A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

SUBMITTED BY

PNT2022TMID39636

TEAM LEADER :

PAVITHRA K

TEAM MEMBERS:

DHANUSHA K
HARITHA THARINI S
KEERTHANA C
HEMAPRIYA S
ISHWARYA V

TABLE OF CONTENTS

CHAPTER-1

INTRODUCTION

PROJECT OVERVIEW

Handwritten digit recognition is the ability of a computer to recognize the human handwritten digits from different sources like images, papers, touch screens, etc, and classify them into 10 predefined classes (0-9).

Immediate applications of the digit recognition techniques include postal mail sorting automatically address reading and mail routing, bank check processing, etc.

PURPOSE

Digit recognition has many applications like number plate recognition, postal mail sorting, bank check processing,etc.This project provides deep learning algorithms for handwritten digit recognition.

One of the major challenges in the Handwritten Digit Recognition in class variance because people do not always write the same digit in exactly the same way.

Many feature extraction approaches have been proposed trying to characterize the shape invariance within a class to improve the discrimination ability.

In this report we train and test a set of classifiers on the MNIST database for pattern analysis in solving the handwritten digit recognition problem.

CHAPTER-2

LITERATURE SURVEY

EXISTING PROBLEM

The fundamental problem with handwritten digit recognition is that handwritten digits do not always have the same size, width, orientation, and margins since they vary from person to person. Additionally, there would be issues with identifying the numbers because of similarities between numerals like 1 and 7, 5 and 6, 3 and 8, 2 and 5, 2 and 7, etc. Finally, the individuality and variation of each individual's handwriting influence the structure and appearance of the digits.

REFERENCES

Survey 1:

Ali Abdullah Yahya , Jieqing Tan and Min Hu

"A Novel Handwritten Digit Classification System Based on Convolutional Neural Network Approach"

An enormous number of CNN classification algorithms 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 short comings of these algorithms, this 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 them to select a typical filter size which leads to enhancing the classification accuracy of 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, they propose to add an additive white Gaussian noise with $s = 0.5$ to the MNIST dataset. As a result, the 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.

Survey 2:

Fengjun Guo , Shijie Chen

"Gesture Recognition Techniques in Handwriting Recognition Application"

Handwriting-gesture recognition has been widely implemented in handwriting input application. Usually gestures are used to conduct edit operations or be set as short cut of an application. In this paper, they compare several handwriting-gesture recognition methods, and address their different user cases. These methods include pixel-matching method, rule based method and discriminant-function based method. For discriminant-function based method, they describe 2 sub-methods. They are prototypes based method and training based method. They not only analyze recognition accuracy of gestures for these methods, but also analyze their distinguished capability when recognizing gestures and alphanumeric in same recognizing mode. Experiments results show that, if the gesture-samples are enough, training based method achieves the highest accuracy. Furthermore, when recognizing mixed input of gestures and other handwriting symbols, training based met.

Survey 3:

Malothu Nagu, n Vijay Shankar, k.Annapurna

"A novel method for Handwritten Digit Recognition with Neural Networks"

Character recognition plays an important role in the modern world. It can solve more complex problems and makes humans' job 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 is also discussed. Bayesian Decision theory, Nearest Neighbor rule, and Linear Classification or Discrimination is types of methods for Pattern Recognition. Shape recognition, Chinese Character and Handwritten Digit recognition uses Neural Network to recognize them. Neural Network is used to train and identify written digits. After training and testing, the accuracy rate reached 99%.

Survey4:

Vladimir I. Pavlovic, Student Member, IEEE Rajeev Sharma, Member, IEEE, and Thomas S. Huang, Fellow, IEEE

"Visual Interpretation of Hand Gestures for Human-Computer Interaction"

The use of hand gestures provides an attractive alternative to cumbersome interface devices for human-computer interaction (HCI). In particular, visual interpretation of hand gestures can help in achieving the ease and naturalness desired for HCI. This has motivated a very active research area concerned with computer vision-based analysis and interpretation of hand gestures. They survey the literature on visual interpretation of hand gestures in the context of its role in HCI. This discussion is organized on the basis of the method used for modeling, analyzing, and recognizing gestures. Important differences in the gesture interpretation approaches arise depending on whether a 3D model of the human hand or an image appearance model of the human hand is used.3D hand models offer a way of more elaborate modeling of hand gestures but lead to computational hurdles that have not been overcome given the real-time requirements of HCI. Appearance-based models lead to computationally efficient "purposive" approaches that work well under constrained situations but seem to lack the generality desirable for HCI. They also discuss implemented gestural systems as well as other potential applications of vision-based gesture recognition. Although the current progress is encouraging, further theoretical as well as computational advances are needed before gestures can be widely used for HCI. They discuss directions of future research in gesture recognition, including its integration with other natural modes of human computer interaction.

REFERENCE:

1. Ali Abdullah Yahya , Jieqing Tan and Min Hu
"A Novel Handwritten Digit Classification System Based on Convolutional Neural Network Approach"

2.Fengjun Guo, Shijie Chen
"Gesture Recognition Techniques in Handwriting Recognition Application"

3.Malothu Nagu, N Vijay Shankar , K. Annapurna
"A novel method for Handwritten Digit Recognition with Neural Networks"

4.Vladimir I. Pavlovic, Student Member, IEEE Rajeev Sharma, Member, IEEE and Thomas S. Huang, Fellow, IEEE
"Visual Interpretation of Hand Gestures for Human-Computer Interaction"

PROBLEM STATEMENT

The handwritten digit recognition is the capability of computer applications to recognize the human handwritten digits. It is a hard task for the machine because handwritten digits are not perfect and can be made with many different shapes and sizes. The handwritten digit recognition system is a way to tackle this problem which uses the image of a digit and recognizes the digit present in the image.

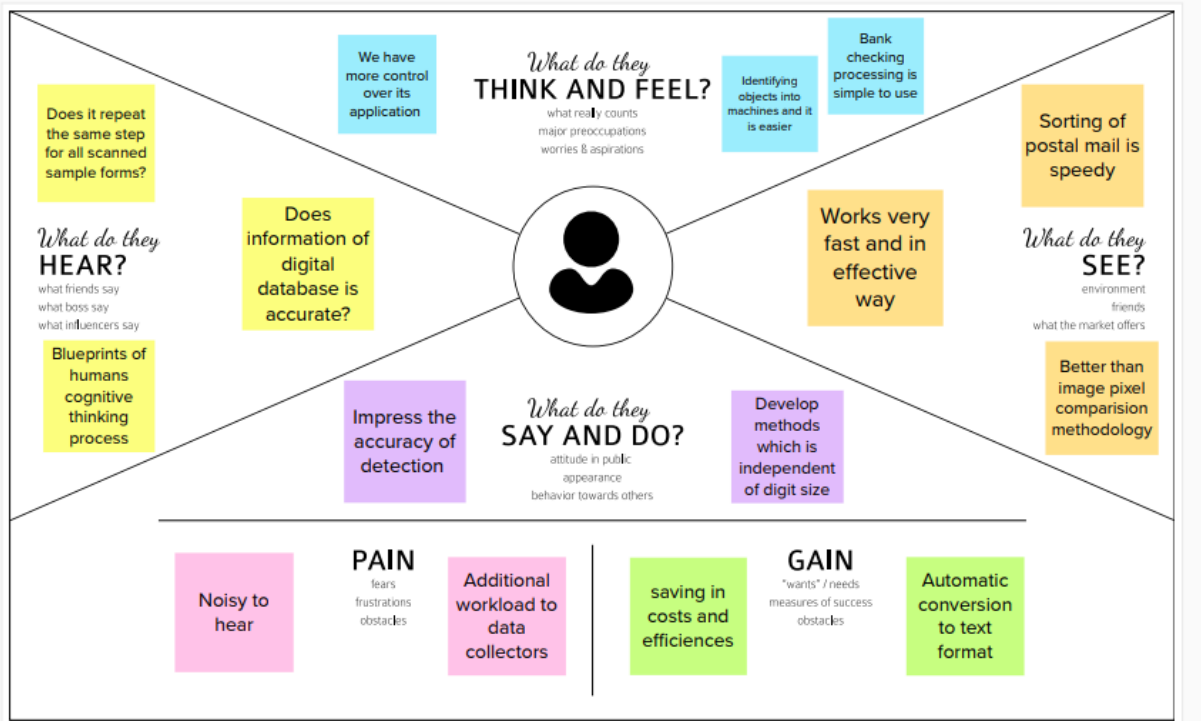
The problem is faced more when many people write a single digit with a variety of different handwritings. The uniqueness and variety in the handwriting of different individuals also influence the formation and appearance of the digits

The issue occurs when the person handwritten digits are not recognized by the machine. This makes it tricky to provide enough examples of how every character might look.

CHAPTER-3

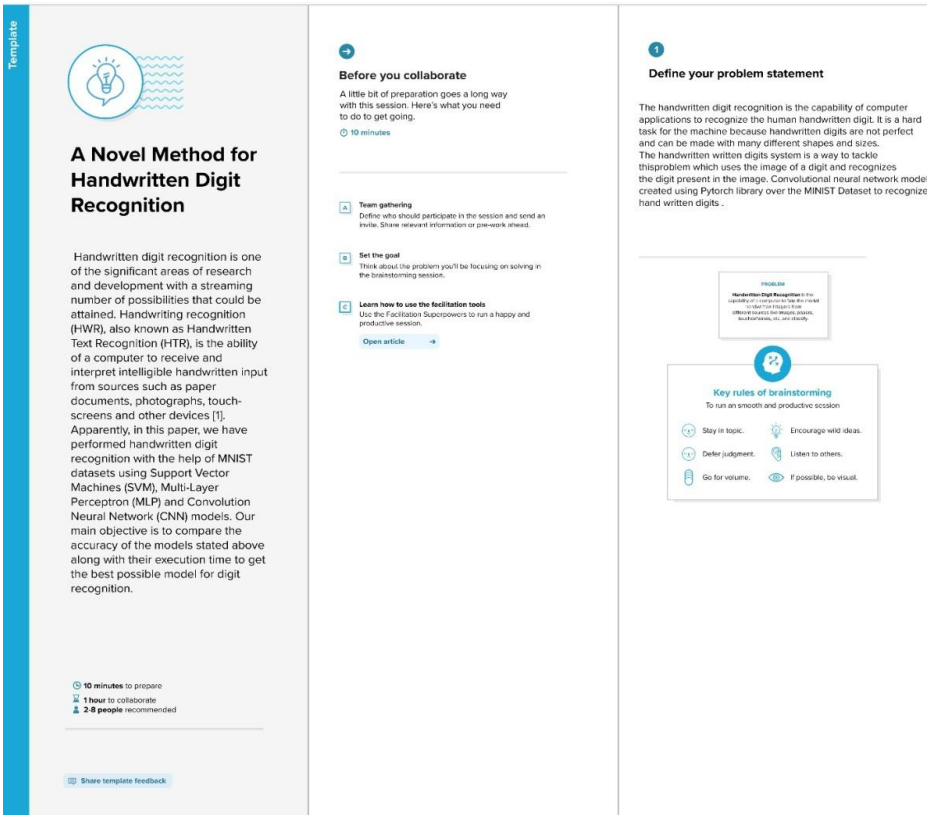
IDEATION AND PROPOSED SOLUTION

EMPATHY MAP CANVAS

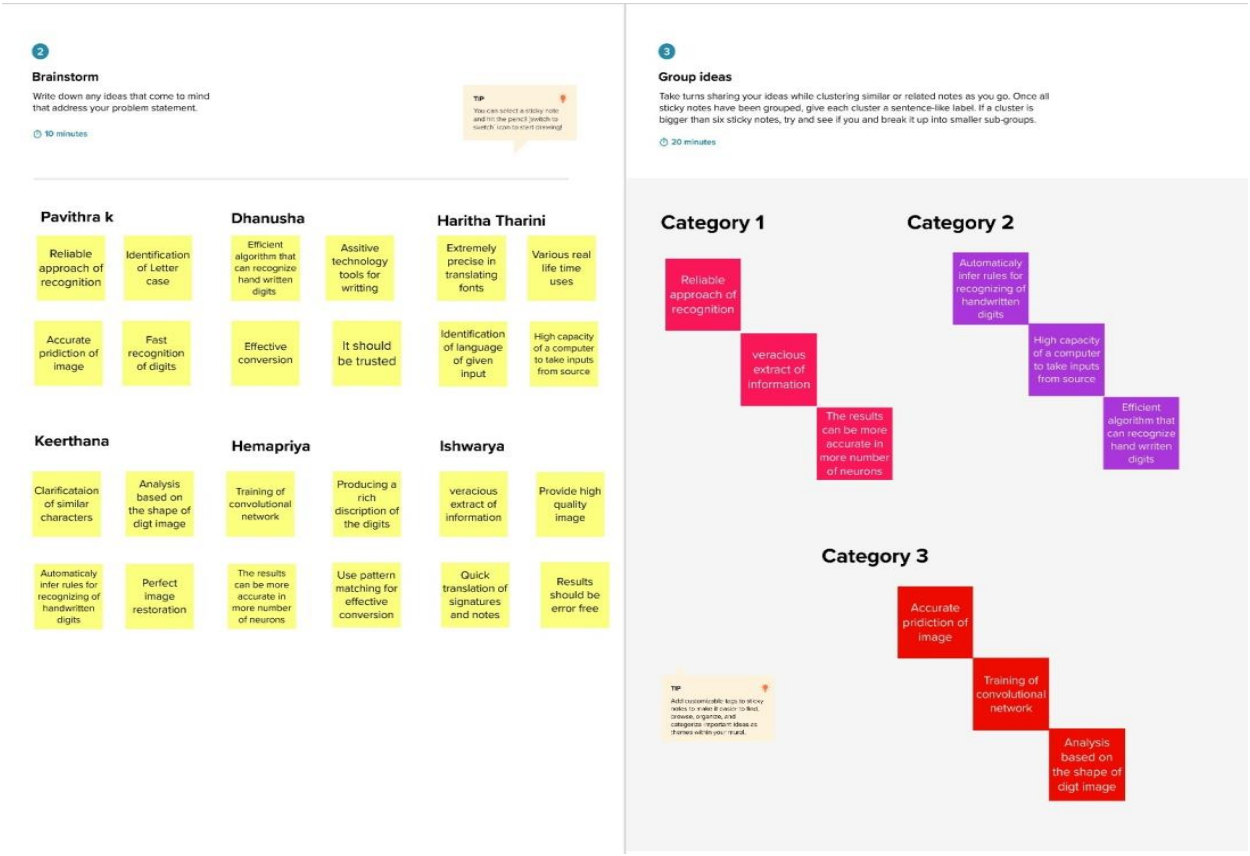


IDEATION AND BRAINSTORMING

Step-1: Team Gathering, Collaboration and Select the Problem Statement

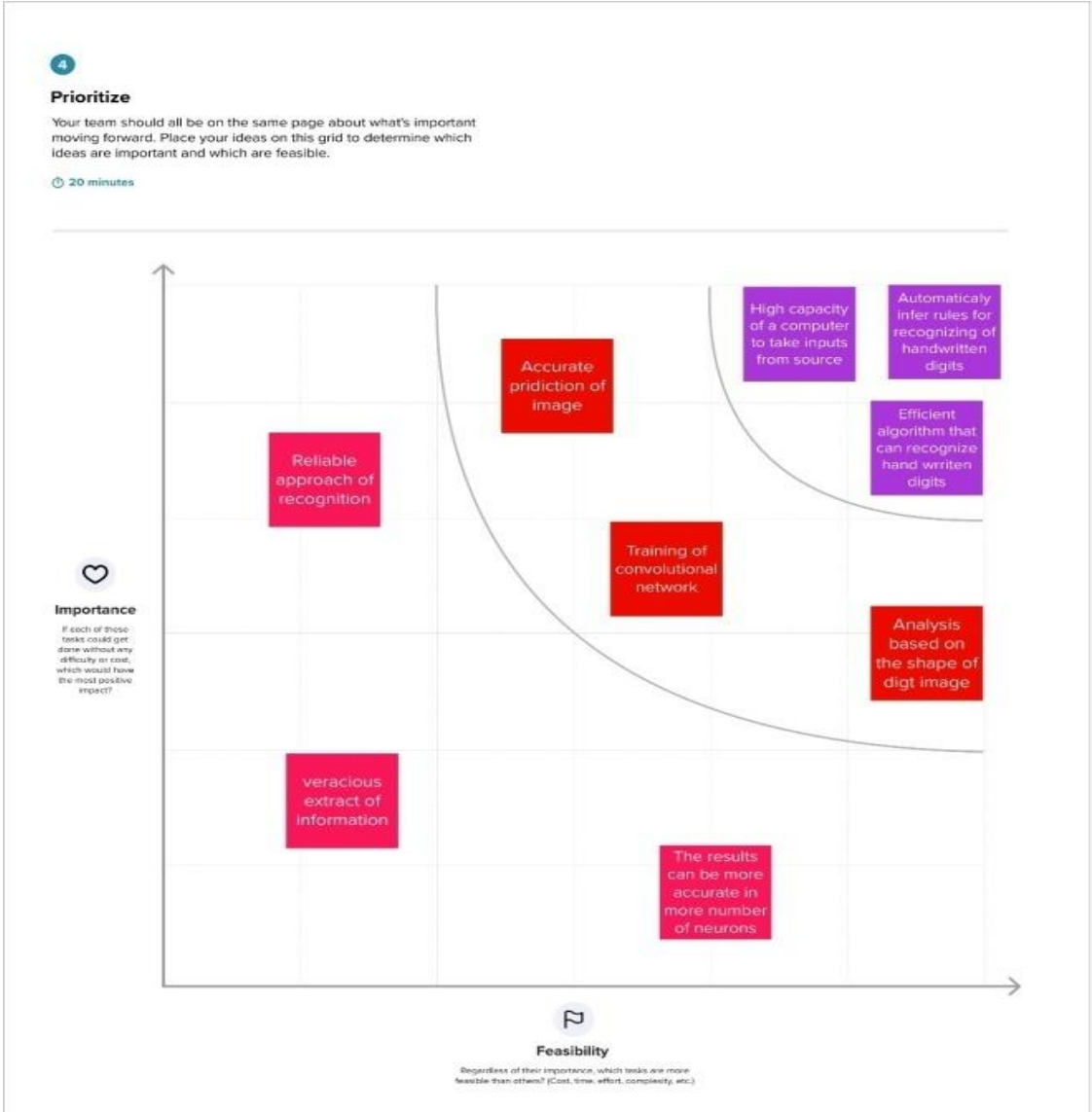


Step-2: Brainstorm, Idea Listing and Grouping



Step-3: Idea Prioritization

PROJECT DESIGN PHASE-I



PROPOSED SOLUTION

PROPOSED SOLUTION :

s.no.	parameter	Description
1.	Problem statement (Problem to be solved)	It is easy for the human to perform a task accurately by practicing it repeatedly and memorizing it for the next time. Human brain can process and analyze images easily. Also, recognize the different elements present in the images. In this project, the goal is to correctly identify digits from a dataset of tens of thousands of handwritten images and experiment with different algorithms to learn first-hand what works well and how techniques compare.
2.	Idea / solution description	The task of HDR is accomplished by using the CNN, incorporating a sequential CNN framework, with rectified linear units (RELU) activations that have never been reported.The goal is achieved by establishing a model that can recognize and determine the handwritten digits from its image with high accuracy and low computation time. We Aim to complete this by using the concepts of convolutional neural network. The proposed CNN framework is well equipped with suitable parameters for high accuracy of MNIST digit classification.

3.	Novelty / uniqueness	The handwritten digits are not always ofthe same size, width, orientation and justified to margins as they differ from writing of person to person. There is similarity between numbers. So, classifying Between these numbers is also a major problem for computers. The uniqueness and variety in the handwriting of different individuals also influence the formation and appearance of the digits
4.	Social impact / customersatisfaction	Character recognition plays an important role in the modern world. It can solve more complex problems and makes humans' job easier. An example is handwritten character recognition. This is a system widely used inthe 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 is also discussed.
5.	Business model (revenue model)	In recent days, artificial neural network (ANN) can be applied to a vast majority of fields including business, medicine, engineering, etc. The most popular areas where ANN is employed nowadays are pattern and sequence recognition, novelty detection, character recognition, regression analysis, speech recognition, image compression, stock market prediction, electronic nose, security, loan applications, data processing, robotics, and control.
6.	Scalability of the solution	It is flexible and suitable for text and document format. It has high speed, robustness ,etc.

PROBLEM-SOLUTION FIT

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Customer who need to identify the digit from handwritten form

6. CUSTOMER CONSTRAINTS CC

It requires much more computation than more standard OCR techniques

5. AVAILABLE SOLUTIONS AS

Handwritten digit recognition using MNIST dataset is a major project made with the help of Neural Network.It basically detects the scanned images of handwritten digits

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS J&P

Diffferent people handwriting varies from each other and they struggle to identify

9. PROBLEM ROOT CAUSE RC

From the number 0 to 9 it's shapes and design are vary.Further according to individual person their handwriting also varies.Thus this handwritten digit recognition is needed

7. BEHAVIOUR BE

The output of an OCR run for an clear image and comparing it to the original version of the same text gives good accuracy

Focus on J&P, tap into BE, understand RC

3. TRIGGERS

TR

While they recognition the handwritten digit

4. EMOTIONS: BEFORE / AFTER

EM

Dilemma,exhausted into satified ,hopeful and comfort

10. YOUR SOLUTION

SL

Neural Network is used to recognise and predict the handwritten digits.Dataset are trained using gradient descent back propagation algorithm and tested using the feed forward algorithm.

Observing the system performance with variation of number of hidden units and iteration.Using this method, digits recognised and its accuracy will be high upto 99% .So we get good output

8.CHANNELS OF BEHAVIOUR

CH

8.1.ONLINE

Here extract from block

8.2.OFFLINE

Here extract from different user for handwriting

Extract online & offline CH of BE

CHAPTER-4

REQUIREMENT ANALYSIS

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Implementation	To import all the modules need for training our model. Import the libraries and load the MINST dataset.
FR-2	User Registration	Registration through Gmail
FR-3	User Confirmation	Confirmation via Email
FR-4	Preprocessing	Model cannot take the image data directly so we need to perform some basic operations and process the data. The CNN model will require one more dimension so we reshape the matrix to shape (60000,28,28,1)
FR-5	Create and Train the model	Creating CNN model in Python data science project. A CNN model generally consists of convolutional and pooling layers. Keras will start the training of the model.
FR-6	Evaluation	We have 10,000 images in our dataset. The MNIST dataset is well balanced so we can get around 99% accuracy.

Non-functional Requirements:

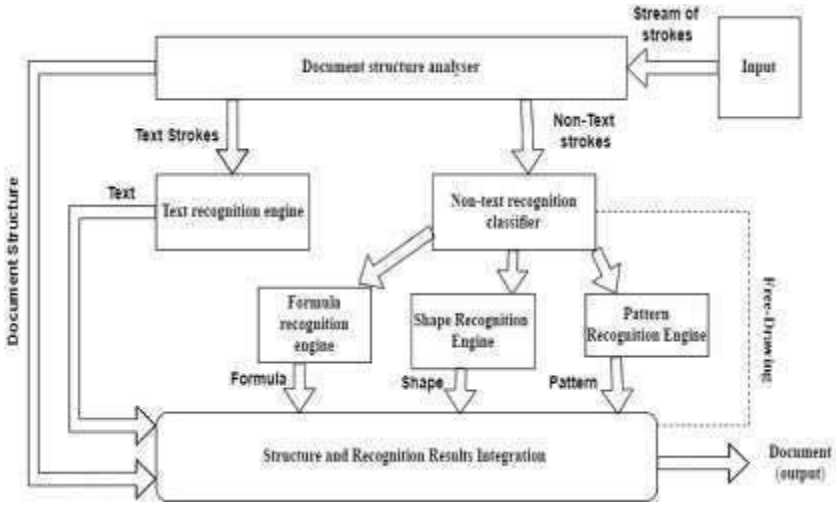
Following are the non-functional requirements of the proposed solution.

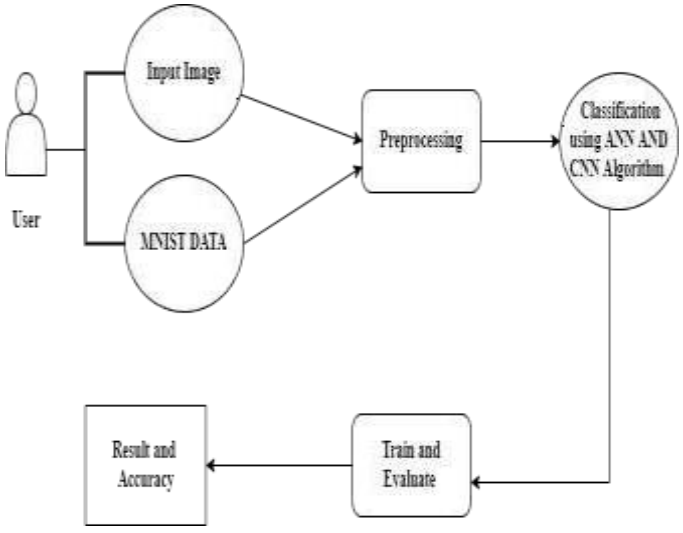
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Handwritten character recognition is one of the practically important issues in pattern recognition applications. The applications of digit recognition include in postal mail sorting, bank check processing, form data entry, etc.
NFR-2	**Security**	Most PC efforts to establish safety include information encryption and passwords, OCR plays an important role for digital libraries, allowing the entry of image textual information into computers by digitization, image restoration, and recognition methods.
NFR-3	**Reliability**	The overall highest accuracy 90.37% is achieved in the recognition process by Multilayer Perceptron.
NFR-4	**Performance**	Most standard implementations of neural networks achieve an accuracy of ~(98–99) percent in correctly classifying the handwritten digits.
NFR-5	**Availability**	The established CNN model can determine and recognize handwritten digits with high accuracy, as it combines the weights of convolution layers during feature extraction with fully connected layers.
NFR-6	**Scalability**	High speed, robustness, flexible and suitable for text and document formats.

CHAPTER-5

PROJECT DESIGN

DATA FLOW DIAGRAM





SOLUTION AND TECHNICAL ARCHITECTURE

TECHNICAL ARCHITECTURE

The deliverables shall include the architectural diagram as below and the information as per the table1 and table2

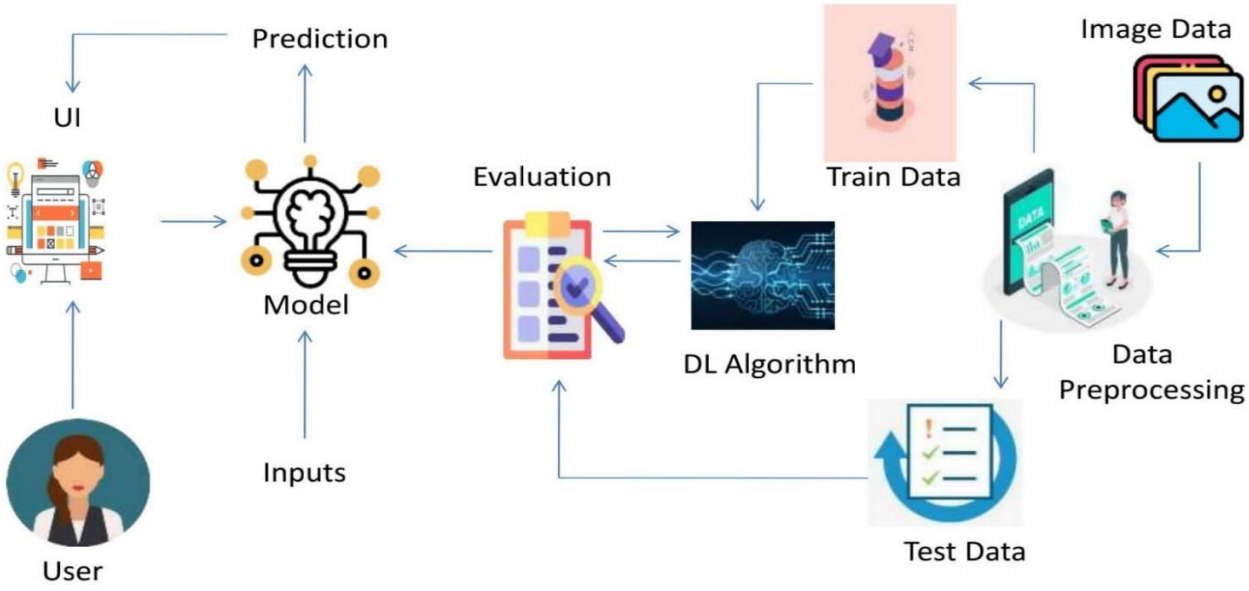


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	user interfaces with application for the recognition of the handwritten digits	HTML, CSS, JavaScript
2.	Application Logic-1	Logic for a process in the application	Java / Python
3.	Application Logic-2	Logic for a process in the application	IBM Watson STT service
4.	Application Logic-3	Logic for a process in the application	IBM Watson Assistant
5.	Database	digits dataset will be stored	MINIST dataset
6.	Cloud Database	Database Service on Cloud	IBM watson cloud.
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.	External API-2	Purpose of External API used in the application	Aadhar API
10.	Machine Learning Model	image processing ,data visualisation and evaluation	CNN, ANN ,RNN.
11.	Infrastructure (Server / Cloud)	cloud base web application:	cloud application

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Notification and redistribution	linux,python
2.	Security Implementations	request for authentication using encryption	encryption algorithm
3.	Scalable Architecture	the behaviour of the application must be correct and predictable	HTML,IBM CLOUD
4.	Availability	The web dashboard must be available to users	IBM CLOUD HOSTING
5.	Performance	To support 10,000 visits at the sametime while maintaining optimal performance	IBM load balancing

USER STORIES

Use the below template to list all the user stories for the product.

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 andawareness to use this application.	I can view the awarenessto use this application and its limitations.	Low	Sprint-1
		USN-2	As a user, I'm allowed to view the guidedvideo to use the interface of this application.	I can gain knowledge to use this application by apractical method.	Low	Sprint-1

		USN-3	As a user, I can read the instructions to usethis application.	I can read instructionsalso to use it in a user-friendly method.	Low	Sprint-2
	Recognize	USN-4	As a user, In this prediction page I get tochoose 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 andchoose the image to be uploaded	I can upload and choosethe image from the system storage and alsoin any virtual storage.	Medium	Sprint-3
		USN-7	As a user, I will train and test the input toget the maximum accuracy of output.	I can able to train and test the application untilit 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 MNISTdata set to produce the accurate result.	Medium	Sprint-3
Customer (Web user)	Home	USN-9	As a user, I can view the guide to use theweb app.	I can view the awareness of this application and its limitations.	Low	Sprint-1
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 andawareness to use this application.	I can view the awarenessto use this application and its limitations.	Low	Sprint-1

		USN-2	As a user, I'm allowed to view the guidedvideo to use the interface of this application.	I can gain knowledge to use this application by apractical method.	Low	Sprint-1
		USN-3	As a user, I can read the instructions to usethis application.	I can read instructions also to use it in a user-friendly method.	Low	Sprint-2
	Recognize	USN-10	As a user, I can use the web applicationvirtually anywhere.	I can use the application portably anywhere.	High	Sprint-1
		USN-11	As it is an open source, can use it costfreely.	I can use it without any payment to be paid for itto access.	Medium	Sprint-2
		USN-12	As it is a web application, it is installationfree	I can use it without the installation of the application or any software.	Medium	Sprint-4
	Predict	USN-13	As a user, I'm Allowed to upload andchoose the image to be uploaded	I can upload and choosethe image from the system storage and alsoin any virtual storage.	Medium	Sprint-3

CHAPTER-6

PROJECT PLANNING AND SCHEDULING

SPRINT PLANNING AND ESTIMATION

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data Collection	USN-1	I can collect the dataset from various resourceswith different handwritings.	10	Low	Pavithra, Dhanusha, Haritha Tharini, Hemapriya, Ishwarya, Keerthana
Sprint-1	Data Processing	USN-2	I can load the dataset and split the data into train and test.	10	Medium	Pavithra, Dhanusha, Haritha Tharini, Hemapriya, Ishwarya, Keerthana
Sprint-2	Add CNN layers	USN-3	Creating the model and adding the input, hidden, and output layer to it.	5	High	Pavithra, Dhanusha, Haritha Tharini, Hemapriya, Ishwarya, Keerthana
Sprint-2	Compiling the model	USN-4	With both the training data defined and modeldefined, it's time to configure the learning process	2	Medium	Pavithra, Dhanusha, Haritha Tharini, Hemapriya, Ishwarya, Keerthana
Sprint-2	Train & Test the model	USN-5	let us train and test our model with image dataset.	6	Medium	Pavithra, Dhanusha, Haritha Tharini, Hemapriya, Ishwarya, Keerthana
Sprint-2	Save the model	USN-6	Model is to be saved for future purposes. This saved model can also be integrated with an webapplication in order to predict something.	2	Medium	Pavithra, Dhanusha, Haritha Tharini, Hemapriya, Ishwarya, Keerthana

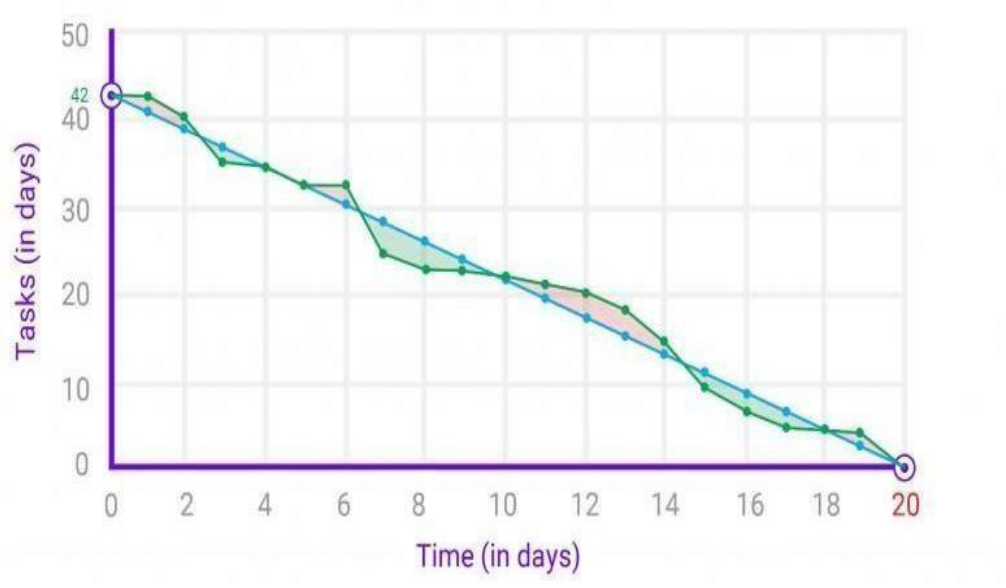
Sprint-3	UI Application Building	USN-7	Building a web application that is integrated intothe model we built. A UI is provided for the uses where he has uploaded an image. The uploaded image is given to the saved model and prediction is showcased on the UI.	5	High	Pavithra, Dhanusha, Haritha Tharini, Hemapriya, Ishwarya, Keerthana
Sprint-3		USN-8	We use HTML to create the front end part of the web page	5	High	Pavithra, Dhanusha, Haritha Tharini, Hemapriya, Ishwarya, Keerthana
Sprint-3		USN-9	Build the flask file which is a web frameworkwritten in python for server-side scripting.	5	High	Pavithra, Dhanusha, Haritha Tharini, Hemapriya, Ishwarya, Keerthana
Sprint-3		USN-10	Run the application.	5	High	Pavithra, Dhanusha, Haritha Tharini, Hemapriya, Ishwarya, Keerthana
Sprint-4	Train the model on IBM	USN-11	Build Deep Learning Model Using the IBM cloud.	5	High	Pavithra, Dhanusha, Haritha Tharini, Hemapriya, Ishwarya, Keerthana

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	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	05 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	05 Nov 2022

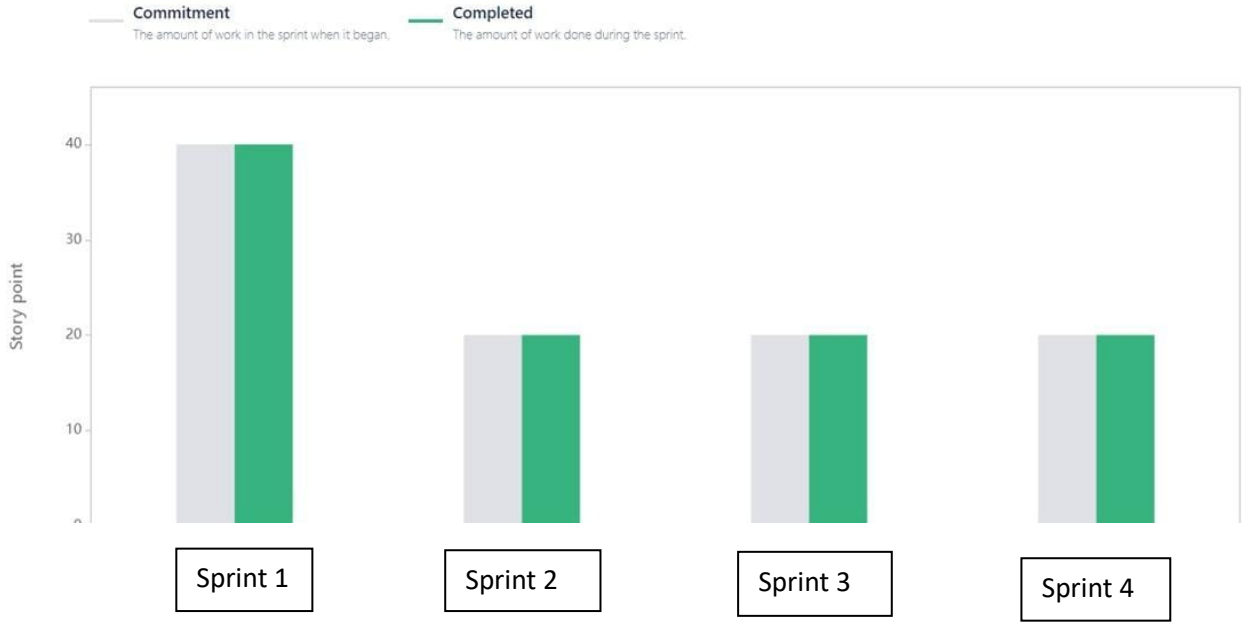
Burndown Chart:

A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies suchas Scrum. However, burn down charts can be applied to any project containing measurable progress over time.

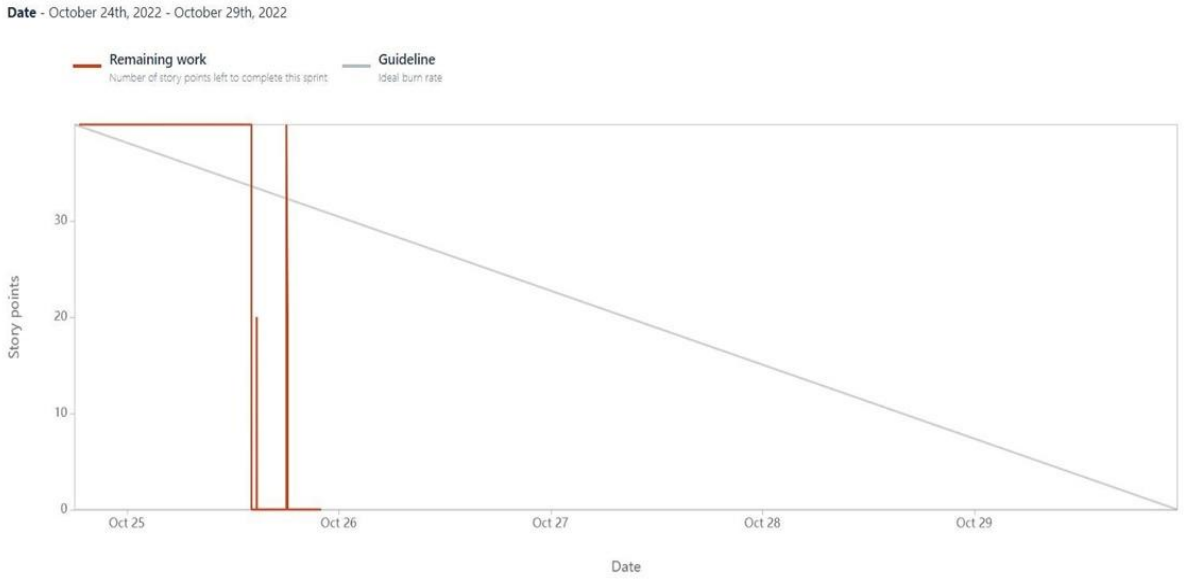


REPORTS FROM JIRA

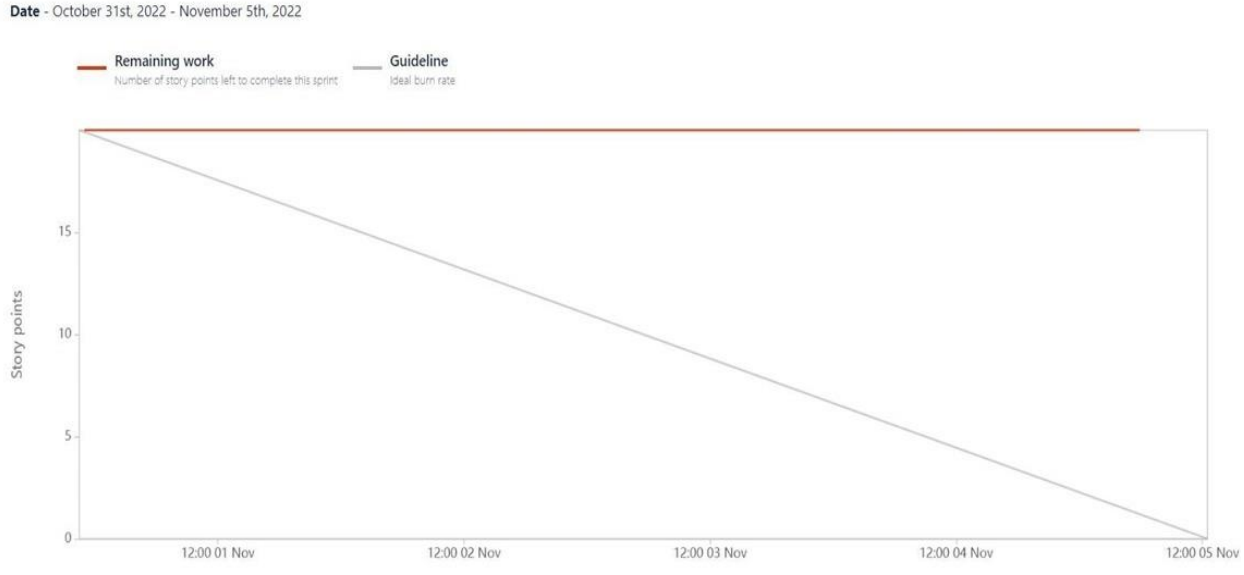
VELOCITY REPORT



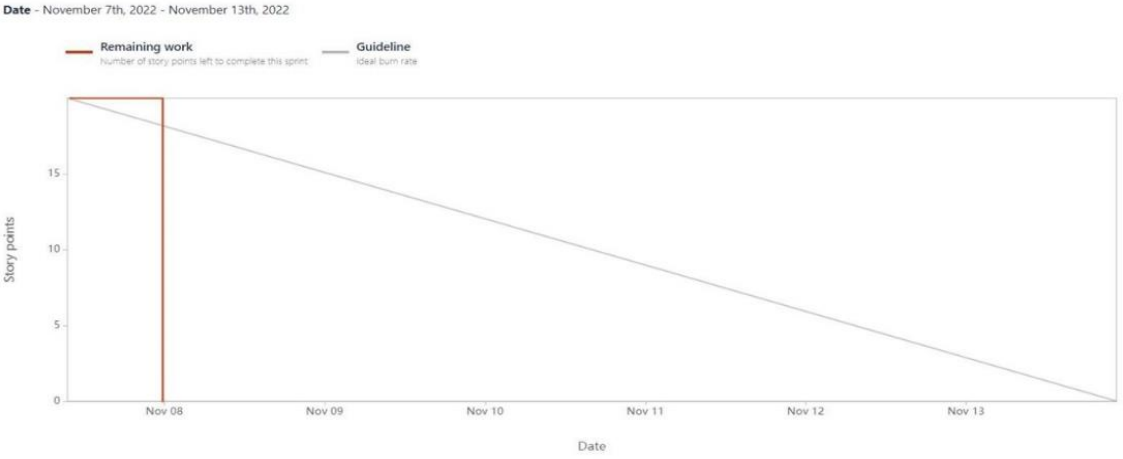
Sprint 1



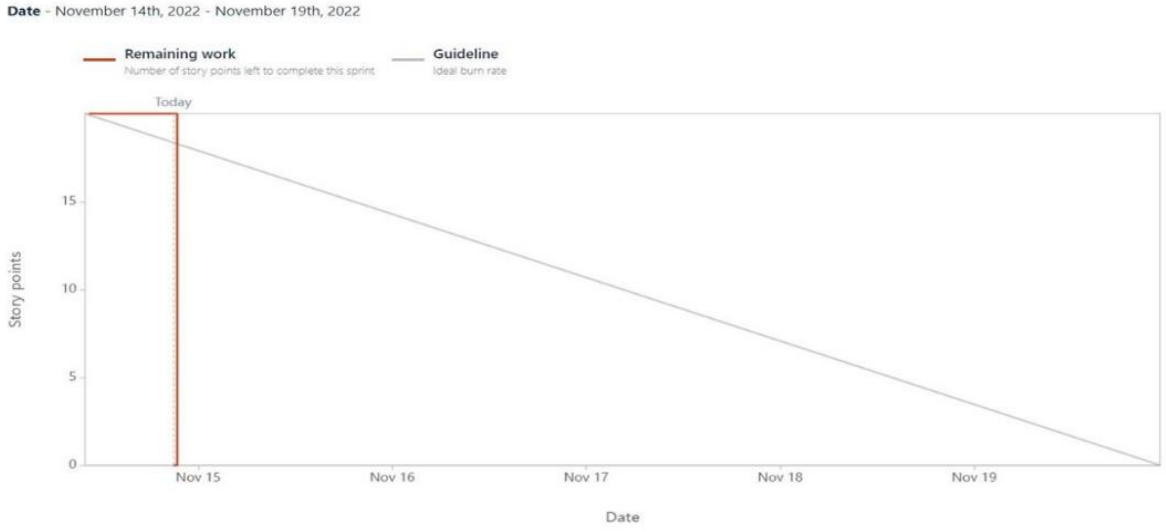
Sprint 2



Sprint 3



Sprint 4



CHAPTER-7

Coding & solutioning (Explain the features added in the project along with code)

```python
import os
from PIL import Image
from flask import Flask, request, render_template, url_for
from werkzeug.utils import secure_filename, redirect
from keras.models import load_model

UPLOAD_FOLDER = (r'C:\Users\HP\PycharmProjects\Digit recognition\uploads')


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

model = load_model("Digits.h5")


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


@app.route('/predict', methods=['GET', 'POST'])
def upload():
    if request.method == "POST":
        f = request.files["image"]
        filepath = secure_filename(f.filename)
        f.save(os.path.join(app.config['UPLOAD_FOLDER'], filepath))

        upload_img = os.path.join(UPLOAD_FOLDER, filepath)
        img = Image.open(upload_img).convert("L")  # convert image to monochrome
        img = img.resize((28, 28))  # resizing of input image
```

```
        im2arr = np.array(img)  # converting to image
        im2arr = im2arr.reshape(1, 28, 28, 1)  # reshaping according to our requirement

        pred = model.predict(im2arr)

        num = np.argmax(pred, axis=1)  # printing our Labels

        return render_template('predict.html', num=str(num[0]))


if __name__ == '__main__':
    app.run()

```

CHAPTER-8

TESTING

TEST CASES

Test case ID	Feature Type	Component	Test Scenario	Steps To Execute	Test Data	Expected Result	Actual Result	Status	TC for Automation(Y/N)	Executed By
Homepage_TC_OO 1	Functional	Home Page	Verify user is able to see the Homepage when clicked on the link	1.Enter URL and click go 2.Verify Homepage displayed or not	127.0.0.1:5000	Home Page should be displayed.	Working as expected	Pass	N	Pavithra Dhanusha Haritha Tharini Keerthana Ishwarya Hemapriya
Homepage_TC_OO 2	UI	Home Page	Verify the UI elements in Homepage	1.Enter URL and click go 2.Verify home screen with below UI elements: a.choose file button b.predict button c.clear button		Application should show below UI elements: a.choose file button b.predict button c.clear button	Working as expected	Pass	N	Pavithra Dhanusha Haritha Tharini Keerthana Ishwarya Hemapriya
Homepage_TC_OO 3	Functional	Home Page	Verify user is able to choose file from the local system and click on predict	1.Enter URL and click go 2.Click on Choose button 3.choose a file in valid format 4.Click on Predict	1.png	Choose file popup screen must be displayed and user should be able to click on predict button	Working as expected	Pass	N	Pavithra Dhanusha Haritha Tharini Keerthana Ishwarya Hemapriya
Homepage_TC_OO 4	Functional	Home page	Verify user able to select invalid file format	1.Enter URL and click go 2.Click on Choose button 3.choose a file in invalid format 4.Click on Predict	2.txt	Application won't allow to attach formats other than ".png, .jiff, .pjp, . jpeg, .jpg, .pjpeg"	Working as expected	Pass	N	Pavithra Dhanusha Haritha Tharini Keerthana Ishwarya Hemapriya
Predict_TC_ OO5	Functional	Predict page	Verify user iable to navigate to the predict to and view the predicted result	1.Enter URL and click go 2.Click on Choose button 3.choose a file in invalid format 4.Click on Predict	1.png	User must be navigated to the predict page and must view the predicted result	Working as expected	Pass	N	Pavithra Dhanusha Haritha Tharini Keerthana Ishwarya Hemapriya

USER ACCEPTANCE TESTING

1. PURPOSE OF DOCUMENT

The purpose of this document is to briefly explain the test coverage and open issues of the [ProductName] project at the time of the release to User Acceptance Testing (UAT).

2. 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	Total
By Design	1	0	1	0	2
Duplicate	0	0	0	0	0
External	0	0	2	0	2
Fixed	4	1	0	1	6
Not Reproduced	0	0	0	1	1
Skipped	0	0	0	1	1
Won't Fix	1	0	1	0	2
Total	6	1	4	3	14

3. Test Case Analysis

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

Section	Total Cases	Not Tested	Fail	Pass
Client Application	10	0	3	7
Security	2	0	1	1
Performance	3	0	1	2
Exception Reporting	2	0	0	2

CHAPTER-9

RESULTS

PERFORMANCE METRICS

MODEL SUMMARY

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 26, 26, 64)        640

 conv2d_1 (Conv2D)           (None, 24, 24, 32)        18464

 flatten (Flatten)           (None, 18432)             0

 dense (Dense)               (None, 10)                184330

=================================================================
Total params: 203,434
Trainable params: 203,434
Non-trainable params: 0
_________________________________________________________________
None
```

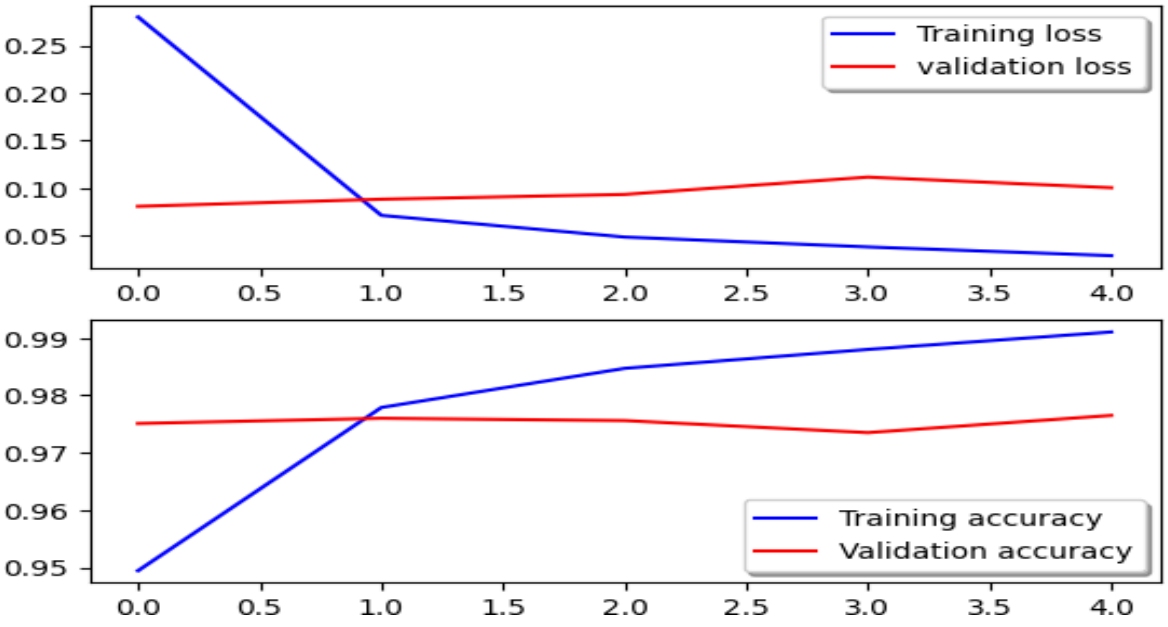
ACCURACY

CONTENT	VALUE
TRAINING ACCURACY	99.4
TRAINING LOSS	0.0442
VALIDATION ACCURACY	97.7
VALIDATION LOSS	0.0896

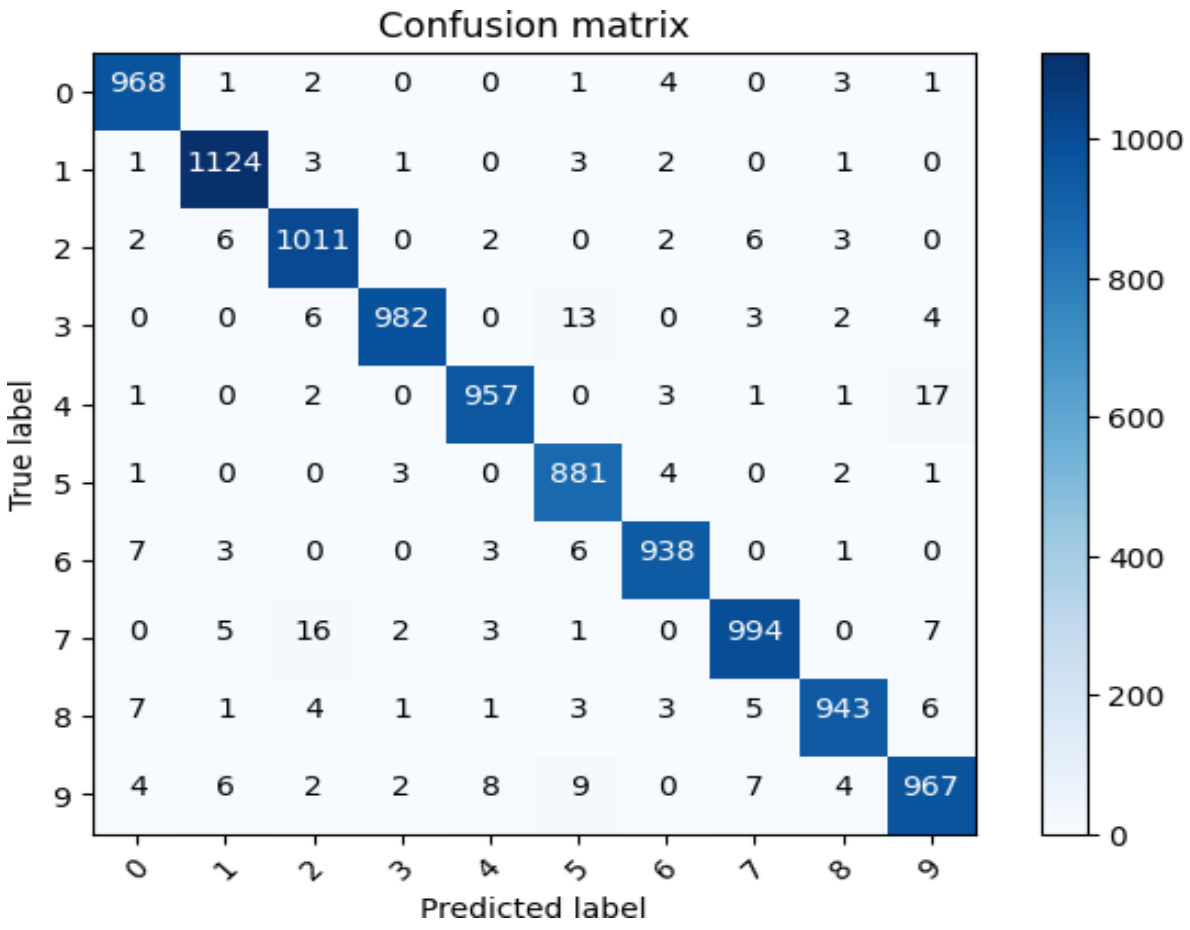
Train the model

```
[ ] model.fit(X_train,y_train,validation_data=(X_test,y_test),epochs=3,batch_size=32)

    Epoch 1/3
    1875/1875 [==============================] - 196s 104ms/step - loss: 0.2106 - accuracy: 0.9532 - val_loss: 0.1078 - val_accuracy: 0.9679
    Epoch 2/3
    1875/1875 [==============================] - 197s 105ms/step - loss: 0.0622 - accuracy: 0.9808 - val_loss: 0.0742 - val_accuracy: 0.9780
    Epoch 3/3
    1875/1875 [==============================] - 199s 106ms/step - loss: 0.0442 - accuracy: 0.9864 - val_loss: 0.0896 - val_accuracy: 0.9773
    <keras.callbacks.History at 0x7fc936da6f50>
```



CONFUSION MATRIX



CLASSIFICATION REPORT

	precision	recall	f1-score	support
0	0.98	0.99	0.98	980
1	0.98	0.99	0.99	1135
2	0.97	0.98	0.97	1032
3	0.99	0.97	0.98	1010
4	0.98	0.97	0.98	982
5	0.96	0.99	0.97	892
6	0.98	0.98	0.98	958
7	0.98	0.97	0.97	1028
8	0.98	0.97	0.98	974
9	0.96	0.96	0.96	1009
accuracy			0.98	10000
macro avg	0.98	0.98	0.98	10000
weighted avg	0.98	0.98	0.98	10000

PERFORMANCE METRICS RESULT:

Locust Test Report

During: 11/15/2022, 10:52:19 PM - 11/15/2022, 10:56:36 PM

Target Host: http://127.0.0.1:5000/

Script: locustfile.py

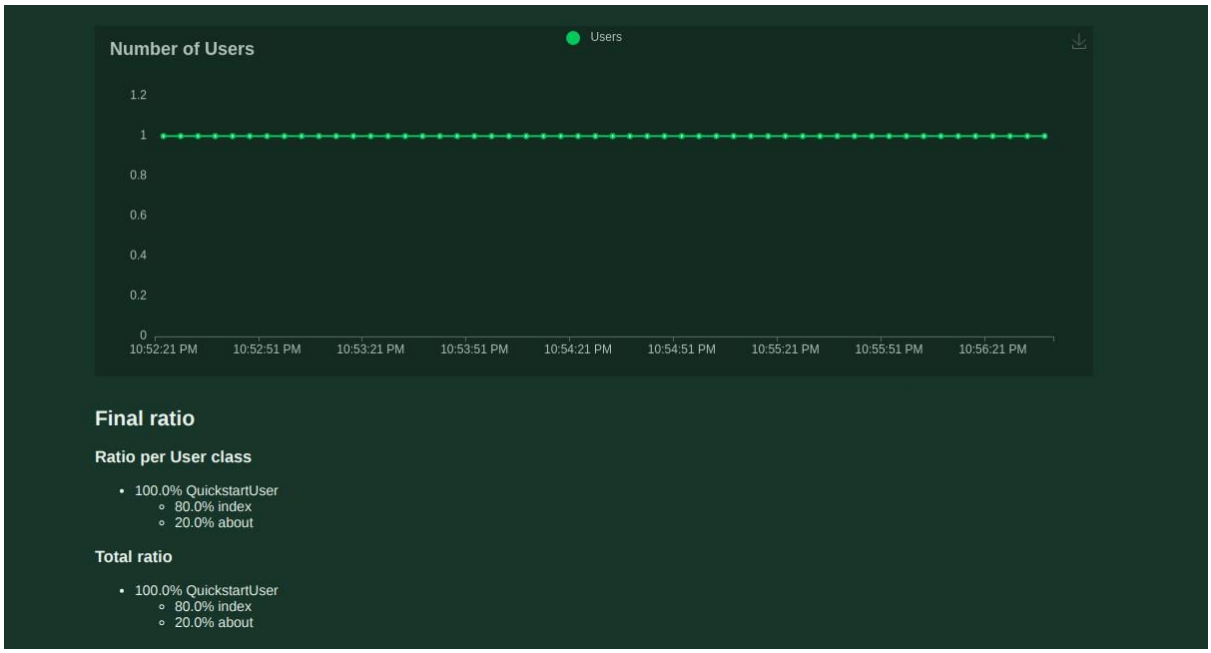
Request Statistics

Method	Name	# Requests	# Fails	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	RPS	Failures/s
GET	//	67	0	17	12	24	5875	0.3	0.0
GET	//predict	23	23	21	11	163	265	0.1	0.1
	Aggregated	**90**	**23**	**18**	**11**	**163**	**4441**	**0.4**	**0.1**

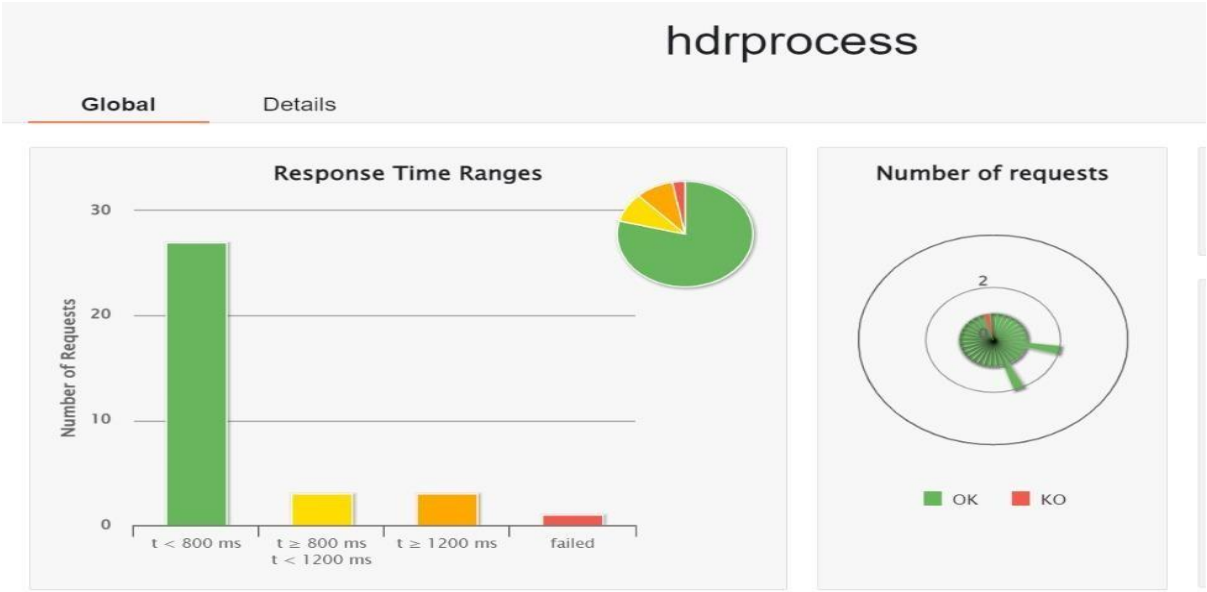
Response Time Statistics

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	//	18	18	19	19	22	23	25	25
GET	//predict	15	15	16	16	17	32	160	160
	Aggregated	**17**	**18**	**18**	**19**	**22**	**23**	**160**	**160**

Charts



GATLING REPORT



CHAPTER-10

ADVANTAGES AND DISADVANTAGES

ADVANTAGES

- Reduces manual work
- More accurate than average human
- Capable of handling a lot of data
- Can be used anywhere from any device

DISADVANTAGES

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

CHAPTER-11

CONCLUSION

- This project demonstrated a web application that uses Deep learning to recognise handwritten numbers. Flask, HTML, CSS, JavaScript, and a few other technologies were used to create this project.
- The model predicts the handwritten digit using a CNN network. The proposed project is scalable and can easily handle a huge number of users.
- Immediate applications of the digit recognition techniques include postal mail sorting automatically address reading and mail routing, bank check processing, etc.

CHAPTER-12

FUTURE SCOPE

This project is far from complete and there is a lot of room for improvement. Some of the improvements that can be made to this project are as follows:

- Add support to detect from digits multiple images and save the results
- Add support to detect multiple digits
- Improve model to detect digits from complex images
- Add support to different languages to help users from all over the world

This project has endless potential and can always be enhanced to become better. Implementing this concept in the real world will benefit several industries and reduce the workload on many workers, enhancing overall work efficiency.

CHAPTER-13

APPENDIX

SOURCE CODE

MODEL CREATION

```python
import os
from PIL import Image
from flask import Flask, request, render_template, url_for
from werkzeug.utils import secure_filename, redirect
from keras.models import load_model

UPLOAD_FOLDER = (r'C:\Users\HP\PycharmProjects\Digit recognition\uploads')


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

model = load_model("Digits.h5")


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


@app.route('/predict', methods=['GET', 'POST'])
def upload():
    if request.method == "POST":
        f = request.files["image"]
        filepath = secure_filename(f.filename)
        f.save(os.path.join(app.config['UPLOAD_FOLDER'], filepath))

        upload_img = os.path.join(UPLOAD_FOLDER, filepath)
        img = Image.open(upload_img).convert("L")  # convert image to monochrome
        img = img.resize((28, 28))  # resizing of input image
```

```python
        im2arr = np.array(img)  # converting to image
        im2arr = im2arr.reshape(1, 28, 28, 1)  # reshaping according to our requirement

        pred = model.predict(im2arr)

        num = np.argmax(pred, axis=1)  # printing our Labels

        return render_template('predict.html', num=str(num[0]))


if __name__ == '__main__':
    app.run()
```

FLASK APP

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

model = load_model("Digits.h5")


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


@app.route('/predict', methods=['GET', 'POST'])
def upload():
    if request.method == "POST":
        f = request.files["image"]
        filepath = secure_filename(f.filename)
        f.save(os.path.join(app.config['UPLOAD_FOLDER'], filepath))

        upload_img = os.path.join(UPLOAD_FOLDER, filepath)
        img = Image.open(upload_img).convert("L")  # convert image to monochrome
        img = img.resize((28, 28))  # resizing of input image
```

```
        im2arr = np.array(img)  # converting to image
        im2arr = im2arr.reshape(1, 28, 28, 1)  # reshaping according to our requirement

        pred = model.predict(im2arr)

        num = np.argmax(pred, axis=1)  # printing our Labels

        return render_template('predict.html', num=str(num[0]))


if __name__ == '__main__':
    app.run()
```

RECOGNIZER

Importing the required libraries

```
import numpy as np
import tensorflow
from tensorflow.keras.datasets import mnist
from tensorflow.keras.models import Sequential
from tensorflow.keras import layers
from tensorflow.keras.layers import Dense,Flatten
from tensorflow.keras.layers import Conv2D
from keras.optimizers import Adam
from keras.utils import np_utils
```

Loading the data

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

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz
11490434/11490434 [==============================] - 1s 0us/step
```

```
print(X_train.shape)
print(X_test.shape)
```

[5]

```
(60000, 28, 28)
(10000, 28, 28)
```

Analysing the data

```
X_train[0]
```

[6]

```
Output exceeds the size limit. Open the full output data in a text editor
array([[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
```

Model_Building.ipynb

D: > Project Development Phase > Sprint 2 > Model_Building.ipynb > #Sprint 2

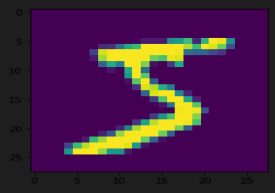
+ Code + Markdown | Run All | Clear Outputs of All Cells | Outline ...

```
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   3,
         18,  18,  18, 126, 136, 175,  26, 166, 255, 247, 127,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,  30,  36,  94, 154, 170,
        253, 253, 253, 253, 253, 225, 172, 253, 242, 195,  64,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,  49, 238, 253, 253, 253, 253,
        253, 253, 253, 253, 251,  93,  82,  82,  56,  39,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,  18, 219, 253, 253, 253, 253,
...
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0]], dtype=uint8)
```

```
y_train[0]
```

```
import matplotlib.pyplot as plt
plt.imshow(X_train[0])
```

```
<matplotlib.image.AxesImage at 0x7fd8393ca110>
```



Reshaping the data

```
X_train=X_train.reshape(60000,28,28,1).astype('float32')
X_test=X_test.reshape(10000,28,28,1).astype('float32')
```

Applying one hot encoding

```
number_of_classes=10
y_train=np_utils.to_categorical(y_train,number_of_classes)
y_test=np_utils.to_categorical(y_test,number_of_classes)
```

```
y_train[0]
```

```
array([0., 0., 0., 0., 0., 1., 0., 0., 0., 0.], dtype=float32)
```

Adding CNN layers

```
model=Sequential()
model.add(Conv2D(64,(3,3),input_shape=(28,28,1),activation='relu'))
model.add(Conv2D(32,(3,3),activation='relu'))
model.add(Flatten())
model.add(Dense(number_of_classes,activation='softmax'))
```

Compiling the model

```
model.compile(loss='categorical_crossentropy',optimizer='Adam',metrics=['accuracy'])
```

Train the model

```
model.fit(X_train,y_train,validation_data=(X_test,y_test),epochs=3,batch_size=32)
```

```
Epoch 1/3
1875/1875 [==============================] - 194s 103ms/step - loss: 0.2002 - accuracy: 0.9540 - val_loss: 0.0878 - val_accuracy:
0.9737
Epoch 2/3
1875/1875 [==============================] - 203s 108ms/step - loss: 0.0679 - accuracy: 0.9793 - val_loss: 0.0729 - val_accuracy:
0.9776
Epoch 3/3
1875/1875 [==============================] - 200s 107ms/step - loss: 0.0471 - accuracy: 0.9854 - val_loss: 0.0819 - val_accuracy:
0.9754

<keras.callbacks.History at 0x7fd839347ad0>
```

Observing the metrics

Observing the metrics

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

```
Metrics(Test loss & Test Accuracy):
[0.08188175410032272, 0.9753999710083008]
```

Test the model

```
prediction=model.predict(X_test[:4])
print(prediction)
```

```
1/1 [==============================] - 0s 134ms/step
[[1.1775780e-10 8.0942963e-14 9.2009266e-08 3.2085583e-07 1.5906704e-13
  8.5603686e-12 1.3867215e-14 9.9999940e-01 1.5925599e-08 1.5883167e-07]
 [7.9389675e-11 1.4074276e-09 1.0000000e+00 2.4798002e-12 1.7286957e-11
  1.0679558e-15 1.2779304e-10 7.8453709e-14 7.5344583e-12 1.1163937e-15]
 [5.0449985e-06 9.8447937e-01 5.1718434e-06 2.9296768e-07 1.5303572e-02
  8.4250714e-06 6.4097156e-07 1.7093236e-05 1.7952148e-04 8.8720617e-07]
 [9.9996698e-01 2.5253707e-10 2.0478637e-06 8.5173255e-08 1.1024300e-05
```

```
print(np.argmax(prediction,axis=1))
print(y_test[:4])
```

```
[7 2 1 0]
[[0. 0. 0. 0. 0. 0. 0. 1. 0. 0.]
 [0. 0. 1. 0. 0. 0. 0. 0. 0. 0.]
 [0. 1. 0. 0. 0. 0. 0. 0. 0. 0.]
 [1. 0. 0. 0. 0. 0. 0. 0. 0. 0.]]
```

Save the model

```
model.save('models/Digits.h5')
```

Home page (HTML)

```html
<html>

<head>
  <title>Digit Recognition WebApp</title>

  <meta name="viewport" content="width=device-width">
  <!-- GoogleFont -->
  <link href="https://fonts.googleapis.com/css2?family=Prompt:wght@600&display=swap" rel="stylesheet">
  <link href="https://fonts.googleapis.com/css2?family=Varela+Round&display=swap" rel="stylesheet">
  <link href="https://fonts.googleapis.com/css2?family=Source+Code+Pro:wght@500&display=swap" rel="stylesheet">
  <link href="https://fonts.googleapis.com/css?family=Calistoga|Josefin+Sans:400,700|Pacifico&display=swap" rel="styles
  <!-- bootstrap -->
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" integrity="sha
  <link rel="stylesheet" type= "text/css" href= "{{ url_for('static',filename='css/style.css') }}">
  <!-- fontawesome -->
  <script src="https://kit.fontawesome.com/b3aed9cb07.js" crossorigin="anonymous"></script>

  <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbz
  <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js" integrity="sha384-UO2eT0CpHqd
  <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js" integrity="sha384-JjSmVgyd0p3pXB
  <script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@latest"></script>
  <style>
    body{
     background-image: url('static/image/number.jpeg');
     background-repeat: no-repeat;
     background-size: cover;
    }
  </style>

</head>

<script>
```

```
  }

    $(document).ready(function() {
          $('#clear_button').on('click', function() {
              $('#image').val('');
              $('#frame').attr('src',"");
            });
        });

  </script>

    <body class="myBox">

      <h1 class="welcome">Handwritten Digit Recognition Website</h1>
      <div>
        <section id="content">
          <div class="center">
            <form action="/predict" method="POST" enctype="multipart/form-data">
            <label><b>SELECT IMAGE TO BE RECOGNIZED</b></label><br>
            <input id="image" type="file" name="image" accept="image/png, image/jpeg" onchange="preview()"><br><br>
            <img id="frame" src="" width="150px" height="150px"/>
              <div class="buttons_div">
                <button type="submit" class="btn btn-dark" id="predict_button">Predict</button>
                <button type="button" class="btn btn-dark" id="clear_button">  Clear  </button>
              </div>
            </form>
          </div>
        </section>
      </div>
    </body>

  </html>
```

Home page (CSS)

```
    margin-left: 15px;
    font-weight: bold;
    color: rgb(22, 24, 6);;
}

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

#content{
    margin: 0 auto;
    padding: 2% 15%;
    padding-bottom: 0;
}

.welcome{
    text-align: center;
    text-shadow: 0 0 3px #340808;
    position: relative;
    color: honeydew;
    background-image: linear-gradient(rgb(252, 195, 61), rgb(221, 220, 246));
    padding-top: 1%;
    padding-bottom: 1%;
    font-weight: bold;
    font-family: 'Prompt', sans-serif;
}

#team_id{
    text-align: right;
```

```
    text-align: right;
    font-size: 25px;
    padding-right: 3%;
}

#predict_button{
    margin-right: 15px;
    color: rgb(21, 23, 8);
    font-weight: bold;
}

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

#result{
    font-size: 5rem;
}

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

.btn {
    font-size: 15px;
    padding: 10px;
    -webkit-appearance: none;
```

```
    border: 1px solid rgb(24, 6, 6);
    margin-top: 20px;
    margin-bottom: 20px;
}

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

.heading{
    font-family: 'Varela Round', sans-serif;
    font-weight: 700;
    font-size: 2rem;
    display: inline;
}

.center{
    width: 600px;
    box-shadow: 0 4px 8px 0 hwb(0 15% 34%), 0 6px 20px 0 rgba(217, 9, 9, 0.19);
    text-align: center;
    text-align: center;
    background-color: rgb(255, 247, 247);
    margin: 0 auto;
    margin-top: 3%;
    /* padding-left: 10%; */
}

#frame{
    margin-right: 10%;
```

```
    width: 50%
}


.myBox {
    background: url(static/image/number.jpeg');
    /*background-image: linear-gradient(rgb(215, 242, 251), rgb(113, 144, 129));*/
    height: 700px; /* You must set a specified height */
    background-repeat: no-repeat; /* Do not repeat the image */
    background-size: cover;
}

label{
    font-size: 1.5em;
    color: rgb(230, 24, 44);

}


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

```

```
#frame{
    margin-right: 10%;
}

.predicted_answer{
    text-align: center;
    margin: 0 auto;
    padding: 3% 5%;
    padding-top: 0;
    /* padding-left: 10%; */
}

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



#main{
    width:100%;
    height:100vh;
}
.left-column{
    float:left;
    width: 50%
}
.right-column{
    float:right;
    width: 50%
```

PREDICT PAGE (HTML)

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

<style>
    body{
     background-image: url('static/image/numbers3.jpg');
     background-repeat: no-repeat;
     background-size: cover;
    }

    #circle{
        background: lightblue;
        border-radius: 50%;
        width: 200px;
        height: 200px;
        position:absolute;
        top:25%;
        left:25%;
        transform:translate(-50%,-50%);
    }


    #ans{
     text-align: center;
     font-size: 35px;
     margin: 0 auto;
     padding: 3% 5%;
     padding-top: 40%;
     color: rgb(57, 1, 7);
    }

</style>
<body>
    <div id="circle">
        <h1 id="ans">Result : {{num}}</h1>
    </div>
</body>
</html>
```

GITHUB & PROJECT DEMO LINK

GITHUB LINK

https://github.com/IBM-EPBL/IBM-Project-17989-1659677747

DEMO VIDEO

https://drive.google.com/drive/folders/1RQqo3jr2JYA1yX-W-bPkXsxt_LroHdkK?usp=sharing