/icon.svg')}}"
        <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    </head>
    <body>
        <div class="container">
            <h1>Prediction</h1>
            <div class="result-wrapper">
                <div class="input-image-container">
                    <img src="{{url_for('static',filename='data/')}}{{img_name}}" />
                </div>
                <div class="result-container">
                    <div class="value">{{best.0}}</div>
                    <div class="accuracy">{{best.1}}%</div>
                </div>
            </div>
            <h1>Other Predictions</h1>
            <div class="other_predictions">
                {% for x in others %}
                <div class="value">
                    <h2>{{x.0}}</h2>
                    <div class="accuracy">{{x.1}}%</div>
                </div>
                {% endfor %}
            </div>
        </div>
    </body>
</html>
```

css main

static > css > # main.css > ...

```css
@import url("https://fonts.googleapis.com/css2?family=Overpass:wght@200;300;400;500;600;700;900&display=swap");

* {

    padding: 0;
    margin: 0;
}

body {
    color: black;
        font-family: "Overpass", sans-serif;
        background-color: #299180;
}
h1{
    width: 100%;
    height: 15%;
    display: flex;
    flex-direction: column;
    justify-content: center;
    align-items: center;
}
.container {
    width: 100%;
    height: 100%;
    display: flex;
    flex-direction: column;
    justify-content: center;
    align-items: center;

}

.heading {
```

app.py predict.html predict.css main.css

static > css > # main.css > ...

```
.heading {
    margin-top: -2rem;
    padding-bottom: 2rem;
    width: fit-content;
    text-align: center;
}

.heading .heading__main {
    font-size: 3rem;
    font-weight: 550;
    font-style: italic;
}

.heading .heading__sub {
    font-size: 1rem;
    color: inherit;
    font-style: unset;
}

.upload-container {
    box-shadow: 0 0 30px black;
    width: 40rem;
    height: 25rem;
    padding: 1.5rem;
}

.form-wrapper {
    background-color: white;
    width: 100%;
    height: 100%;
    display: flex;
    border: 1px  black;
```

static > css > # main.css > ...

```
64      justify-content: center;
65      align-items: center;
66
67  }
68
69  .form-wrapper #loading {
70      display:none;
71      position: absolute;
72  }
73
74  .form-wrapper .upload {
75      display: flex;
76      justify-content: center;
77      align-items: center;
78      width: 8rem;
79      height: -webkit-fit-content;
80      height: -moz-fit-content;
81      height: fit-content;
82      border-radius: 6px;
83      color: white;
84      background-color: rgb(11, 96, 182);
85      box-shadow: 0 0 10px black;
86  }
87
88  .form-wrapper .upload #up_btn {
89      display: none;
90  }
91
92  .form-wrapper .upload label {
93      font-size: 1rem;
94      font-weight: 600;
95      color: white;
```

static > css > # main.css > ...

```
96          height: 100%;
97          width: 100%;
98          padding: 10px;
99          display: block;
100     }
101
102     .form-wrapper .upload svg {
103         height: 15px;
104         width: auto;
105         padding-right: 8px;
106         margin-bottom: -2px;
107     }
108
109     @media screen and (max-width: 700px) {
110         .upload-container {
111             height: 20rem;
112             width: 18rem;
113             margin-top: 3.5rem;
114             margin-bottom: -8rem;
115         }
116
117         .heading .heading__main {
118             margin-top: -6rem;
119             font-size: 2rem;
120             padding-bottom: 1rem;
121         }
122     }
```

css predict

static > css > # predict.css > ...

```css
@import url("https://fonts.googleapis.com/css2?family=Overpass:wght@200;300;400;500;600;700;900&display=swap");

body {
    color: black;
    font-family: "Overpass", sans-serif;
}

h1 {
    padding-top: 2rem;
}

.container {
    display: flex;
    justify-content: center;
    align-items: center;
    flex-direction: column;
}

.result-wrapper {
    width: -webkit-fit-content;
    width: -moz-fit-content;
    width: fit-content;
    height: -webkit-fit-content;
    height: -moz-fit-content;
    height: fit-content;
    box-shadow: 0 0 10px rgb(126, 125, 125);
    padding: 1.5rem;
    display: flex;
    justify-content: center;
    align-items: center;
    -moz-column-gap: 1rem;
    column-gap: 1rem;
```

app.py | predict.html | # predict.css | # main.css

static > css > # predict.css > ...

```
}

.result-wrapper .input-image-container,
.result-wrapper .result-container {
    width: 15rem;
    height: 15rem;
    border: 1px dashed black;
    justify-content: center;
    display: flex;
    align-items: center;
    flex-direction: column;
    background-color: rgb(209, 206, 206);
}

.result-wrapper .input-image-container img {
    width: 60%;
    height: 60%;
    background-color: aqua;
    background-size: contain;
}

.result-wrapper .result-container .value {
    font-size: 6rem;
}

.result-wrapper .result-container .accuracy {
    margin-top: -1rem;
}

.other_predictions {
    display: flex;
    justify-content: center;
```

app.py predict.html # predict.css # main.css

static > css > # predict.css > ...

```
    align-items: center;
    flex-wrap: wrap;
    column-gap: 1rem;
    row-gap: 1rem;
    font-weight: 700;
}

.other_predictions .value {
    display: flex;
    justify-content: center;
    align-items: center;
    flex-direction: column;
    width: 5rem;
    height: 5rem;
    box-shadow: 0 0 7px rgb(158, 157, 157);
}

.other_predictions .value div {
    margin-top: -1.2rem;
}

@media screen and (max-width: 700px) {
    h1 {
        font-size: 2.3rem;
    }

    .result-wrapper .input-image-container,
    .result-wrapper .result-container {
        width: 7rem;
        height: 7rem;
    }

```

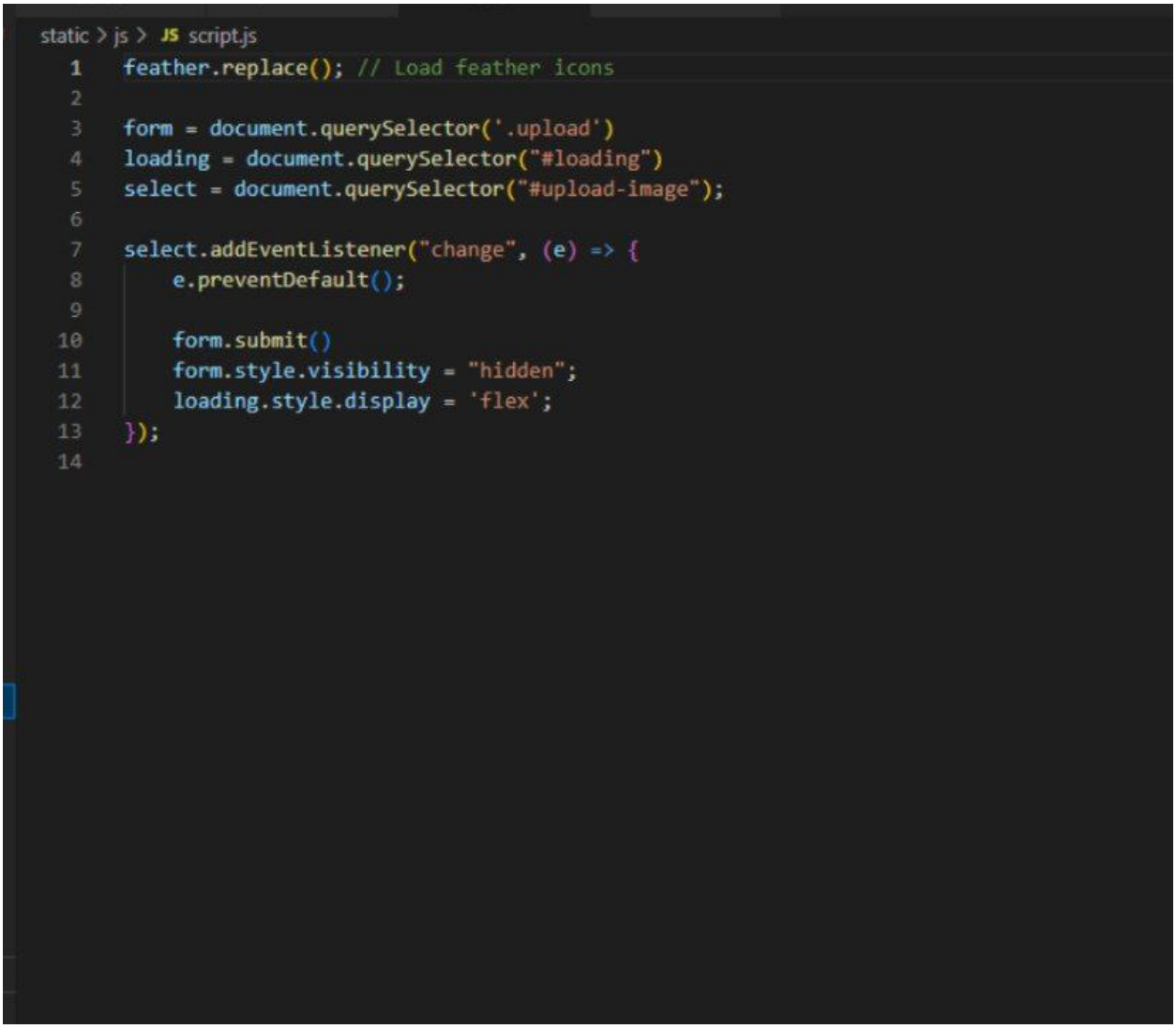
app.py predict.html # predict.css # main.css

static > css > # predict.css > ...

```
        .result-wrapper .result-container .value {
            font-size: 4rem;
        }
    }

```

css js



13.2 GITHUB AND PROJECT DEMO LINK

Github link

https://github.com/IBM-EPBL/IBM-Project-54910-1663054938

Project demo link

https://www.youtube.com/embed/fLOGhP6tRrA