A Novel Method for Handwritten Digit Recognition System

NALAIYA THIRAN PROJECT REPORT 2022

Submitted by

Bollavaram Vamsidharreddy	212219060048
Mugilan P	212219060171
Karthikeyan G	212219060129
Karthickkeyan R	212219060131

in partial fulfillment for the award of the degree
of
BACHELOR OF ENGINEERING

IN

ELECTRIONICS AND COMMUNICATION ENGINEERING

SAVEETHA ENGINEERING COLLEGE

ANNA UNIVERSITY: CHENNAI 600 025

TEAM ID : PNT2022TMID03640

Project report

ABSTRACT

Handwritten character recognition is one of the practically important issues in pattern recognition applications. The main purpose of this project is to build an automatic handwritten digit recognition method for the recognition of handwritten digit strings. To accomplish the recognition task, first, the digits will be segmented into individual digits. Then, a digit recognition module is employed to classify each segmented digit completing the handwritten digit string recognition task. The applications of digit recognition include postal mail sorting, bank check processing, form data entry, etc. The heart of the problem lies within the ability to develop an efficient algorithm that can recognize handwritten digits and which is submitted by users by the way of a scanner, tablet, and other digital devices.

1. INTRODUCTION

1.1 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). This has been a topic of boundless research in the field of deep learning. Digit recognition has many applications like number plate recognition, postal mail sorting, bank check processing, etc. In Handwritten digit recognition, we face many challenges because of different styles of writing of different peoples as it is not an Optical character recognition. This research provides a comprehensive comparison between different machine learning and deep learning algorithms for the purpose of handwritten digit recognition. For this, we have used Support Vector Machine, Multilayer Perceptron, and Convolutional Neural Network. The comparison between these algorithms is carried out on the basis of their accuracy, errors, and testing-training time corroborated by plots and charts that have been constructed using matplotlib for visualization.

The accuracy of any model is paramount as more accurate models make better decisions. The models with low accuracy are not suitable for real- world applications. Ex- For an automated bank cheque processing system where the system recognizes the amount and date on the check, high accuracy is very critical. If the system incorrectly recognizes a digit, it can lead to major damage which is not desirable. That's why an algorithm with high accuracy is required in these real-world applications. Hence, we are providing a comparison of different algorithms based on their accuracy so that the most accurate algorithm with the least chances of errors can be employed in various applications of handwritten digit recognition.

This paper provides a reasonable understanding of machine learning and deep learning algorithms like SVM, CNN, and MLP for handwritten digit recognition. It furthermore gives you the information about which algorithm is efficient in performing the task of digit recognition. In further sections of this paper, we will be discussing the related work that

has been done in this field followed by the methodology and implementation of all the three algorithms for the fairer understanding of them. Next, it presents the conclusion and result bolstered by the work we have done in this paper. Moreover, it will also give you some potential future enhancements that can be done in this field. The last section of this paper contains citations and references used.

1.2 PURPOSE

We describe a method of recognizing handwritten digits by fitting generative models that are built from deformable B-splines with Gaussian "ink generators" spaced along the length of the spline. The splines are adjusted using a novel elastic matching procedure based on the expectation maximization algorithm that maximizes the likelihood of the model generating the data. This approach has many advantages: 1) the system not only produces a classification of the digit but also a rich description of the instantiation parameters which can yield information such as the writing style; 2) the generative models can perform recognition driven segmentation; 3) the method involves a relatively small number of parameters and hence training is relatively easy and fast; and 4) unlike many other recognition schemes, it does not rely on some form of pre-normalization of input images, but can handle arbitrary scalings, translations and a limited degree of image rotation. We have demonstrated that our method of fitting models to images does not get trapped in poor local minima. The main disadvantage of the method is that it requires much more computation than more standard OCR techniques.

2. LITERATURE SURVEY

2.1 EXISTING PROBLEM

An early notable attempt in the area of character recognition research is by Grimsdale in 1959. The origin of a great deal of research work in the early sixties was based on an approach known as analysis by-synthesis method suggested by Eden in 1968. The great importance of Eden's work was that he formally proved that all handwritten characters are formed by a finite number of schematic features, a point that was implicitly included in previous works. This notion was later used in all methods in syntactic (structural) approaches of character recognition.

Paper 1: Handwritten Digit Recognition of MNIST dataset using Deep Learning state-of-the-art Artificial Neural Network (ANN) and Convolutional Neural Network (CNN) Drishti Beohar, Akhtar Rasool 2021 International Conference on Emerging Smart Computing and Informatics (ESCI) Handwritten digit recognition is an intricate assignment that is vital for developing applications, in computer vision digit recognition is one of the major applications. There has been a copious exploration done in the Handwritten Character Recognition utilizing different deep learning models. Deep learning is rapidly increasing in demand due to its resemblance to the human brain. The two major Deep learning algorithms Artificial Neural Network and Convolutional Neural Network which have been compared in this paper considering their feature extraction and classification stages of recognition. The models were trained using categorical cross-entropy loss and ADAM optimizer on the MNIST dataset. Backpropagation along with Gradient Descent is being used to train the networks along with reLU activations in the network which do automatic feature extraction. In neural networks, Convolution Neural Network (ConvNets or Convolutional neural networks) is one of the primary classifiers to do image recognition, image classification tasks in Computer Vision. Paper 2: Handwritten Digit Recognition Using Machine Learning Algorithms S M Shamim Global Journal of Computer Science and Technology, 18(1), 17–23, 2018-04-13 Handwritten character recognition is one of the practically important issues in pattern recognition applications. The applications of digit recognition includes in postal mail sorting, bank check processing, form data entry, etc. The heart of the problem lies within the ability to develop an efficient algorithm that can recognize hand written digits and which is submitted by users by the way of a scanner, tablet, and other digital devices. This paper presents an approach to off-line handwritten digit recognition based on different machine learning technique. The main objective of this paper is to ensure effective and reliable approaches for recognition of handwritten digits. Several machines learning algorithm namely, Multilayer Perceptron, Support Vector Machine, NaFDA5; Bayes, Bayes Net, Random Forest, J48 and Random Tree has been used for the recognition of digits using WEKA. The result of this paper shows that highest 90.37% accuracy has been obtained for Multilayer Perceptron Paper 3: Character Recognition using Artificial Neural Network Pranjali Pohankar, Namrata Taralkar, Snehalata Karmare, Smita Kulkarni

International Journal of Electronics Communication and Computer Engineering 5 (4), 2014 A neural network is a machine designed to model the way in which the brain performs a particular task. Character recognition techniques help in recognizing the characters written on paper documents and converting it in digital form. Handwritten character recognition is a very difficult problem due to great variation of writing style, different size and shape of the character. Neural network is a technique used to improve the accuracy and efficiency of the handwritten character recognition system. The error back propagation algorithm is used to train the MLP networks. The main advantage of the back propagation neural network (BPN) method is that it can fairly approximate a large class of functions. The aim of the paper is to use the improved neural network technique to recognize the offline handwritten characters. Paper 4: Handwritten Digit Recognition Based on Convolutional Neural Network Chao Zhang, Zhiyao Zhou, Lan Lin Department of Electronic Science and Technology Tongji University Shanghai In order to meet the needs of paperless offices and greatly improve work efficiency, it is necessary to research and implement a handwritten digit recognition system. Handwritten digit recognition plays an important role in large-scale data statistics and the financial business, such as industry annual inspection, population census, tax statements and checks, etc. This paper proposes a new type of handwritten digit recognition system based on convolutional neural networks (CNN). In order to improve the recognition performance, the network was trained with a large number of standardized pictures to automatically learn the spatial characteristics of handwritten digits. For model training, according to the loss function, the convolutional neural network continuously updates the network parameters with the data set in MNIST, which contains 60,000 examples. For model tests, the system uses the camera to capture the pictures composed of the images generated by the test data set of MNIST and the samples written by different people, then continuously processes the captured graphics and refreshes the output every 0.5 seconds. With the trained deep learning model, we got a recognition accuracy of 97.3% in the test process. Good performance in this experiment shows that our system can automatically recognize the handwritten digital content appearing in the target area and output the content label in real time. Paper 5: Offline handwritten digit recognition using neural network Sumedha B Hallale, Geeta D Salunke International Journal of

Advanced Research in Electrical, Electronics and Instrumentation Engineering 2 (9), 4373-4377, 2013 Optical character recognition is a typical field of application of automatic classification methods. In this paper, we have introduced a whole new idea of recognition of isolated handwritten digits which is known to be a difficult task and still lacks a satisfactory technical solution. The present paper proposes a novel approach for recognition of handwritten digits ie neural network classification. Back propagation neural network is one of the simplest methods for training multilayer neural networks. In this paper, we designed a back propagation neural network and trained it with a set of handwritten digits. The average success rates of recognition of all digits are 91.2%. Paper 6: Simplified Neural Network Design for Hand Written Digit Recognition Muhammad Zubair Asghar, Hussain Ahmad, Shakeel Ahmad, Sheikh Muhammad Saqib, Bashir Ahmad, Muhammad Junaid Asghar International Journal of Computer Science and Information Security 9 (6), 319, 2011 Neural Network is an abstraction of the central nervous system and works as a parallel processing system. Optimization, image processing, Diagnosis and many other applications are made very simple through neural networks, which are difficult and time consuming when conventional methods are used for their implementation. Neural Network is the simplified version of the human brain. Like the human brain, neural networks also exhibit efficient performance on perceptive tasks like recognition of visual images of objects and handwritten characters etc: Recognition of handwritten digits is one of the oldest applications of ANN. The recognition of digits written in different handwritings and also from scanned text has remained a trouble thus it has received much attention from researchers in the field of artificial neural networks. In this research work a very simple and flexible neural network scheme is proposed and implemented for handwritten digit recognition, which will assist beginners and AI students who want to understand the perceptive capability of neural networks. In the proposed system, a very simple design of artificial neural networks is implemented. Paper 7: Persian Handwritten Digit Recognition using Support Vector Machines Omid Rashnodi, Hedieh Sajedi, Mohammad Saniee Abadeh International Journal of Computer Applications (0975 –8887), Volume 29–No.12, September 2011 In this paper, appropriate features set based on Discrete Fourier Transform coefficients and the box approach have been proposed to achieve higher recognition

accuracy, decreasing the features set dimensions and recognition time of Persian numerals. In classification phase, support vector machine (SVM) has been employed as the classifier. Feature sets consists of 154 dimensions, which are the Fourier coefficients in the contour pixels of input image, average angle and distance pixels which are equal to one in each boxthe box approach.The scheme has been evaluated on 80,000 handwritten samples of Persian numerals. Using 60,000 samples for training, scheme was testedon other 20,000 samples and 98.84% correct recognition rate was obtained. Paper 8: A Survey on using Neural Networkbased Algorithms forHandWritten Digit Recognition Muhammad Ramzan,Shahid Mehmood Awan, Hikmat Ullah Khan,Waseem Akhtar, Ammara Zamir,Mahwish Ilyas,Ahsan Mahmood International Journal of Advanced Computer Science and Applications,Vol. 9, No. 9, 2018 The detection and recognition of handwrittencontent is the process of converting non-intelligent information such as images into machine edit-able text. This research domain has become an active research area due to vast applications in a number of fields such as hand writtenfiling of forms or documents in banks, exam form filled by students, users' authentication applications. Generally, the handwritten content recognition process consists of four steps: data preprocessing, segmentation, the feature extraction and selection, application of supervised learning algorithms. In this paper, a detailed survey of existing techniques used for Hand Written Digit Recognition (HWDR) is carried out. This review is novel as it is focused on HWDR and also it only discusses the application of Neural Network (NN) and its modified algorithms. We discuss an overview of NN and different algorithms which have been adopted from NN. In addition, this research study presents a detailed survey of the use of NN and its variants for digit recognition. Each existing work, we elaborate its steps, novelty, use of dataset and advantages and limitations as well. Moreover, we present a Sciento metric analysis of HWDR which presents top journals and sources of research contentin this research domain. We also present research challenges and potential future work. Paper 9: Handwritten Digits Recognition with Decision tree Classification: a Machine learning approach Tsehay Admassu Assegie, Pramod Sekharan Nair International Journal of Electrical and Computer Engineering (IJECE) ,Vol. 9, No. 5, October 2019. Handwritten digits recognition is an area of machine learning, in which a machine is trained

to identify handwritten digits. One method of achieving this is with decision tree classification model. A decision tree classification is a machine learning approach that uses the predefined labels from the past known sets to determine or predict the classes of the future data sets where the class labels are unknown. In this paper we have used the standard kaggle digits dataset for recognition of handwritten digits using a decision tree classification approach. And we have evaluated the accuracy of the model against each digit from 0 to 9. Paper 10: Handwritten Digit Recognition Using Various Machine Learning Algorithms and Models Pranit Patil International Journal of Innovative Research in Computer Science & Technology (IJIRCST) ISSN: 2347-5552, Volume- 8, Issue- 4, July-2020 Handwritten digit recognition is a technique or technology for automatically recognizing and detecting handwritten digital data through different Machine Learning models. In this paper we use various Machine Learning algorithms to enhance the productiveness of technique and reduce the complexity using various models. Machine Learning is an application of Artificial Intelligence that learns from previous experience and improves automatically through experience. We illustrate various Machine learning algorithms such as Support Vector Machine, Convolutional Neural Network, Quantum Computing, K-Nearest Neighbor Algorithm, Deep Learning used in Recognition technique

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. 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.

Reference:

https://miro.com/welcomeonboard/dlVxWjZWaUN5eHhEVXh5MkNpMldZV3FKUEZKSnhBNG11bXhZb3g4bGhmNWZMOElVY1dWMDh1d1Y5RkdxZU9MWnwzNDU4NzY0NTM1ODExOTI5MzQxfDI=?share_link_id=13642599616

Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	Perhaps the best attribute to analyse a customers behaviour, which will ultimately flourish your sales marketing is the determinant attribute. Now, the list of determinant attributes run the gamut.	Become an inspiration to others. Master a difficult skill. Become a thought leader in your industry. Get promoted to an executive role at your company. Learn about how to become a millionaire. Go on a trip around the world.	Not enough time. Lack of recognition. Poor communication skills. Lack of managerial involvement. Lack of growth opportunities.	when a new student takes your course without having access to the base knowledge beforehand, it can deter them from engaging in or completing the session entirely. How to overcome this learning barrier: Design your course with a learning path in mind, and use learning sequences	A customer feels **Surprise** immediately when he or she experiences customer service that is extraordinary or unexpected in a positive way. Even before help or she feels joy, there's a feeling of surprise at an unexpected level of service. Once surprise wears off, a customer is left with a feeling of Happiness

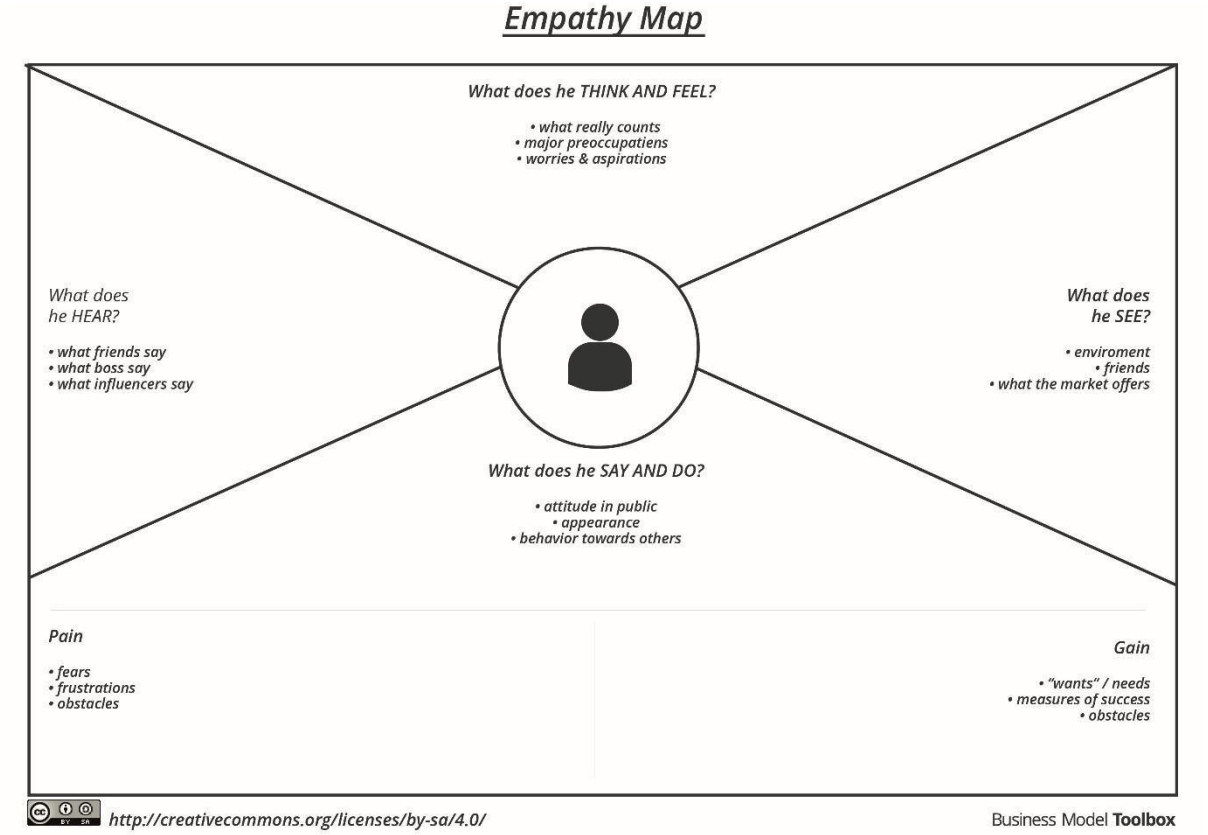
3.IDEATION & PROPOSED SOLUTION

3.1 **Empathy Map Canvas**:

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes.

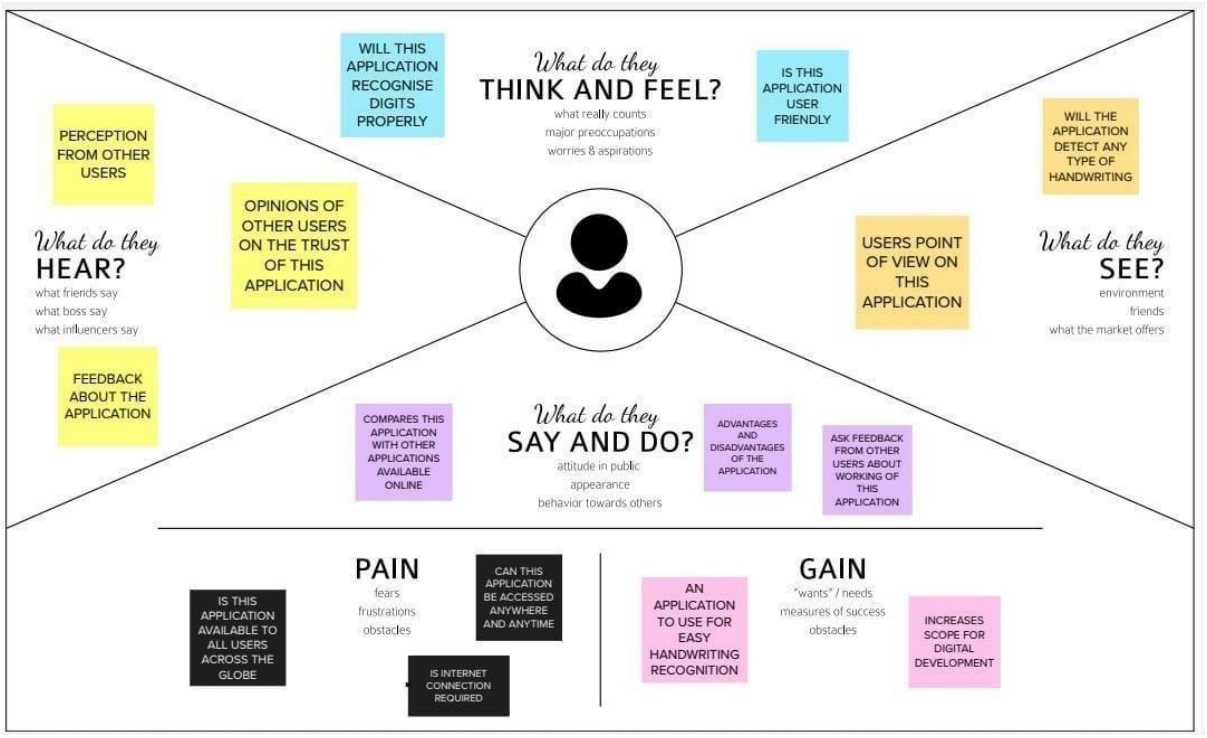
It is a useful tool to helps teams better understand their users.
Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.

Example:



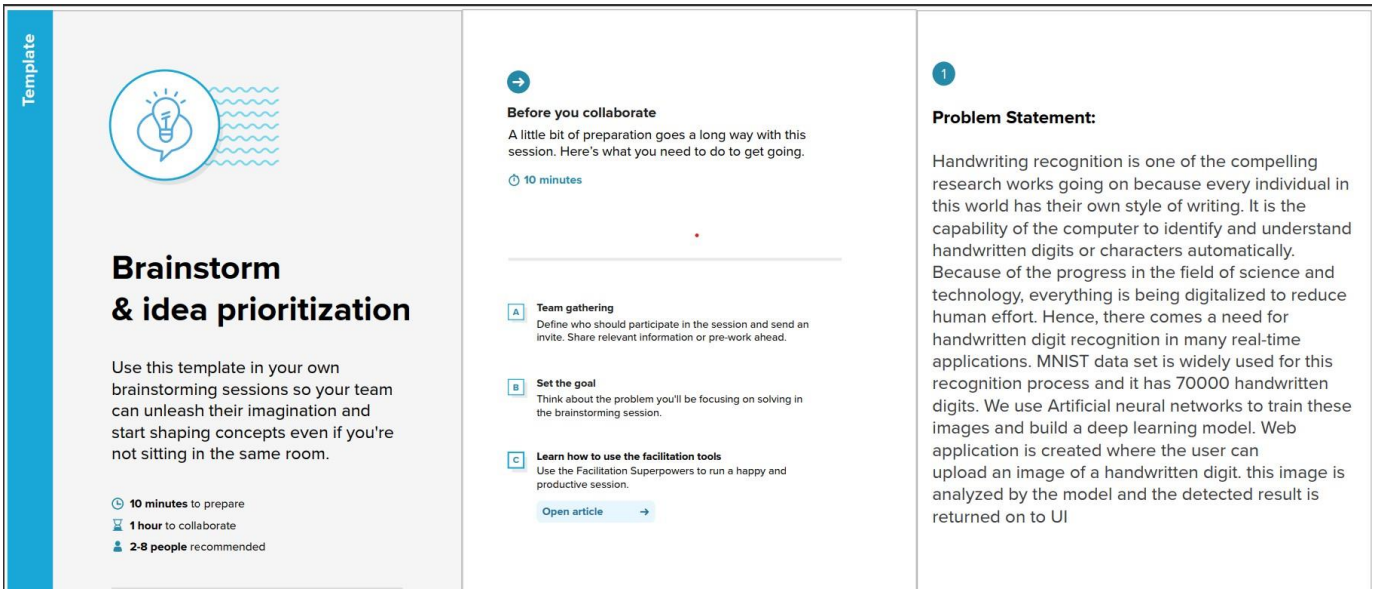
Reference: https://www.mural.co/templates/empathy-map-canvas

EMPATHY MAP FOR A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM



3.2 IDEATION & BRAINSTORMING

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



Step-2: Brainstorm, Idea Listing and Grouping

2

Brainstorm

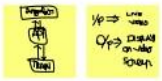
Write down any ideas that come to mind that address your problem statement.

10 minutes

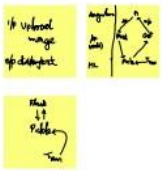
TIP

You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

Deepak M



Arun E



Rohith P M

Devakipriyadharshini V N

3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10 minutes, 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

Using Database for input and output

Using pickle

Based on no.of i/o pages

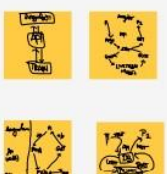
Single page

Separate input output page

Based on I/P type

Based on Interface

angular page



Flask

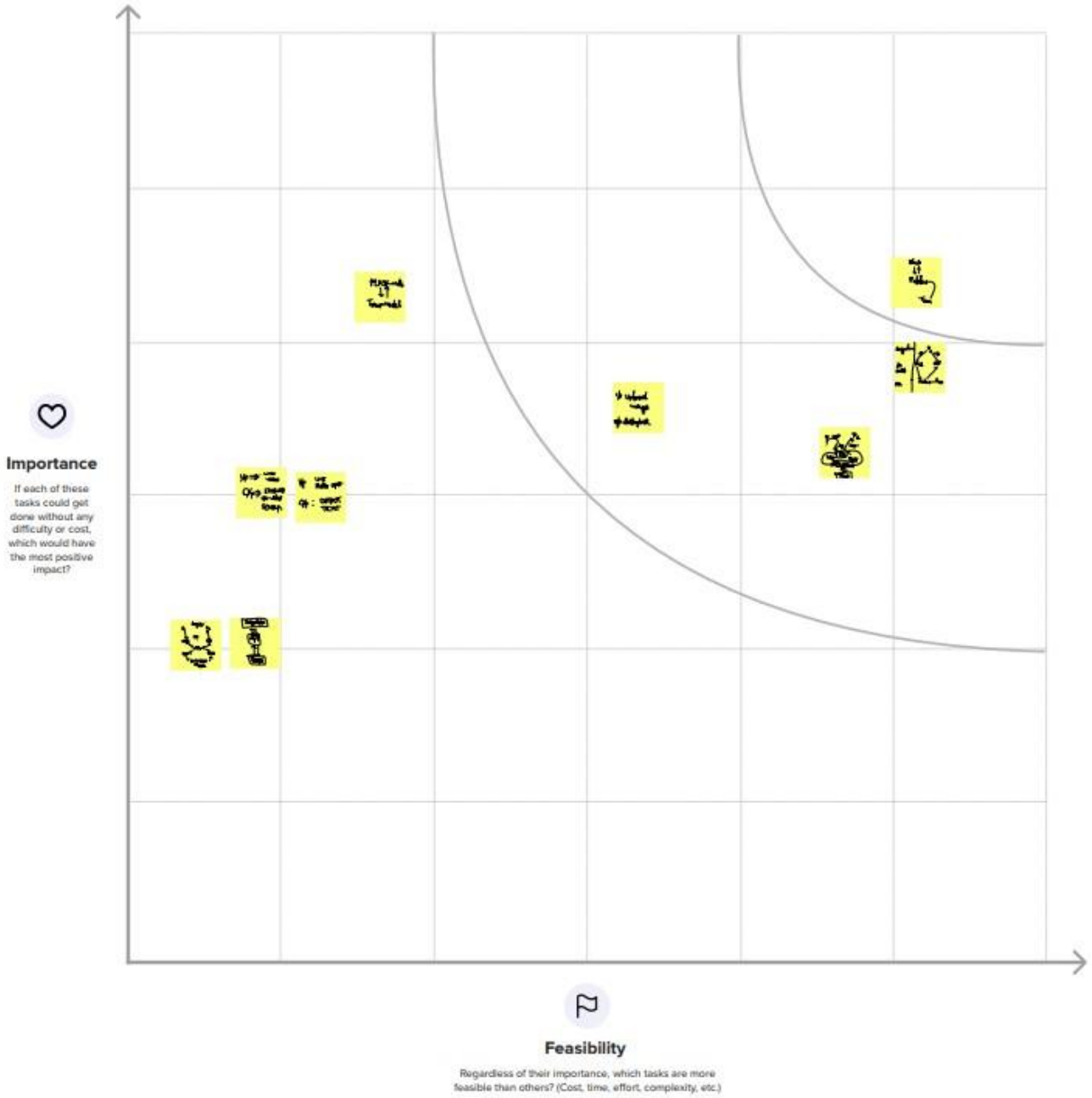
Step-3: Idea Prioritization

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



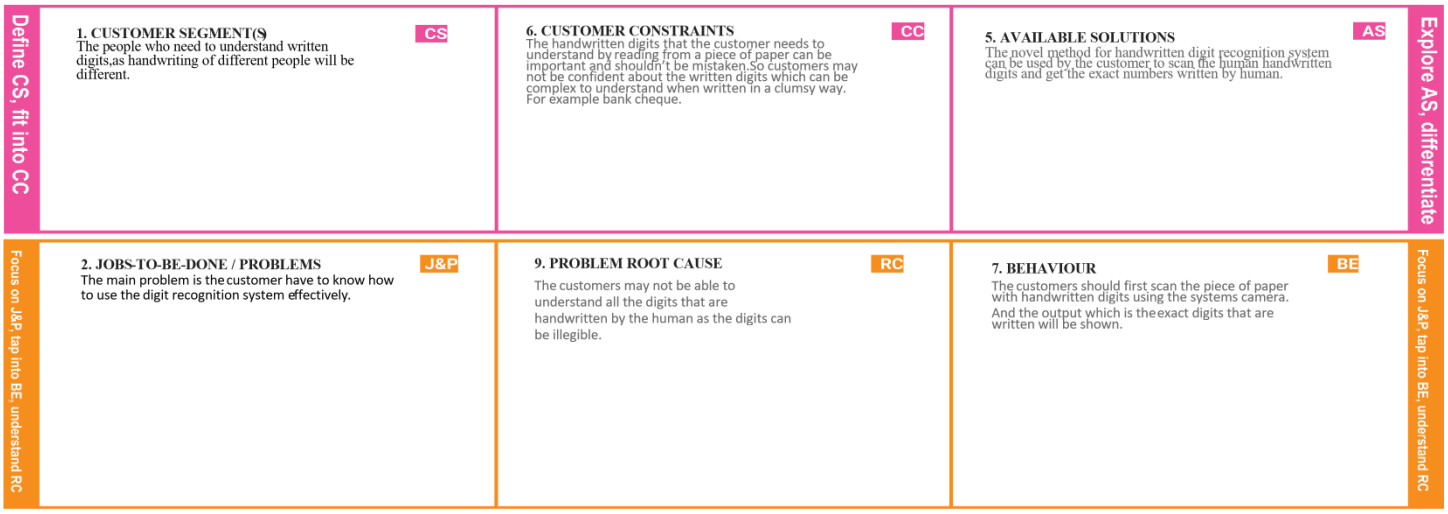
3.3 PROPOSED SOLUTION

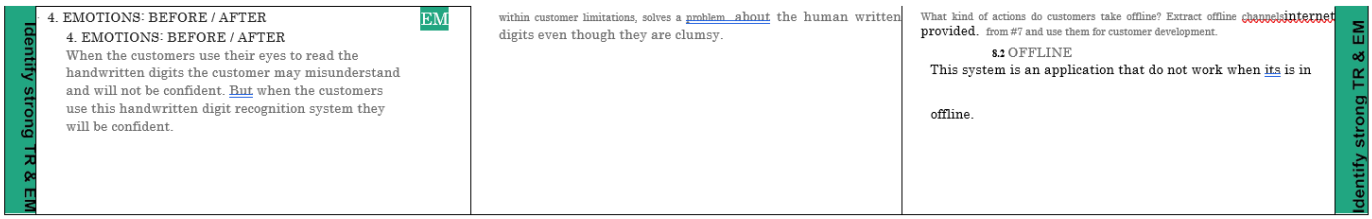
Proposed Solution Template:

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	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
2.	Idea / Solution description	HANDWRITTEN digit recognition is the ability of a computer system to recognize the handwritten inputs like digits, characters etc. from a wide variety of sources like emails, papers, images, letters etc. This has been a topic of research for decades. Some of the research areas include signature verification, bank check processing, postal address interpretation from envelopes etc
3.	Novelty / Uniqueness	One of the techniques that can be used to recognize handwritten Chinese characters is using Optical Character Recognition (OCR). Here, OCR uses probabilistic neural network to recognize Chinese characters[3]. The training of the classifier starts with using the distortion- modeled characters from four fonts.

4.	Social Impact / Customer Satisfaction	1) the system not only produces a classification of the digit but also a rich description of the instantiation parameters which can yield information such as the writing style; 2) the
		generative models can perform recognition driven segmentation; 3) the method involves a relatively small number of parameters and hence training is relatively easy and fast.
5.	Business Model (Revenue Model)	In business, System Analysis and Design refers to the process of examining a business situation with the intent of improving it through better procedures and methods. System analysis and design relates to shaping organizations, improving performance and achieving objectives for profitability and growth. The emphasis is on systems in action, the relationships among subsystems and their contribution to meeting a common goal.
6.	Scalability of the Solution	• The first layer of the architecture is the User layer. User layer will comprise of the people who interacts with the app and for the required results. • The next three layers is the frontend architecture of the application. The application will be developed using Bootstrap which is the open source platform for HTML, CSS and JavaScript. The application is deployed in the localhost which is shown on the browser. Through the app, the user will be able to upload pictures of the handwritten digits and convert it into the digitalized form.

3.4 PROBLEM SOLUTION FIT





4. REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT

4.1.1 System Configuration:

Software requirements**:**

These are the software configurations used:

Operating system: Windows 10.

IDE: Jupyter Notebook.

Python: Python is an interpreted, high-level, general purpose programming language created by Guido Van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant Whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is dynamically typed, and garbage collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming.

Jupyter Notebook: Jupyter is a free, open-source, interactive web tool known as a computational notebook, which researchers can use to combine software code, computational output, explanatory text and multimedia resources in a single document. Computational notebooks have been around for decades, but Jupyter in particular has exploded in popularity over the past couple of years. This rapid uptake has been aided by an enthusiastic community of user–developers and a redesigned architecture that allows the notebook.

4.2 NON-FUNCTIONAL REQUIREMENTS

4.2.1 Hardware requirements:

These are the Hardware interfaces used Processor: Intel Pentium 4 or equivalent

RAM: Minimum of 256 MB or higher HDD: 10 GB or higher

Monitor: 15’ or 17’ colour monitor

Mouse: Scroll or optical mouse

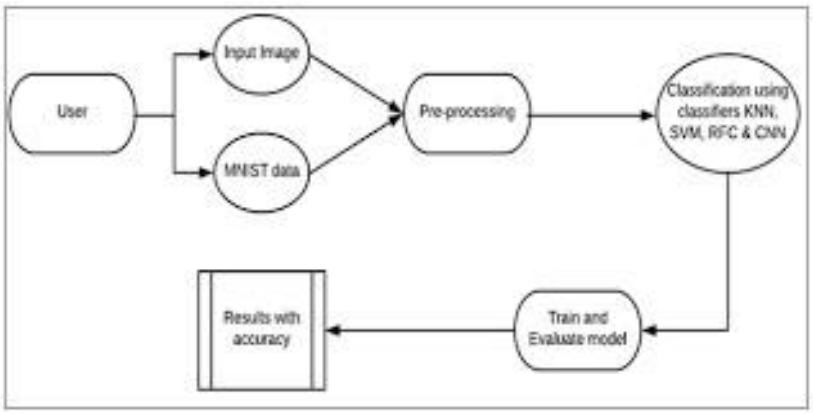
Keyboard: Standard 110 keys keyboard

5. PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS

Data Flow Diagrams:

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



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 (postal mail sorting, bank check processing, form data entry, etc)	Upload Image	USN-1	As a user, I will upload an image of the handwritten digits	I can scan the handwritten digits	High	Sprint-1
		USN-2	As a user, I will receive digital form of the handwritten digits	I can get the digital form of the handwritten digits	High	Sprint-1
Banking Sector	Scan form	USN-3	User will upload the form like deposit or cheque and automate further process		High	Sprint -1

5.2 SOLUTION & TECHNICAL ARCHITECTURE

Solution Architecture:

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.
- Define features, development phases, and solution requirements.
- Provide specifications according to which the solution is defined, managed, and delivered.

Example - Solution Architecture Diagram:

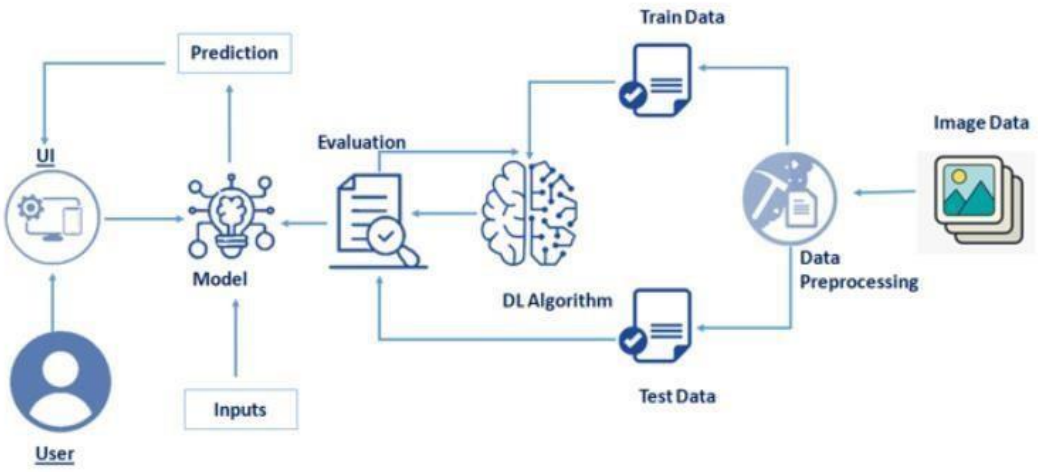
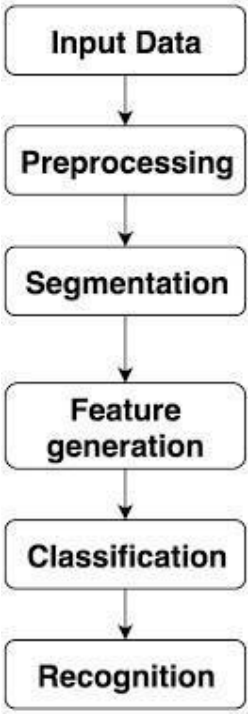


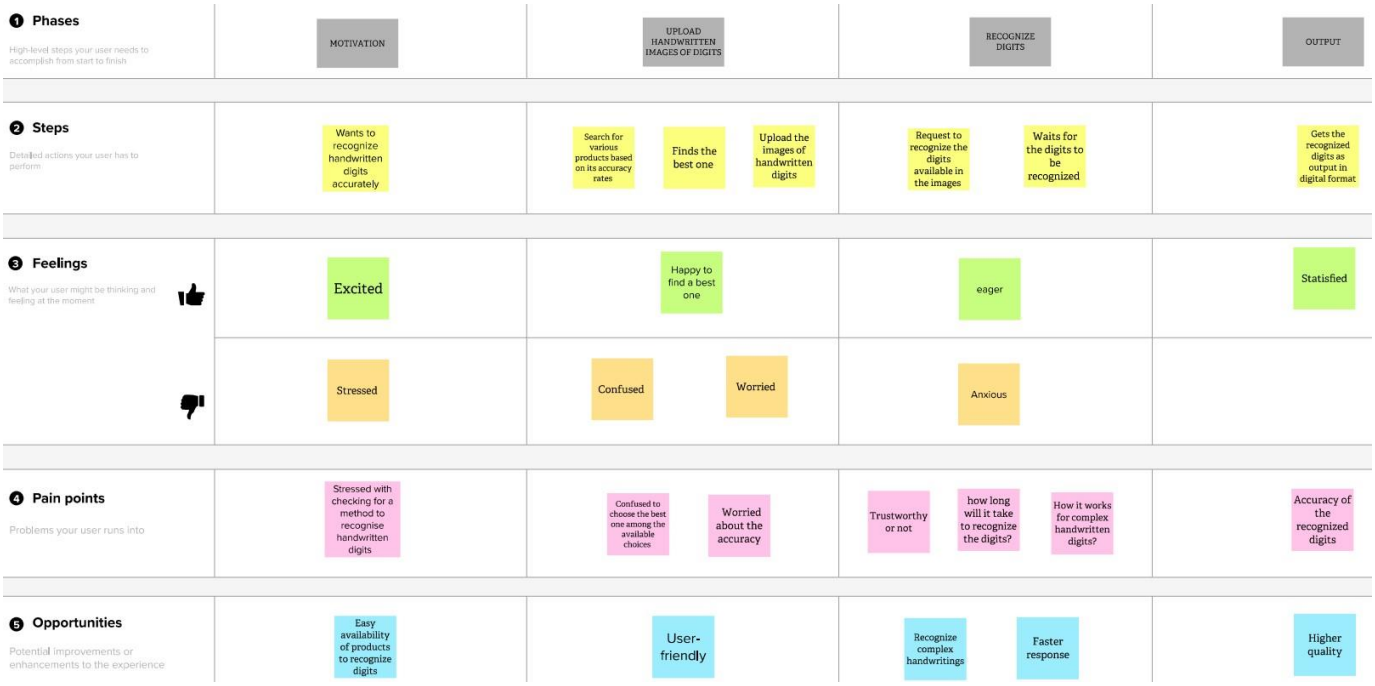
Figure 1: Architecture and data flow of the novel method for handwritten digit recognition system

Reference: https://careereducation.smartinternz.com/saas-guided-project/3/a-novelmethod-forhandwritten-digit-recognition-system



S.No	Component	Description	Technology
1.	User Interface	User interacts with application through Web UI	HTML, CSS, JavaScript
2.	Application Logic-1	Forward propagation: Input data is fed in the forward direction through the network. Each hidden layer accepts the input data, processes it as per the activation function and passes it to the successive layer. We will use the sigmoid function as our "activation function".	Python
3.	Application Logic-2	Backward propagation: It is the practice of finetuning the weights of a neural net based on the error rate obtained in the previous iteration.	Python
4.	Database	MNIST Dataset	Dataset

5.3 USER STORIES


1 Phases
High-level steps your user needs to accomplish from start to finish
MOTIVATION
UPLOAD HANDWRITTEN IMAGES OF DIGITS
RECOGNIZE DIGITS
OUTPUT
2 Steps
Detailed actions your user has to perform
Wants to recognize handwritten digits accurately
Search for various products based on its accuracy rates
Finds the best one
Upload the images of handwritten digits
Request to recognize the digits available in the images
Waits for the digits to be recognized
Gets the recognized digits as output in digital format
3 Feelings
What your user might be thinking and feeling at the moment
Excited
Happy to find a best one
eager
Statisfied
Stressed
Confused
Worried
Anxious
4 Pain points
Problems your user runs into
Stressed with checking for a method to recognise handwritten digits
Confused to choose the best one among the available choices
Worried about the accuracy
Trustworthy or not
how long will it take to recognize the digits?
How it works for complex handwritten digits?
Accuracy of the recognized digits
5 Opportunities
Potential improvements or enhancements to the experience
Easy availability of products to recognize digits
User-friendly
Recognize complex handwritings
Faster response
Higher quality

6. PROJECT PLANNING & SCHEDULING

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority
Sprint-1	Dashboard	USN-1	As a user, they can see the information regarding the prediction of handwritten digit recognition.	2	High
Sprint-1	Launch	USN-2	On clicking the launch button, it will redirect the user to a page where the images to be predicted can be uploaded.	2	High
Sprint-2	Upload	USN-3	Users can select the image from the local storage.	2	High
Sprint-3	Predict	USN-4	Once the image is uploaded, it will predict the	2	High

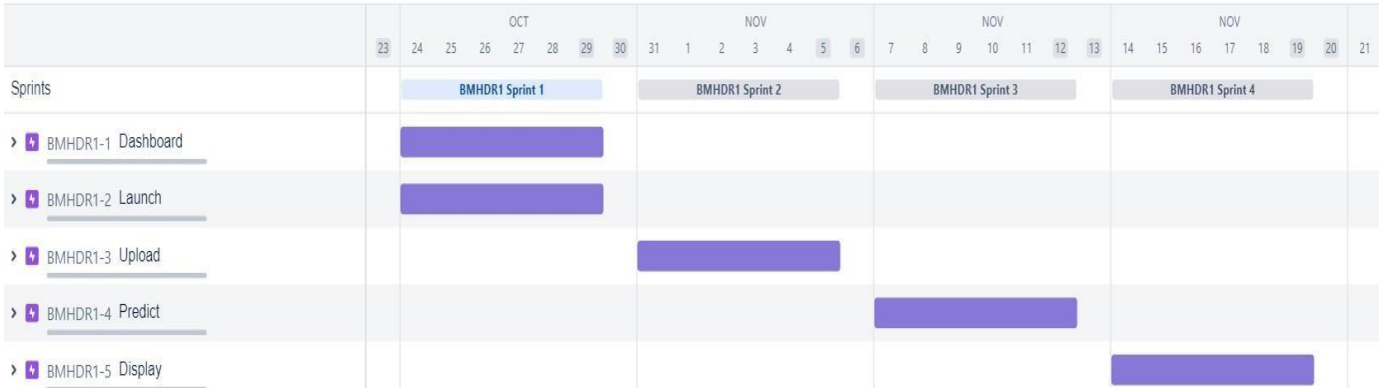
			respective image.		
Sprint-4	Display	USN-5	The predicted image will be displayed with the accuracy chart.	2	High

Project Tracker, Velocity & Burndown Chart

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

Velocity:

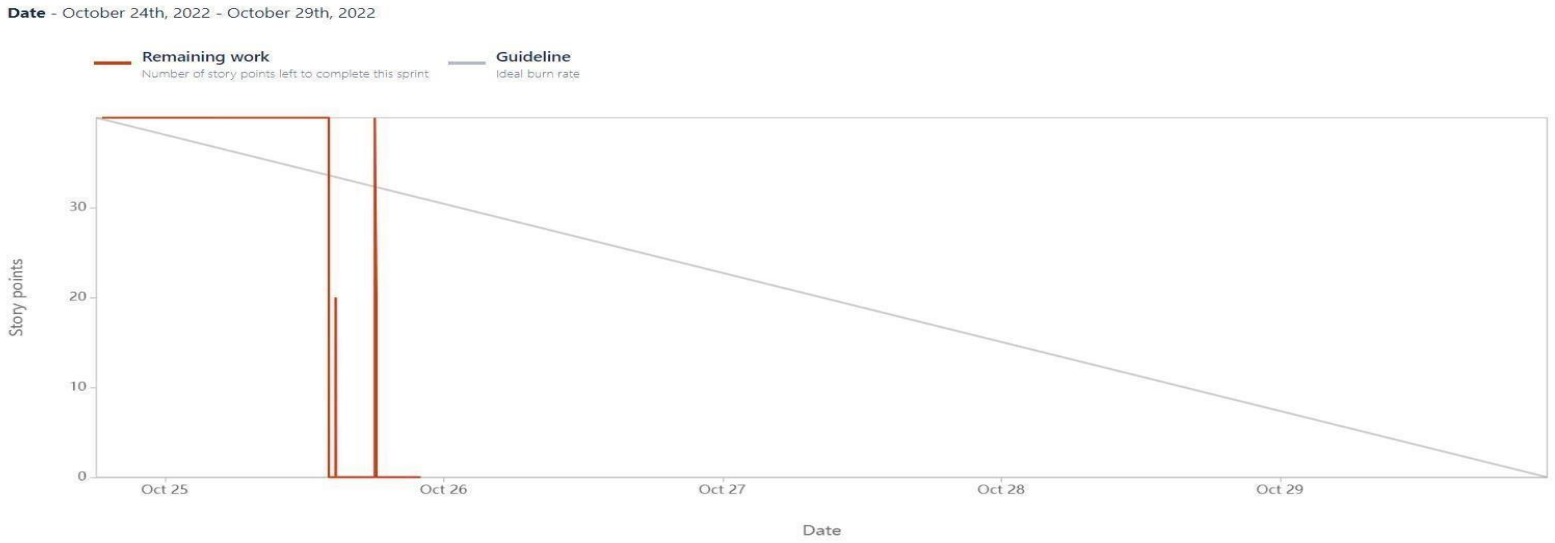
Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)



$$AV = \frac{\textit{Sprint Duration}}{\textit{Velocity}} = \frac{20}{6} = 3.33$$

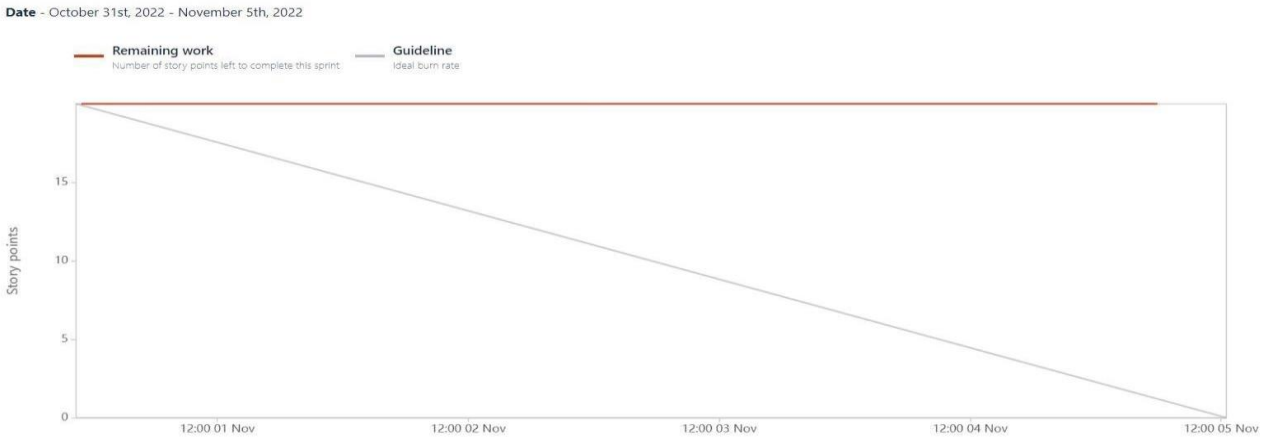
Burndown Chart:

A burndown 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.

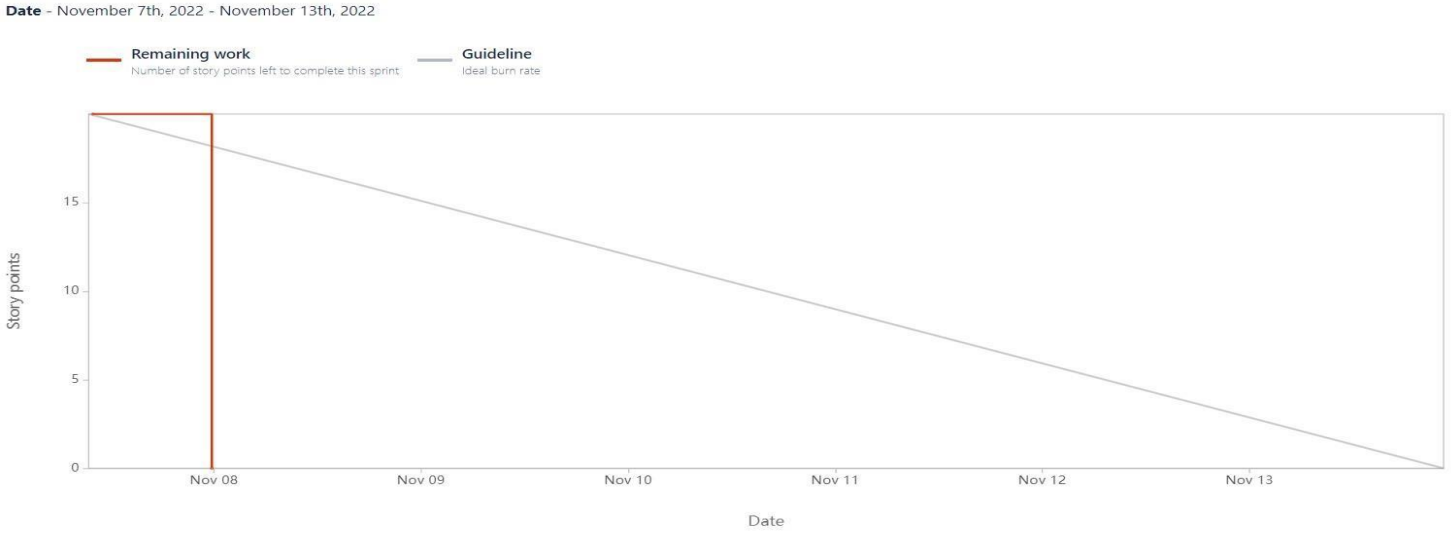


Sprint 1:

Sprint 2:


Date - October 31st, 2022 - November 5th, 2022
Remaining work
Number of story points left to complete this sprint
Guideline
Ideal burn rate
Story points
15
10
5
0
12:00 01 Nov
12:00 02 Nov
12:00 03 Nov
12:00 04 Nov
12:00 05 Nov

Sprint 3:


Date - November 7th, 2022 - November 13th, 2022
Remaining work
Number of story points left to complete this sprint
Guideline
Ideal burn rate
Story points
15
10
5
0
Nov 08
Nov 09
Nov 10
Nov 11
Nov 12
Nov 13
Date

7. CODING & SOLUTIONS

7.1 FEATURE 1

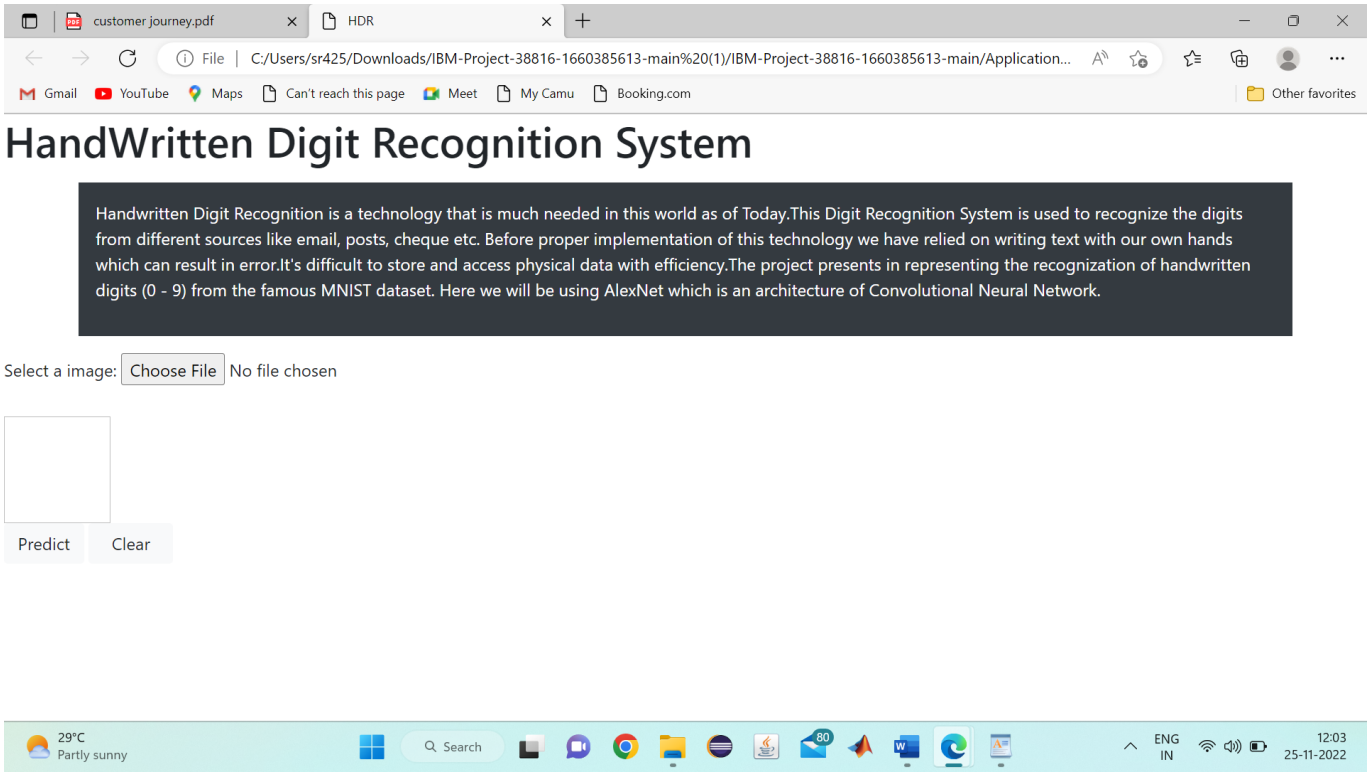
```
<html>

<head>
  <title>HDR</title>

  <meta name="viewport" content="width=device-width">

  <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&displ
ay=swap" rel="stylesheet">
  <link
href="https://fonts.googleapis.com/css?family=Calistoga|Josefin+Sans:400,700|
Pacifico&display=swap" rel="stylesheet">

  <link                                                    rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.cs
s"                                                      integrity="sha384-
ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">
  <link      rel="stylesheet"      type=      "text/css"      href=      "{{
url_for('static',filename='css/style.css') }}">

  <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+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
crossorigin="anonymous"></script>
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.j
s"                                                      integrity="sha384-
UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1"
crossorigin="anonymous"></script>
  <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"
integrity="sha384-
JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM"
crossorigin="anonymous"></script>
  <script
src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@latest"></script>
  <link                                                    rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.1/dist/css/bootstrap.min.css
">
```

```
  <script
src="https://cdn.jsdelivr.net/npm/jquery@3.6.0/dist/jquery.slim.min.js"></scr
ipt>
  <script
src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js"></
script>
  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.1/dist/js/bootstrap.bundle.mi
n.js"></script>

</head>
<style>
    body{
     background-image: url('static/images/bc1.jpg');
     background-repeat: no-repeat;
     background-size: cover;
    }
</style>

<script>
  function preview() {
    frame.src=URL.createObjectURL(event.target.files[0]);
}

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

</script>

<body>
    <h1>HandWritten Digit Recognition System</h1>
        <div class="container p-3 my-3 bg-dark text-white">
            <p>Handwritten Digit Recognition is a technology that is much
needed in this world as of Today.This Digit Recognition System is used to
recognize the digits from different sources like email, posts, cheque etc.
Before proper implementation of this technology we have relied on writing text
with our own hands which can result in error.It's difficult to store and access
physical data with efficiency.The project presents in representing the
recognization of handwritten digits (0 - 9) from the famous MNIST dataset. Here
we will be using AlexNet which is an architecture of Convolutional Neural
Network.</p>
        </div>
        <section id="content">

            <div class="leftside">
            <form  action="/predict"  method="POST"  enctype="multipart/form-
data">
            <label>Select a image:</label>
            <input  id="image"  type="file"  name="image"  accept="image/png,
image/jpeg" onchange="preview()"><br><br>
              <img id="frame" width="100px" height="100px"/>
              <div class="buttons_div">
```

```
                <button type="submit" class="btn btn-light">Predict</button>
                <button   type="button"   class="btn   btn-light">    Clear
 </button>
              </div>
            </form>
            </div>
      </section>
<!--

  <h1 class="welcome">IBM PROJECT
  <div id="team_id">TEAM ID : PNT2022TMID19491</div>
  </h1>
  <section id="title">
    <h4 class="heading">Handwritten Digit Recognition Website</h4>
    <br><br>
      <p>
        The website is designed to predict the handwritten digit.
      </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.</p>

      <br>
    <p> 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>

  </section>

-->

  <!--<section id="content">

        <div class="leftside">
        <form action="/predict" method="POST" enctype="multipart/form-data">
        <label>Select a image:</label>
        <input   id="image"   type="file"   name="image"   accept="image/png,
image/jpeg" onchange="preview()"><br><br>
         <img id="frame" width="100px" height="100px"/>
          <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>
```

```
            <button type="submit" class="btn btn-light">Predict</button>
            <button    type="button"    class="btn    btn-light">     Clear
 </button>
          </div>
        </form>
        </div>
  </section>-->

</body>

</html>
```

HOME PAGE



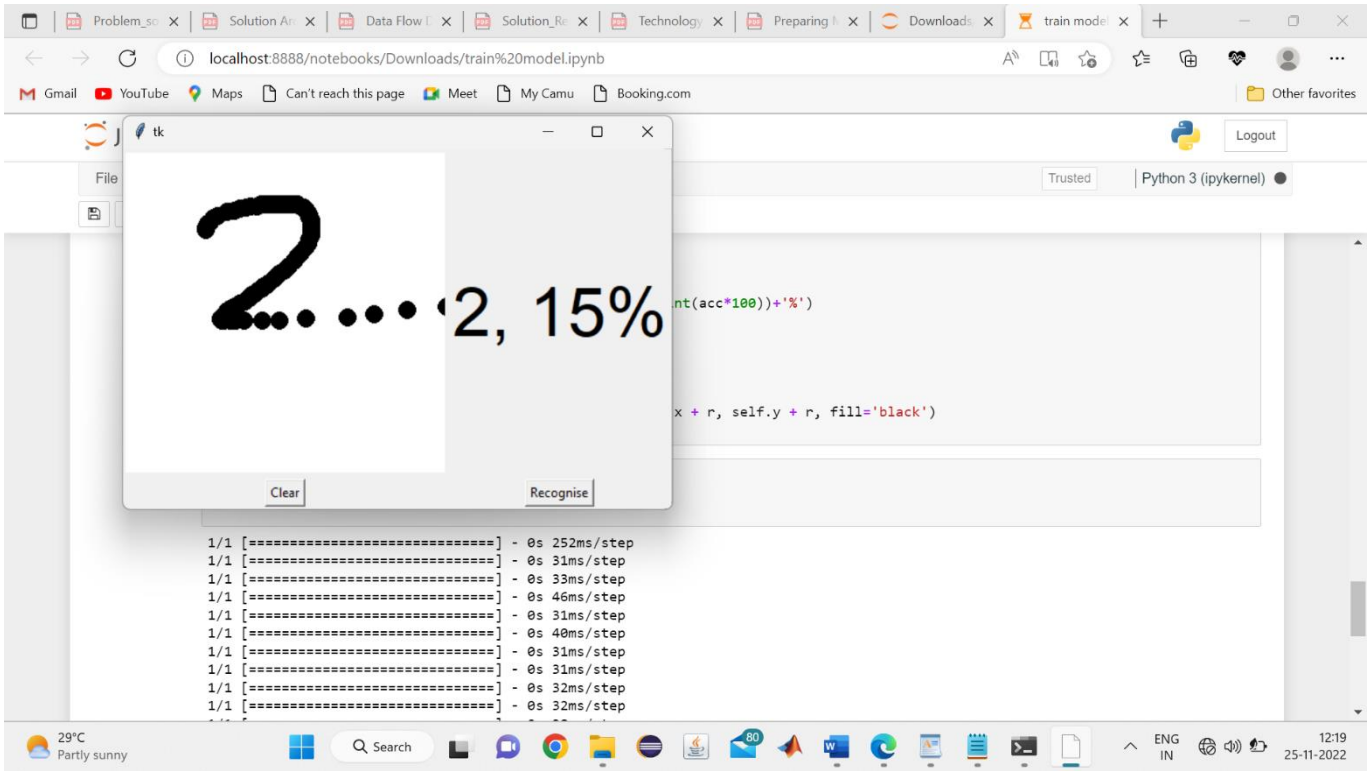
7.2 FEATURE 2

```
import keras
from keras.datasets import mnist
from keras.models import Sequential
from keras.layers import Dense, Dropout, Flatten
from keras.layers import Conv2D, MaxPooling2D
from keras import backend as K

# 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)

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)

# convert class vectors to binary class matrices
y_train = keras.utils.to_categorical(y_train, 10)
y_test = keras.utils.to_categorical(y_test, 10)

x_train = x_train.astype('float32')
x_test = x_test.astype('float32')
x_train /= 255
x_test /= 255
print('x_train shape:', x_train.shape)
print(x_train.shape[0], 'train samples')
print(x_test.shape[0], 'test samples')

batch_size = 128
num_classes = 10
epochs = 10

model = Sequential()
model.add(Conv2D(32, kernel_size=(5, 5),activation='relu',input_shape=input_shape))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.3))
model.add(Dense(64, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(num_classes, activation='softmax'))

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

```
hist = model.fit(x_train, y_train,batch_size=batch_size,epochs=epochs,verbose=1,validation_data=(x_test, y_test))
print("The model has successfully trained")

score = model.evaluate(x_test, y_test, verbose=0)
print('Test loss:', score[0])
print('Test accuracy:', score[1])

model.save('mnist.h5')
print("Saving the model as mnist.h5")
from keras.models import load_model
from tkinter import *
import tkinter as tk
import win32gui
from PIL import ImageGrab, Image
import numpy as np

model = load_model('mnist.h5')

def predict_digit(img):
    #resize image to 28x28 pixels
    img = img.resize((28,28))
    #convert rgb to grayscale
    img = img.convert('L')
    img = np.array(img)
    #reshaping to support our model input and normalizing
    img = img.reshape(1,28,28,1)
    img = img/255.0
    #predicting the class
    res = model.predict([img])[0]
    return np.argmax(res), max(res)

class App(tk.Tk):
    def __init__(self):
        tk.Tk.__init__(self)

        self.x = self.y = 0

        # Creating elements
        self.canvas = tk.Canvas(self, width=300, height=300, bg = "white", cursor="cross")
        self.label = tk.Label(self, text="Draw..", font=("Helvetica", 48))
        self.classify_btn = tk.Button(self, text = "Recognise", command = self.classify_handwriting)
        self.button_clear = tk.Button(self, text = "Clear", command = self.clear_all)

        # Grid structure
        self.canvas.grid(row=0, column=0, pady=2, sticky=W, )
        self.label.grid(row=0, column=1,pady=2, padx=2)
        self.classify_btn.grid(row=1, column=1, pady=2, padx=2)
        self.button_clear.grid(row=1, column=0, pady=2)

        #self.canvas.bind("<Motion>", self.start_pos)
        self.canvas.bind("<B1-Motion>", self.draw_lines)
```

```
    def clear_all(self):
        self.canvas.delete("all")

    def classify_handwriting(self):
        HWND = self.canvas.winfo_id()  # get the handle of the canvas
        rect = win32gui.GetWindowRect(HWND)  # get the coordinate of the canvas
        a,b,c,d = rect
        rect=(a+4,b+4,c-4,d-4)
        im = ImageGrab.grab(rect)

        digit, acc = predict_digit(im)
        self.label.configure(text= str(digit)+', '+ str(int(acc*100))+'%')

    def draw_lines(self, event):
        self.x = event.x
        self.y = event.y
        r=8
        self.canvas.create_oval(self.x-r, self.y-r, self.x + r, self.y + r, fill='black')

app = App()
mainloop()
```

OUTPUT:



8.TESTING

8.1 TEST CASES

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

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	10	0	0	10
Security	1	0	0	1
Outsource Shipping	2	0	0	2
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

8.2 USER ACCEPTANCE TESTING

PURPOSE OF THE DOCUMENT

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

DEFECT ANALYSIS

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	0	4	2	3	9
Duplicate	0	0	3	0	3
External	0	0	0	1	1
Fixed	0	4	5	4	13
Not Reproduced	0	0	0	0	0
Skipped	0	0	0	1	1

Won't Fix	0	0	0	1	1
Totals	0	8	11	10	26

9. RESULTS

9.1 PERFORMANCE METRICS

After implementing all the three algorithms that are SVM, MLP and CNN we have compared their accuracies and execution time with the help of experimental graphs for perspicuous understanding. We have considered the Training and Testing Accuracy of all the models stated above. After executing all the models, we found that SVM has the highest accuracy on training data while on testing dataset CNN accomplishes the utmost accuracy. Additionally, we have compared the execution time to gain more insight into the working of the algorithms. Generally, the running time of an algorithm depends on the number of operations it has performed. So, we have trained our deep learning model up to 30 epochs and SVM models according to norms to get the apt outcome. SVM took the minimum time for execution while CNN accounts for the maximum running time.

10. ADVANTAGES AND DISADVANTAGES

10.1 ADVANTAGES

1. The system not only produces a classification of the digit but also a rich description of the instantiation parameters which can yield information such as writing style.

2. The generative models can perform recognition driven segmentation.

3. Handwriting forces your brain to mentally engage with the information, improving both literacy and reading comprehension.

4. Digitalization

5. Data Collection.

10.2 DISADVANTAGES

1. Despite that there are enormous convolutional neural network algorithms proposed for handwritten digit recognition, issues such as recognition accuracy.

2. Get alternative, less likely predictions when available.

3. Anyway Higher processor is required.

4.High cost

5.Time consuming

6.computation time still require for further improvement.

11. CONCLUSION

Our project HANDWRITTEN DIGIT RECOGNITION deals with identifying the digits. The main purpose of this project is to build an automatic handwritten digit recognition method for the recognition of handwritten digit strings. In this project, different machine learning methods, which are SVM (Support Vector Machine), ANN (Artificial Neural Networks), and CNN (Convolutional Neural Networks) architectures are used to achieve high performance on the digit string recognition problem.

Recognition of characters and digits is viral in today's digitized world, especially in organizations that deal with handwritten documents that they need to analyse using computer systems. 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. A comparison on different Machine Learning algorithms like Random Forest Classifier, Convolutional Neural Network, Linear Regression, K-Nearest Neighbours, Support vector machine is done, in which the accuracy for CNN

is 99.63%. It can be used to convert books, newspapers and handwritten notes into digital text format using machine learning models.

12. FUTURE SCOPE

The proposed system takes 28x28 pixel sized images as input. The same system with further modifications and improvements in the dataset and the model can be used to build Handwritten Character Recognition System which recognizes human handwritten characters and predicts the output. 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

13.REFERENCES

1. https://www.tensorflow.org/learn
2. https://www.geeksforgeeks.org/python-tkinter-tutorial/
3.https://medium.com/the-andela-way/applying-machine-learning-to recognize hand written characters- babcd4b8d705
4. https://nanonets.com/blog/handwritten-character-recognition/
5. https://www.geeksforgeeks.org/python-tkinter-tutorial/
6. https://medium.com/the-andela-way/applying-machine-learning-to- recognize hand written characters- babcd4b8d705
7. https://nanonets.com/blog/handwritten-character-recognition/
8. https://www.tensorflow.org/learn
9. R. Alhajj and A. Elnagar, "Multiagents to separating handwritten connected digits," in IEEE Transactions on Systems, Man, and Cybernetics- Part A: Systems and Humans, vol. 35, no. 5, pp. 593-602, Sept. 2005. https://doi.org/10.1109/TSMCA.2005.843389

10.https://towardsdatascience.com/useful-plots-to-diagnose-your-neural- network-521907fa2f45

11. https://towardsdatascience.com/the-best-machine-learning-algorithm-for-handwritten-digitsrecognition-2c6089ad8f09

12.https://www.datacamp.com/community/tutorials/convolutional-neural- networks-python

13.https://www.analyticsvidhya.com/blog/2021/01/building-a-cnn-model- with-95-accuracy/

14. https://www.analyticsvidhya.com/blog/2020/10/what-is-the convolutional -neural-network architecture/
15. https://www.kdnuggets.com/2018/11/top-python-deep-learning libraries.html
16. https://analyticsindiamag.com/7-types-classification-algorithms/

17.https://towardsdatascience.com/image-recognition-with-machin elearning -on-python-imageprocessing-3abe6b158e9a

18. https://miro.com/app/board/uXjVPNIVgn0=/