

## PROJECT REPORT

# A Novel Method for Handwritten Digit Recognition System

**Submitted by**

<b>REGISTER NUMBER</b>	<b>NAME</b>
<b>612819106023</b>	<b>MUNIVEL M</b>
<b>612819106010</b>	<b>GOKUL SURYA G</b>
<b>612819106031</b>	<b>PREETHI G</b>
<b>612819106047</b>	<b>SRIPRIYA G</b>

<b>1. INTRODUCTION</b>	<b>4</b>
1.1 Project Overview	4
1.2 Purpose	4
<b>2. LITERATURE SURVEY</b>	<b>5</b>
2.1 Existing problem	5
2.2 References	5
2.3 Problem Statement Definition	8
<b>3. IDEATION &amp; PROPOSED SOLUTION</b>	<b>10</b>
3.1 Empathy Map Canvas	10
3.2 Ideation & Brainstorming	11
3.3 Proposed Solution	14
3.4 Problem Solution fit	16
<b>4. REQUIREMENT ANALYSIS</b>	<b>16</b>
4.1 Functional requirement	16
4.2 Non-Functional requirements	16
<b>5. PROJECT DESIGN</b>	<b>19</b>
5.1 Data Flow Diagrams	19
5.2 Solution & Technical Architecture	20
5.3 User Stories	24
<b>6. PROJECT PLANNING &amp; SCHEDULING</b>	<b>27</b>
6.1 Sprint Planning & Estimation	27
6.2 Sprint Delivery Schedule	29
6.3 Reports from JIRA	30
<b>7. CODING &amp; SOLUTIONING</b>	<b>33</b>

7.1	Feature 1	33
7.2	Feature 2	35
8.	TESTING	49
8.1	Test Cases	49
8.2	User Acceptance Testing	51
9.	RESULTS	52
9.1	Performance Metrics	52
10.	ADVANTAGES & DISADVANTAGES	54
11.	CONCLUSION	55
12.	FUTURE SCOPE	56
13.	APPENDIX	57
13.1	Source Code	63
13.2	GitHub & Project Demo Link	63

# **CHAPTER -I**

## **INTRODUCTION**

## **1.1 PROJECT OVERVIEW**

Machine learning and deep learning play an important role in computer technology and artificial intelligence. With the use of deep learning and machine learning, human effort can be reduced in recognizing, learning, predictions and in many more areas.

Handwritten Digit Recognition is the ability of computer systems to recognize handwritten digits from various sources, such as images, documents, and so on. This project aims to let users take advantage of machine learning to reduce manual tasks in recognizing digits.

## **1.2 PURPOSE**

Digit recognition systems are capable of recognizing the digits from different sources like emails, bank cheque, papers, images, etc. and in different real-world scenarios for online handwriting recognition on computer tablets or system, recognize number plates of vehicles, processing bank cheque amounts, numeric entries in forms filled up by hand (tax forms) and so on.

# **CHAPTER -II**

## **LITERATURE REVIEW**

## **2.1 EXISTING PROBLEM**

The fundamental problem with handwritten digit recognition is that handwritten digits do not always have the same size, width, orientation, and margin 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.

## **2.2 REFERENCE**

**[1] Hai Pham, Amrith Setlur, Saket Dingliwal, Tzu-Hsiang Lin, Barnabás Póczos, Kang Huang, Zhuo Li, Jae Lim, Collin McCormack, Tam Vu., "Robust Handwriting Recognition with Limited and Noisy Data" in 17th International Conference on Frontiers in Handwriting Recognition (ICFHR)**

Despite the advent of deep learning in computer vision, the general handwriting recognition problem is far from solved. Most existing approaches focus on handwriting datasets that have clearly written text and carefully segmented labels. In this paper, we instead focus on learning handwritten characters from maintenance logs, a constrained setting where data is very limited and noisy. We break the problem into two consecutive stages of word segmentation and word recognition respectively, and utilize data augmentation techniques to train both stages. Extensive comparisons with popular baselines for scene-text detection and word recognition show that our system achieves a lower error rate and is more suited to handle noisy and difficult documents.

**[2] Anuran Chakraborty, Rajonya De, Samir Malakar, Friedhelm Schwenker and Ram Sarkar, "Handwritten Digit String Recognition using Deep Autoencoder based Segmentation and ResNet based Recognition Approach" in 2020 25th**

**International Conference on Pattern Recognition (ICPR) Milan, Italy, Jan 10-15, 2021**

Recognition of isolated handwritten digits is a wellstudied research problem and several models show high recognition accuracy on different standard datasets. But the same is not true while we consider recognition of handwritten digit strings although it has many real-life applications like bank cheque processing, postal code recognition, and numeric field understanding from filled-in form images. The problem becomes more difficult when digits in the string are not neatly written which is commonly seen in freestyle handwriting. The performance of any such model primarily suffers due to the presence of touching digits in the string. To handle these issues, in the present work, we first use a deep autoencoder based segmentation technique for isolating the digits from a handwritten digit string, and then we pass the isolated digits to a Residual Network (ResNet) based recognition model to obtain the machine-encoded digit string. The proposed model has been evaluated on the Computer Vision Lab (CVL) Handwritten Digit Strings (HDS) database, used in HDSRC 2013 competition on handwritten digit string recognition, and a competent result with respect to state-of-the-art techniques has been achieved.

**[3] Jinze Li, Gongbo Sun, Leiye Yi, Qian Cao, Fusen Liang , Yu Sun, "Handwritten Digit Recognition System Based on Convolutional Neural Network " in 2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA)**

Image recognition is widely used in the field of computer vision today. As a kind of image recognition, digit recognition is widely used. Today, the online recognition technology in digit recognition is relatively mature while the offline recognition technology is not. This paper mainly introduces an offline recognition system for handwritten digits based on convolutional neural networks. The system uses the MINST dataset as a training sample and pre-processes the picture with the Opencv toolkit. Then it

uses LeNet-5 in the convolutional neural network to extract the handwritten digit image features, repeatedly convolution pooling, and pull the result into a one-dimensional vector. And finally find the highest probability point to determine the result to achieve handwritten digit recognition with the Softmax regression model. The application of this system can greatly reduce labor costs and improve work efficiency, which is of great significance in many fields.

**[4] Hao Zeng, "An Off-line Handwriting Recognition Employing Tensorflow" in 2020 International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE).**

Abstract-Handwriting has been a conventional means of communication and recording in daily life since early time. Given its ubiquity in human transactions, machine recognition of handwriting has practical significance, such as, in reading handwritten notes in a PDA, in postal addresses on envelopes, in amounts in bank checks, or in handwritten fields in forms. Handwriting recognition is a vital application in daily activities and the researches of especially handwritten digits recognition is vital. This paper focuses on using simpler neural network instead of complicated ones that require high quality of computer configuration to recognize handwriting digits with relatively promising accuracy. To do this research, a neural network to recognize handwriting in MNIST dataset using Softmax Regression algorithm with a high accuracy is built.

**[5] Ayush Kumar Agrawal; A.K. Shrivastava; Vineet Kumar Awasthi, A Robust Model for Handwritten Digit Recognition using Machine and Deep Learning Technique in 2021 2nd International Conference for Emerging Technology (INCET).**

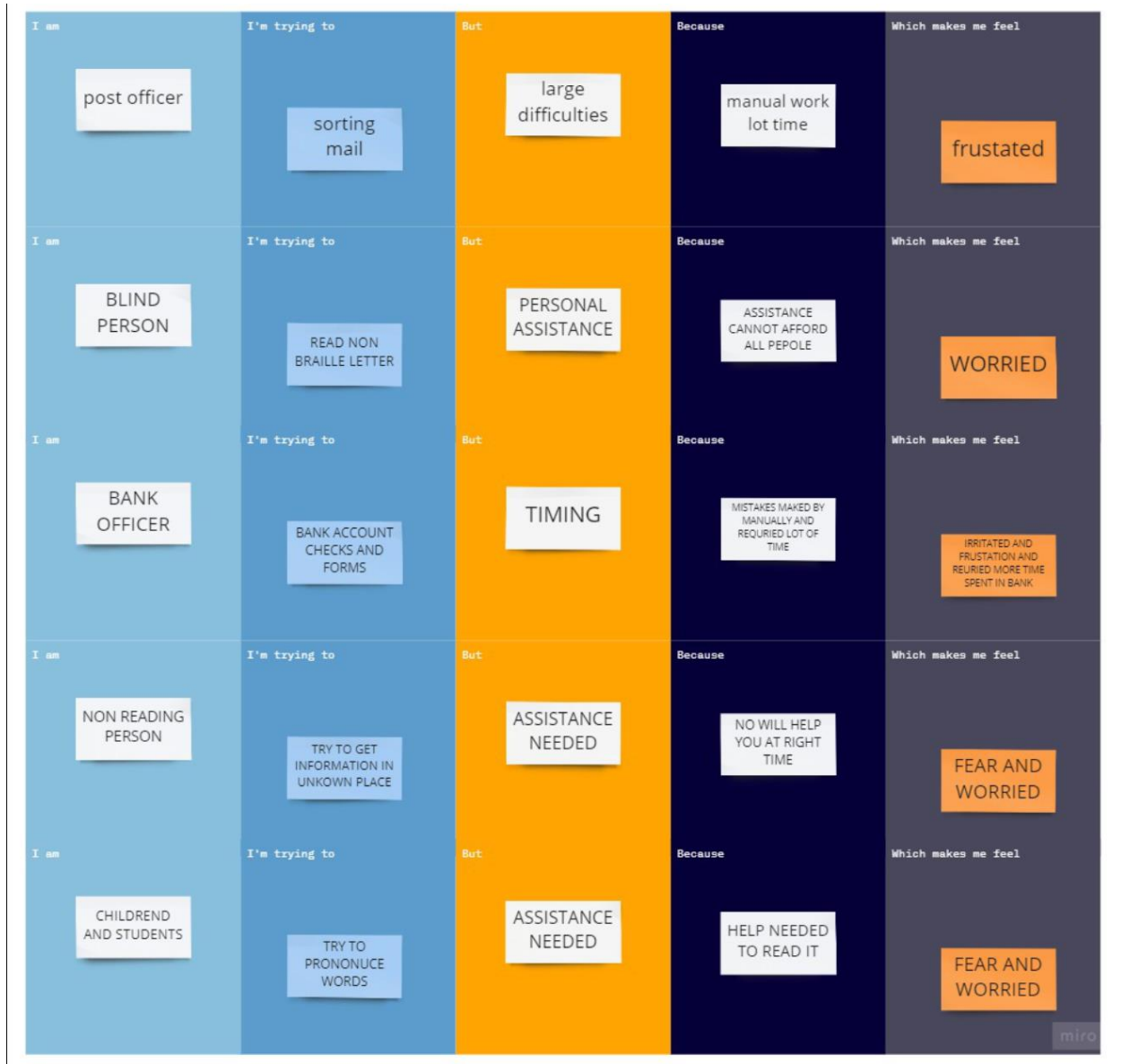
In the era of research, pattern recognition is one of the most famous and widely used area in the field of research work. There are various types of patterns are available for the researches like: audio, video, handwritten digit images and handwritten characters images

etc. In this paper, we concentrate in the field of handwritten digit recognition for classification of patterns. We have used famous handwritten digit datasets named as MNIST, which is collection of 70000 images. Many of machine learning and deep learning techniques have been already used by the researches for handwritten digit recognition like Support Vector Machine (SVM), RFC, K-nearest Neighbor (K-NN), Multilayer Perceptron (MLP), Convolutional Neural Network (CNN) etc. In this research work, we have suggested CNN as deep learning technique on keras for MNIST handwritten digit recognition and compare the performance of CNN with SVM and KNN. The proposed CNN based on keras model used to classify handwritten digit images with RMSprop optimizer for optimizing the model. The main contribution of this research work is to increase the convolutional layer with pooling and dropout and also tuned the model using filter, kernel size and number of neurons. The proposed CNN model achieves 99.06% of training accuracy and 98.80% of testing accuracy with epoch 10. Experiment results reveals that proposed CNN is more effective compare to other techniques.

## **2.3 PROBLEM STATEMENT DEFINITION**

For years, the traffic department has been combating traffic law violators. These offenders endanger not only their own lives, but also the lives of other individuals. Punishing these offenders is critical to ensuring that others do not become like them. Identification of these offenders is next to impossible because it is impossible for the average individual to write down the license plate of a reckless driver. Therefore, the goal of this project is to help the traffic department identify these offenders and reduce traffic.

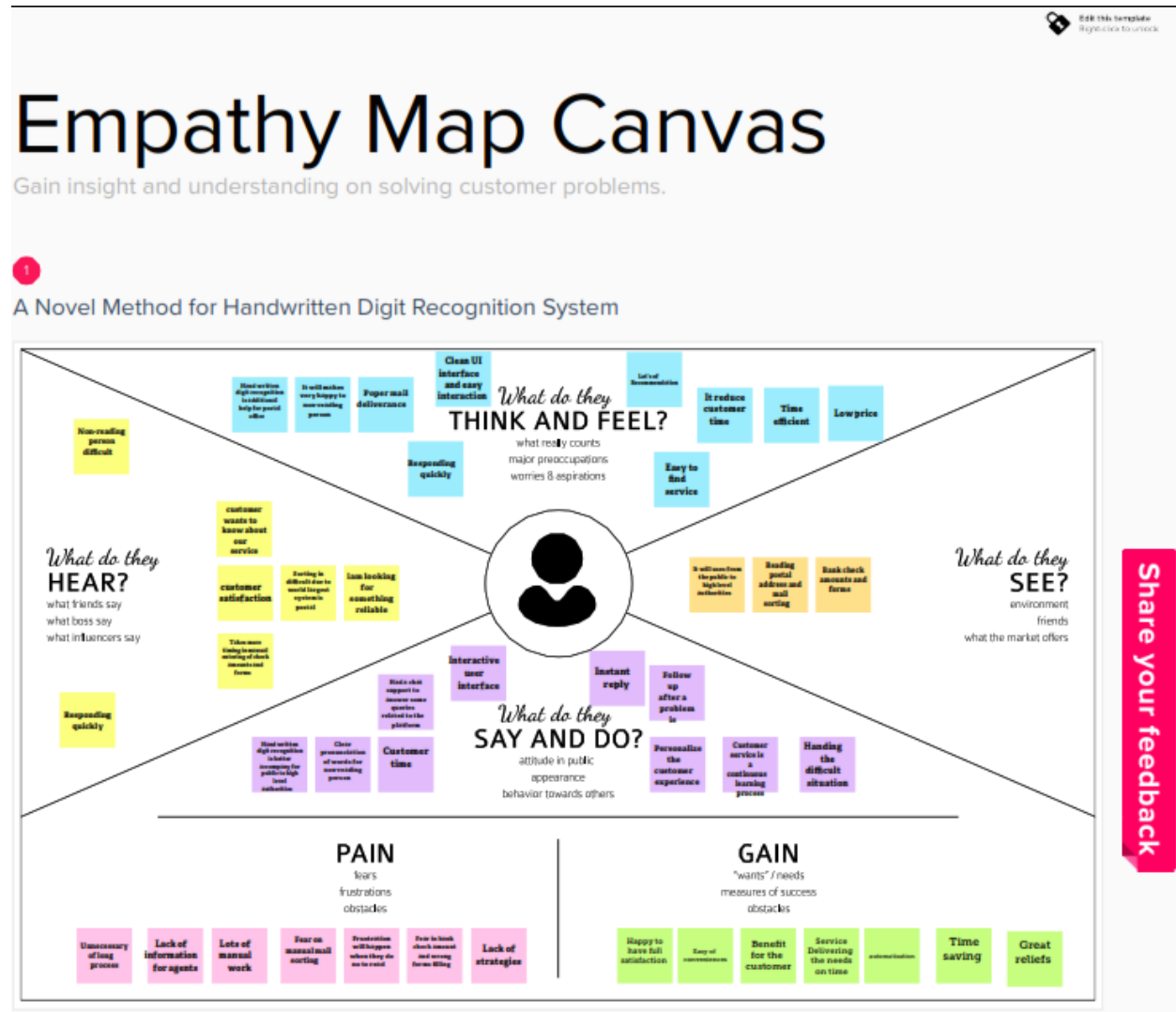




## CHAPTER - 3

# IDEATION & PROPOSED SOLUTION

## 3.1 EMPATHY MAP CANVAS



## 3.2 IDEATION & BRAINSTORMING



## Brainstorm & idea prioritization

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

🕒 10 minutes to prepare

🕒 1 hour to collaborate

👤 4 people recommended

💬 Share template feedback



### Need some inspiration?

See a finished version of this template to kickstart your work.

[Open example](#) →



### Before you collaborate

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

🕒 10 minutes



#### Team gathering

TEAM LEADER - MUNIVEL M  
TEAM MEMBER 1 - GOKULSURYA G  
TEAM MEMBER 2 - PREETHI G  
TEAM MEMBER 3 - SRIPRIYA G



#### Set the goal

Think about the problem you'll be focusing on solving in the brainstorming session.



#### Learn how to use the facilitation tools

Use the Facilitation Superpowers to run a happy and productive session.

[Open article](#) →



### Define your problem statement

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

🕒 5 minutes

#### PROBLEM

How to design hand recognition system with use friendly way?



#### Key rules of brainstorming

To run an smooth and productive session



Stay in topic.



Encourage wild ideas.



Defer judgment.



Listen to others.



Go for volume.



If possible, be visual.

2

## Brainstorm

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

10 minutes

TIP

You can select a sticky note and hit the pencil icon to edit it (or to start drawing)!

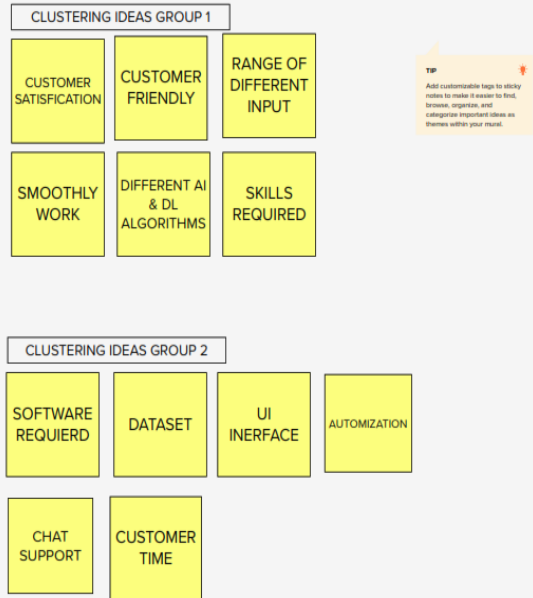


3

## Group Ideas

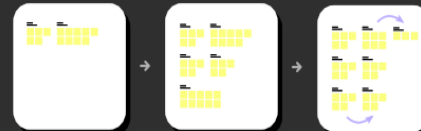
Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you can break it up into smaller sub-groups.

20 minutes



TIP

Add customizable tags to sticky notes to make it easier to find, browse, organize and categorize important ideas as themes within your mural.

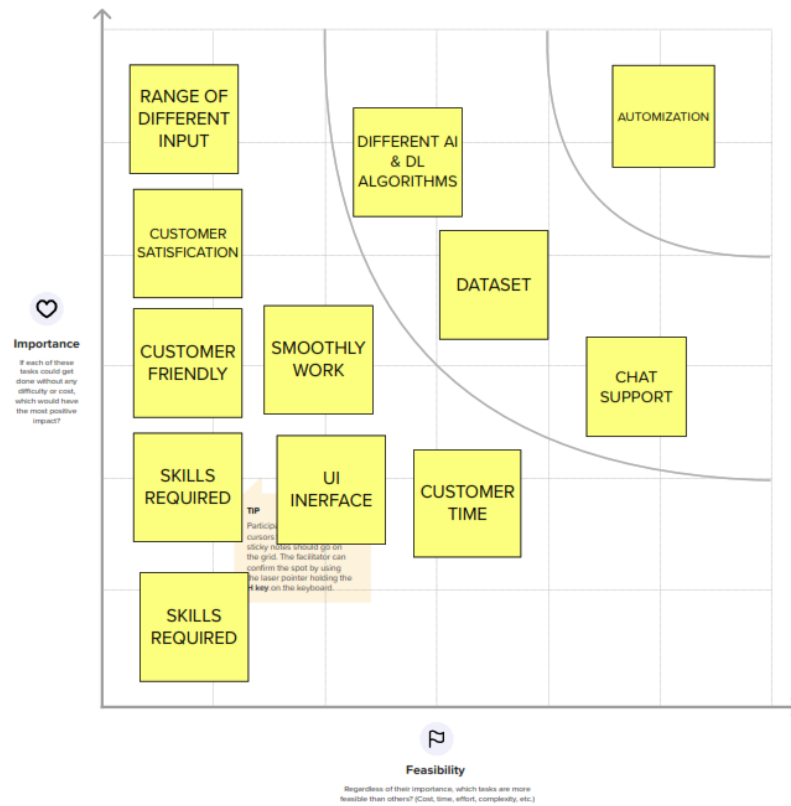


4

### 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



➔

### After you collaborate

You can export the mural as an image or pdf to share with members of your company who might find it helpful.

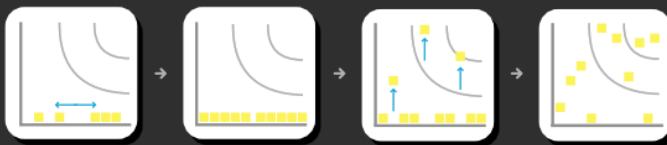
#### Quick add-ons

- A Share the mural**  
Share a view link to the mural with stakeholders to keep them in the loop about the outcomes of the session.
- B Export the mural**  
Export a copy of the mural as a PNG or PDF to attach to emails, include in slides, or save in your drive.

#### Keep moving forward

- Strategy blueprint**  
Define the components of a new idea or strategy.  
[Open the template →](#)
- Customer experience journey map**  
Understand customer needs, motivations, and obstacles for an experience.  
[Open the template →](#)
- Strengths, weaknesses, opportunities & threats**  
Identify strengths, weaknesses, opportunities, and threats (SWOT) to develop a plan.  
[Open the template →](#)

[Share template feedback](#)



### 3.3 PROPOSED SOLUTION:

The main objective of this project is to convert handwritten digits into machine-readable formats and make banking operations, vehicle number detection, form data entry easier and error free.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	<ol style="list-style-type: none"><li>1. Huge variability and ambiguity of strokes from person to person</li><li>2. Handwriting style of an individual person also varies time to time and is inconsistent</li><li>3. Poor quality of the source document/image due to degradation over time</li><li>4. Text in printed documents sit in a straight line whereas humans need not write a line of text in a straight line on white paper</li><li>5. Cursive handwriting makes separation and recognition of characters challenging</li><li>6. Text in handwriting can have variable rotation to the right which is in contrast to printed text where all the text sits up straight</li><li>7. Collecting a good labelled dataset to learn is not cheap compared to synthetic data</li></ol>
2.	Idea / Solution description	Using MNIST dataset over the neural network algorithms, it is possible to recognize the digits which is useful for banks sectors, data entry etc.

3.	Novelty / Uniqueness	Using Convolutional Neural Network(CNN) gives greater accuracy and it can detect automatically without any human supervision.
4.	Social Impact / Customer Satisfaction	working hard on machines to make them more smart and intelligent by using machine learning and deep learning techniques so that they can perform tasks similar to humans. With the help of these techniques <i>human effort can be reduced and much time can be saved in recognizing, learning, predictions and many other areas.</i>
5.	Business Model (Revenue Model)	Collaboration with bank sectors, government sectors and individual app for non-readable person.
6.	Scalability of the Solution	The handwriting will be detected by any of the formats such as image,documents , etc.. so that it can be user friendly and flexible where there willbe a growth for the users.

### 3.4 PROBLEM SOLUTION FIT

Define CS, fit into CC	<p><b>1. CUSTOMER SEGMENT(S)</b> <span>CS</span> Who is your customer?</p> <ol style="list-style-type: none"> <li>1. Bank sector</li> <li>2. Public sector</li> <li>3. Post office</li> <li>4. Blind person</li> <li>5. number plate recognition</li> <li>7. Online recognition</li> <li>8. Offline recognition</li> <li>9. Signature verification</li> <li>10. Postal address interpretation</li> <li>11. Bank-Check processing</li> </ol>	<p><b>6. CUSTOMER CONSTRAINTS</b> <span>RC</span> What constraints prevent your customers from taking action or limit their choices of solutions?</p> <ol style="list-style-type: none"> <li>1. Spending more time in queue and sorting mails take long time</li> <li>2. Not enough method to recognize input data in previous solution.</li> <li>3. Spending data for online mode.</li> <li>4. Requires much more computation cannot determine symbols, age, personality.</li> </ol>	<p><b>5. AVAILABLE SOLUTIONS</b> <span>AS</span> Which solutions are available to the customers when they face the problem</p> <ol style="list-style-type: none"> <li>1. Different styles of jotting of different peoples as it is not an Optic character recognition.</li> <li>2. Separate digit give good accuracy Has holistic method estimate complicate</li> <li>3. segmentation and quickly perform the task using dataset</li> </ol>	Explore AS, differentiate
	<p><b>2. JOBS-TO-BE-DONE / PROBLEMS</b> <span>J&amp;P</span> Which jobs-to-be-done (or problems) do you address for your</p> <ol style="list-style-type: none"> <li>1. Problems with letter shapes. Problems with spacing.</li> <li>2. Problems with grip and posture.</li> <li>3. Many algorithms have been developed to recognize handwritten digits.</li> <li>4. Due to infinity variety of writing styles, they are still inadequate customers? There could be more than one; explore different sides.</li> </ol>	<p><b>9. PROBLEM ROOT CAUSE</b> <span>RC</span> What is the real reason that this problem exists? What is the back story behind the need to do this job?</p> <p>The high variance in handwriting styles across people and poor quality of the handwritten text compared to printed text pose significant hurdles in converting it to machine readable text. Nevertheless it's a crucial problem to solve for multiple industries like healthcare, insurance and banking.</p>	<p><b>7. BEHAVIOUR</b> <span>BE</span> What does your customer do to address the problem and get the job done?</p> <p>working hard on machines to make them more smart and intelligent by using machine learning and deep learning techniques so that they can perform tasks similar to humans. With the help of these techniques <b>human effort can be reduced and much time can be saved in recognizing, learning, predictions and many other areas.</b></p>	
Identify strong TR & EM	<p><b>3. TRIGGERS</b> <span>TR</span> What triggers customers to act?</p> <p>Digital automatization of world, example work in postal, number plate recognition</p>	<p><b>10. YOUR SOLUTION</b> <span>SL</span> If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality.</p> <p>The main objective of this work is to ensure effective and reliable approaches for recognition of handwritten digits and make operations like vehicle number plate detection, postal locations, data entry easier and error-free. This method is for increasing efficiency of the learning algorithm by preprocessing the images and increasing The performance for real time application. With the usage of MNIST technology database accuracy is obtained..</p>	<p><b>8. CHANNELS of BEHAVIOUR</b> <span>CH</span> <b>ONLINE</b> What kind of actions do customers take online?</p> <p>recognition is performed when digits are under creation</p>	Extract online & offline CH of BE
	<p><b>4. EMOTIONS: BEFORE / AFTER</b> <span>EM</span> How do customers feel when they face a problem or a job and afterwards?</p> <p>Before Depression ,anxiety, stress</p> <p>After Feeling smart, active and better approach.</p>		<p><b>OFFLINE</b> What kind of actions do customers take offline?</p> <p>first document are generated , scanned, stored in computer and they are recognized.</p>	



## **CHAPTER -4**

### **REQUIREMENT ANALYSIS**

#### **4.1 FUNCTIONAL REQUIREMENTS**

Following are the functional requirements of the proposed solution.

<b>FR No.</b>	<b>Functional Requirement (Epic)</b>	<b>Sub Requirement (Story / Sub-Task)</b>
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User interface	API interface to system should be friendly
FR-4	User feedback	Submit their queries and feedback forms about app/software (24*7 hours)
FR-5	Storage	Storage data in cloud services

#### **4.2 NON FUNCTIONAL REQUIREMENTS**

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

<b>FR No.</b>	<b>Non-Functional Requirement</b>	<b>Description</b>	<b>Requirement</b>
NFR-1	<b>Usability</b>	How effectively users can learn, use or control the solution/system How easy is it to learn and use? Does it support accessibility standards	User friendly

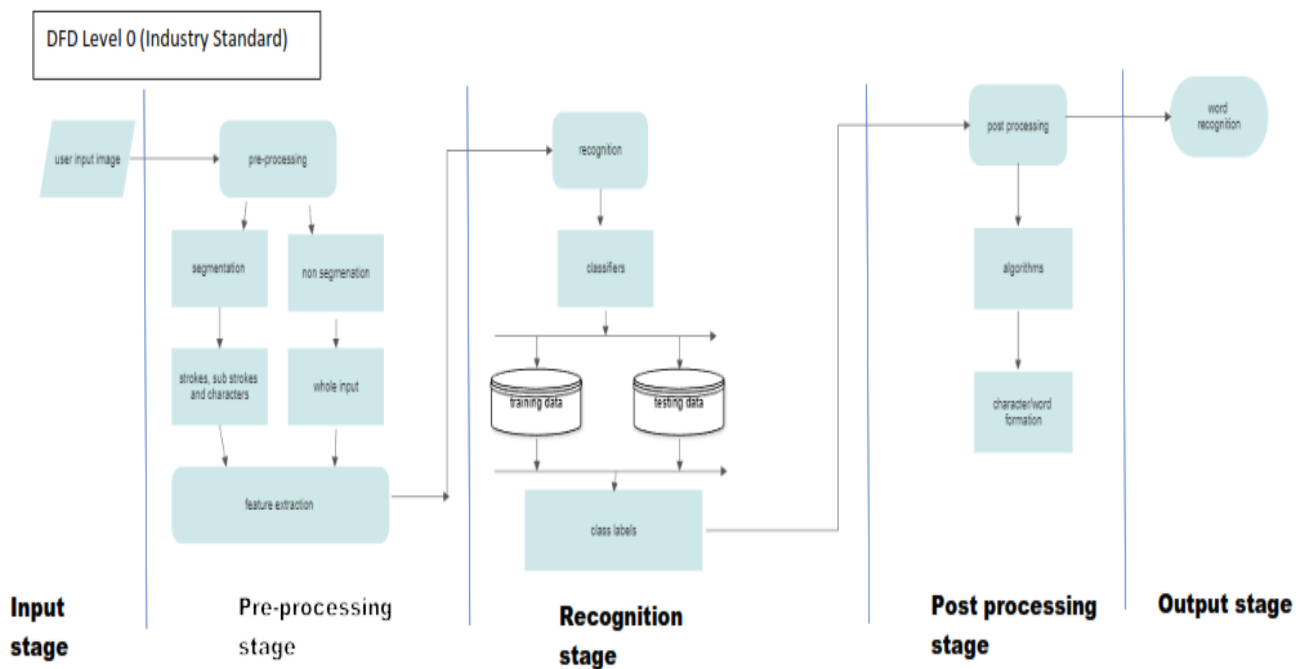
NFR-2	<b>Security</b>	A system's ability to prohibited unauthorized access, usage, or behaviour modification while providing service to authorized users	prohibited unauthorized services
NFR-3	<b>Reliability</b>	The system can perform without failure in 95 percent of the user cases	Accuracy in find hand written recognized
NFR-4	<b>Performance</b>	Performance can measured using execution time, throughput and accuracy	Response time
NFR-5	<b>Availability</b>	The web dashboard must be available to all users 99.98 percent of the time every month during the business hours	Available at business hours
NFR-6	<b>Scalability</b>	The ability of the solution or system to increase its capacity to server clients and/ or increasing processing rate to match demand	The number of concurrent players in the app/software/ web dashboard

# CHAPTER 5

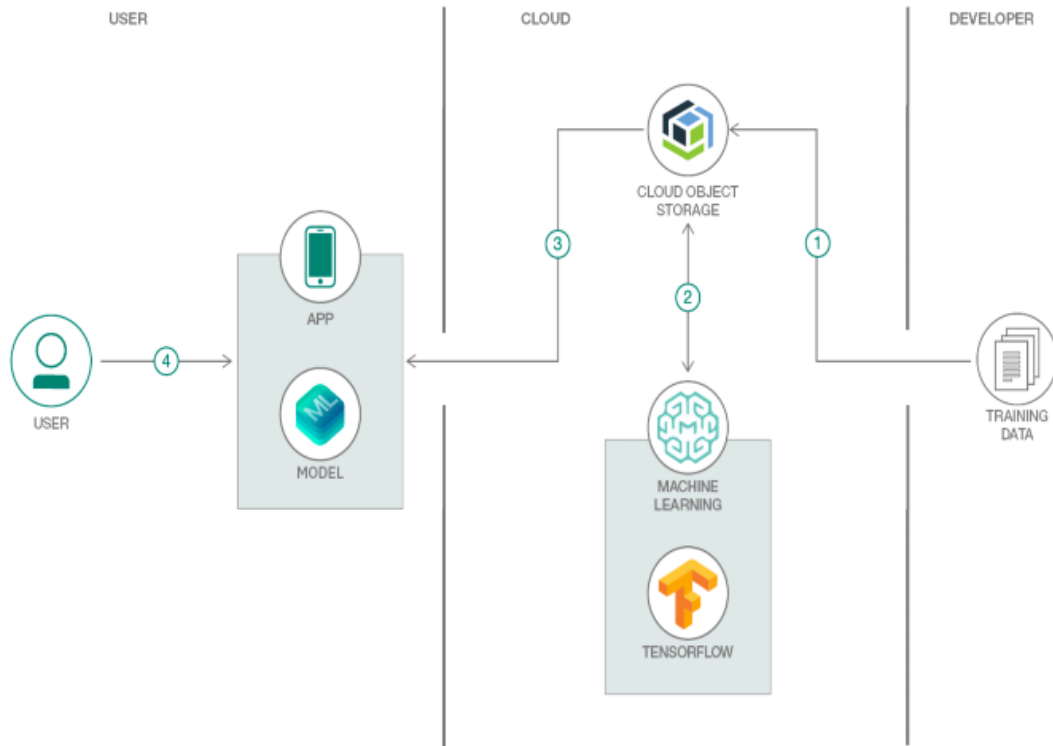
## PROJECT DESIGN

### 5.1 DATA FLOW DIAGRAM

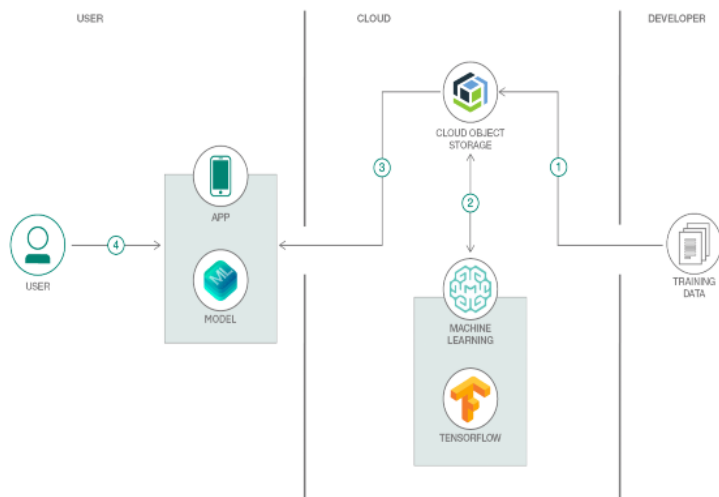
A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.



## FLOW DIAGRAM (SMPLIFIED)

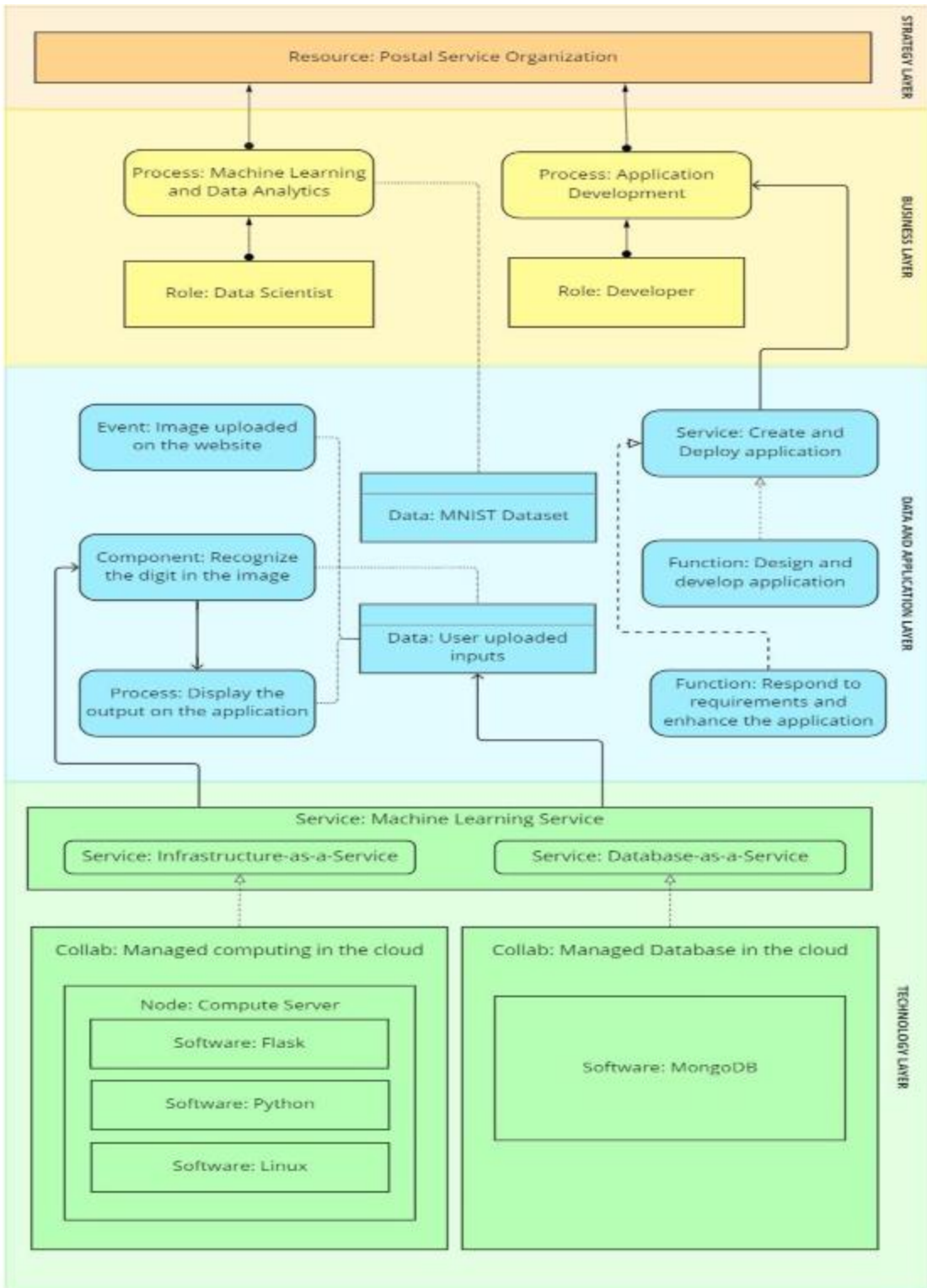


## 5.2 SOLUTION & TECHNICAL ARCHITECTURE



### Guidelines:

1. Upload the training data to IBM Cloud Object Storage.
2. Watson Machine Learning pulls the training data from IBM Cloud Object Storage and trains a model with TensorFlow. The trained model is saved back to IBM Cloud Object Storage.
3. The trained models are added to the app.
4. The user interacts with the apps that can detect objects in real time.



**Table-1 : Components & Technologies:**

<b>S.No</b>	<b>Component</b>	<b>Description</b>	<b>Technology</b>
1.	User Interface	How user interacts with application e.g. Web UI, Mobile App, Chatbot etc.	HTML, CSS, JavaScript / Angular Js / React Js etc.
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	Data Type, Configurations etc.	MySQL, NoSQL, etc.
6.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc.
7.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem
8.	External API-1	Purpose of External API used in the application	IBM Weather API, etc.
9.	External API-2	Purpose of External API used in the application	Aadhar API, etc.
10.	Machine Learning Model	Purpose of Machine Learning Model	Object Recognition Model, etc.
11.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration:	Local, Cloud Foundry, Kubernetes, etc.

		Cloud Server Configuration :	
--	--	------------------------------	--

**Table-2: Application Characteristics:**

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used	Technology of Opensource framework
2.	Security Implementations	List all the security / access controls implemented, use of firewalls etc.	e.g. SHA-256, Encryptions, IAM Controls, OWASP etc.
3.	Scalable Architecture	Justify the scalability of architecture (3 – tier, Micro-services)	Technology used
4.	Availability	Justify the availability of application (e.g. use of load balancers, distributed servers etc.)	Technology used
5.	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc.	Technology used

### 5.3 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)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1



User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail		Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password		High	Sprint-1

<b>User Type</b>	<b>Functional Requirement (Epic)</b>	<b>User Story Number</b>	<b>User Story / Task</b>	<b>Acceptance criteria</b>	<b>Priority</b>	<b>Release</b>
	Dashboard					
Customer (Web user)						
Customer Care Executive						
Administrator						

# CHAPTER - 6

## PROJECT PLANNING & SCHEDULING

### 6.1 SPRINT PLANNING & ESTIMATION

#### JIRA SOFTWARE ROADMAP OF OUR PROJECT

<https://pnt2022tmid41196.atlassian.net/jira/software/projects/NOV/boards/2/roadmap?shared=&atlOrigin=eyJpIjoiNWZMmYzMjJjZGMwNDA2YWEwNTgyOTYxZjQwMjA1NzEiLCJwIjoiajJ9>

<b>Sprint</b>	<b>Functional Requirement (Epic)</b>	<b>User Story Number</b>	<b>User Story / Task</b>	<b>Story Points</b>	<b>Priority</b>	<b>Team Members</b>
Sprint-1	Registration (NOV-6)	NOV-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	M Munivel
Sprint-1		NOV -2	As a user, I will receive confirmation email once I have registered for the application	1	High	G Gokul surya
Sprint-2		NOV -3	As a user, I can register for the application	2	Low	G Gokul surya

<b>Sprint</b>	<b>Functional Requirement (Epic)</b>	<b>User Story Number</b>	<b>User Story / Task</b>	<b>Story Points</b>	<b>Priority</b>	<b>Team Members</b>
			through Facebook			
Sprint-1		NOV -4	As a user, I can register for the application through Gmail	2	Medium	G Preethi
Sprint-1	Login (NOV-7)	NOV -5	As a user, I can log into the application by entering email & password	1	High	G Sripriya
Sprint-2	Dashboard (NOV-8)	NOV-9	See a list of all member's and visitors	1	Low	M Munivel
Sprint-2		NOV-10	Add new security groups	1	High	G Gokul surya
Sprint-2		NOV-11	Change my password	1	High	G Preethi
Sprint-3		NOV-12	Update my contact details	2	High	G Sripriya
Sprint-3		NOV-13	Upadate my email preferences	2	High	M Munivel
Sprint-3		NOV-19	Add comments	1	Low	G Gokul surya

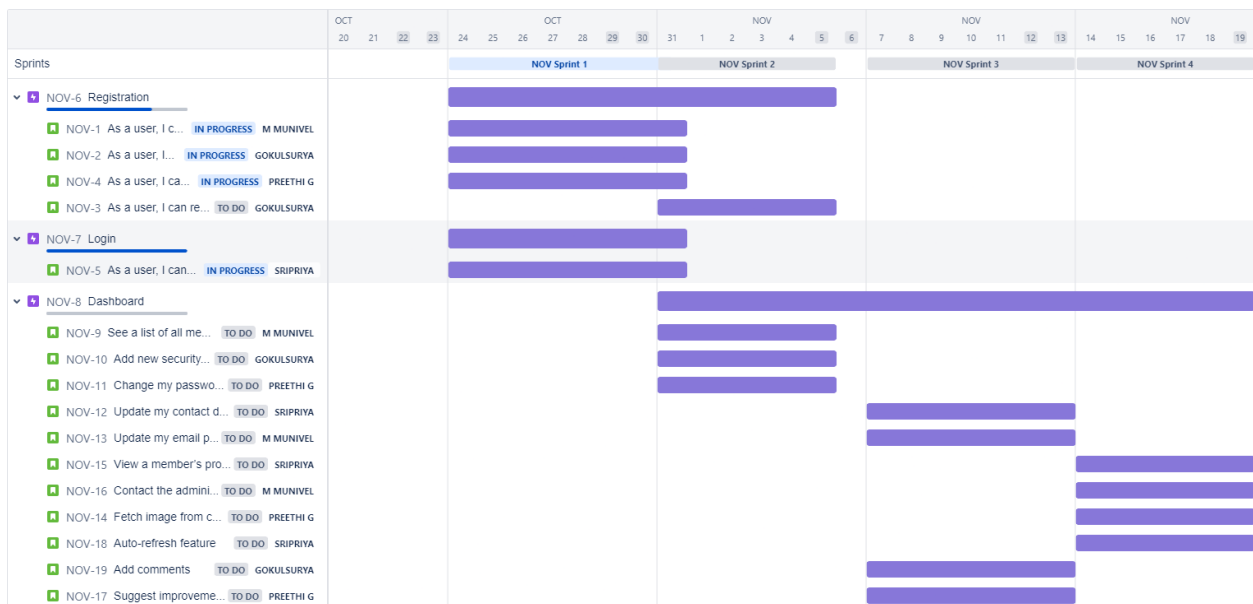
<b>Sprint</b>	<b>Functional Requirement (Epic)</b>	<b>User Story Number</b>	<b>User Story / Task</b>	<b>Story Points</b>	<b>Priority</b>	<b>Team Members</b>
Sprint-3		NOV-17	Suggest improvements	1	Low	G Preethi
Sprint-4		NOV-16	Contact the administrators	1	Low	G Sripriya
Sprint-4		NOV-15	View a member's profile	1	Low	M Munivel
Sprint-4		NOV-14	Fetch image from camera and from gallery	2	High	G Preethi
Sprint-4		NOV-18	Auto-refresh feature	1	Low	G Sripriya

## 6.2 SPRINT DELIVERY SCHEDULE

<b>Sprint</b>	<b>Total Story Points</b>	<b>Duration</b>	<b>Sprint Start Date</b>	<b>Sprint End Date (Planned)</b>	<b>Story Points Completed (as on Planned End Date)</b>	<b>Sprint Release Date (Actual)</b>
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	6	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	6	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	6	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	5	19 Nov 2022

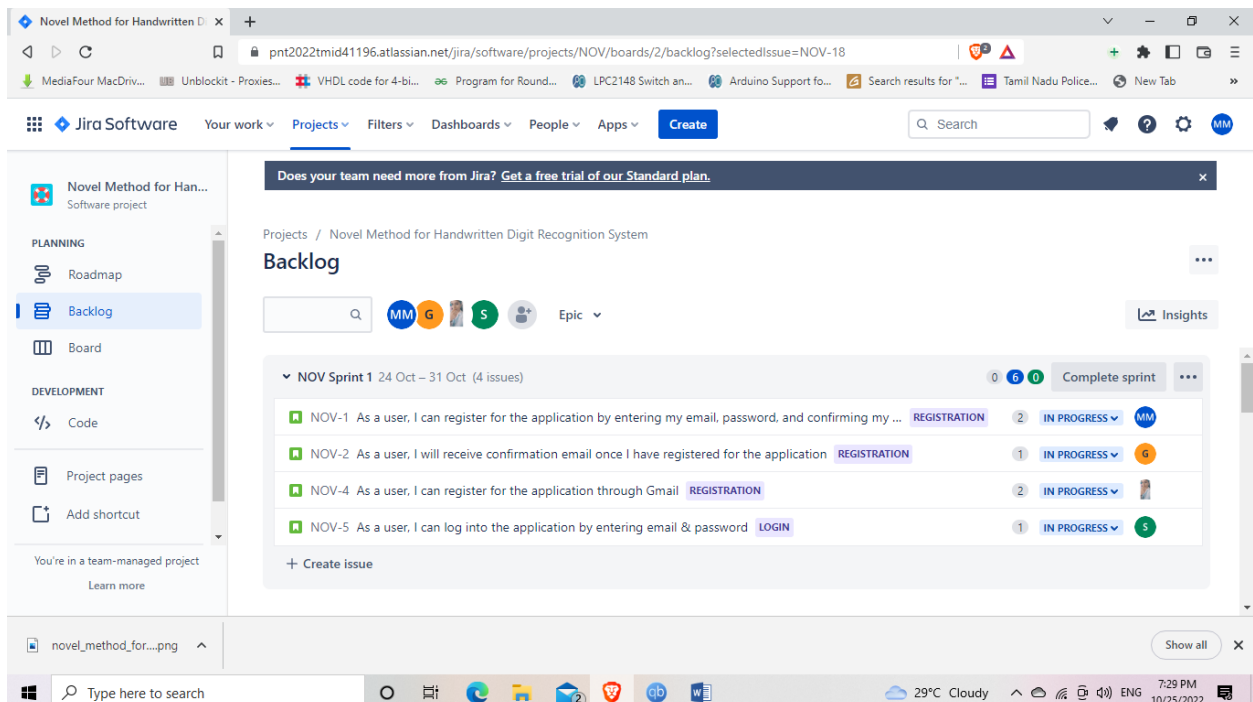
## 6.3 REPORTS FROM JIRA

### 6.3.1 ROADMAPS



## 6.3.2 PRODUCT BACKLOGS

### SPRINT -1



## SPRINT -2

The screenshot shows the Jira Software interface for a project named "Novel Method for Handwritten Digit Recognition System". The left sidebar contains navigation options: PLANNING (Roadmap, Backlog, Board) and DEVELOPMENT (Code, Project pages, Add shortcut, Project settings). The main area displays the "Backlog" for "NOV Sprint 2" (31 Oct – 5 Nov, 4 issues). The backlog items are:

- NOV-3 As a user, I can register for the application through Facebook (REGISTRATION) - 2 TO DO - 6
- NOV-9 See a list of all member's and visitors (DASHBOARD) - 1 TO DO - MM
- NOV-10 Add new security groups (DASHBOARD) - 1 TO DO - 6
- NOV-11 Change my password (DASHBOARD) - 2 TO DO - 1

Below the sprint, there is a section for "NOV Sprint 3" (7 Nov – 13 Nov, 4 issues) with a "Start sprint" button. The Windows taskbar at the bottom shows the date as 10/25/2022 and time as 7:30 PM.

## SPRINT-3

The screenshot shows the Jira Software interface for the same project. The left sidebar is identical to the previous screenshot. The main area displays the "Backlog" for "NOV Sprint 3" (7 Nov – 13 Nov, 4 issues). The backlog items are:

- NOV-12 Update my contact details (DASHBOARD) - 2 TO DO - 5
- NOV-13 Update my email preferences (DASHBOARD) - 2 TO DO - MM
- NOV-19 Add comments (DASHBOARD) - 1 TO DO - 6
- NOV-17 Suggest improvements (DASHBOARD) - 1 TO DO - 1

Below the sprint, there is a section for "NOV Sprint 4" (14 Nov – 19 Nov, 4 issues) with a "Start sprint" button. The Windows taskbar at the bottom shows the date as 10/25/2022 and time as 7:31 PM.

# SPRINT-4

The screenshot displays the Jira Software web interface. The browser address bar shows the URL: `pnt2022tmid41196.atlassian.net/jira/software/projects/NOV/boards/2/backlog?selectedIssue=NOV-18`. The Jira header includes navigation links like 'Your work', 'Projects', 'Filters', 'Dashboards', 'People', and 'Apps', along with a 'Create' button and a search bar. A sidebar on the left lists project navigation options: 'Roadmap', 'Backlog' (selected), 'Board', 'Code', 'Project pages', 'Add shortcut', and 'Project settings'. A banner at the top of the main content area asks: 'Does your team need more from Jira? Get a free trial of our Standard plan.' Below this, the breadcrumb 'Projects / Novel Method for Handwritten Digit Recognition System' is shown. The main section is titled 'Backlog' and features a search bar, user avatars (MM, G, S), and an 'Epic' dropdown. A 'NOV Sprint 4' (14 Nov - 19 Nov, 4 issues) is expanded, showing a list of issues:

Issue ID	Issue Title	Label	Priority	Status	Assignee
NOV-15	View a member's profile	DASHBOARD	1	TO DO	S
NOV-16	Contact the administrators	DASHBOARD	1	TO DO	MM
NOV-14	Fetch image from camera and from gallery	DASHBOARD	2	TO DO	G
NOV-18	Auto-refresh feature	DASHBOARD	1	TO DO	S

At the bottom of the interface, the Windows taskbar is visible, showing the search bar, application icons, and system tray information including 'Polluted air', '7:31 PM', and '10/25/2022'.



# CHAPTER 7

## CODING & SOLUTIONING

### 7.1 FEATURE 1

```
import numpy as np
import os
from PIL import Image
from flask import Flask, request, render_template, url_for
from werkzeug.utils import secure_filename, redirect
#from gevent.pywsgi import WSGIServer
from keras.models import load_model
from keras.preprocessing import image
from flask import send_from_directory

UPLOAD_FOLDER = r'C:\Users\nithy\handwrtten\data'

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

model = load_model(r"C:\Users\nithy\mnistCNN.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(debug=True, threaded=False)
```

## 7.2 FEATURE 2

```
{
  "cells": [
    {
      "cell_type": "markdown",
      "id": "379a204c",
      "metadata": {},
      "source": [
        "# Model Building"
      ]
    },
    {
      "cell_type": "code",
      "execution_count": 1,
      "id": "e4941749",
      "metadata": {},
      "outputs": [],
      "source": [
        "import numpy\n",
        "import matplotlib.pyplot as plt\n",
        "from keras.utils import np_utils\n",
        "from tensorflow.keras.datasets import mnist\n",
        "from tensorflow.keras.models import Sequential\n",
        "from tensorflow.keras.layers import Conv2D, Dense, Flatten\n",
        "from tensorflow.keras.optimizers import Adam"
      ]
    },
    {
      "cell_type": "code",
      "execution_count": 2,
      "id": "36f393b7",
```

```
"metadata": {},
"outputs": [],
"source": [
    "(X_train, y_train), (X_test, y_test) = mnist.load_data()"
]
},
{
    "cell_type": "code",
    "execution_count": 3,
    "id": "8b15642f",
    "metadata": {},
    "outputs": [
        {
            "name": "stdout",
            "output_type": "stream",
            "text": [
                "(60000, 28, 28)\n",
                "(10000, 28, 28)\n"
            ]
        }
    ],
    "source": [
        "print(X_train.shape)\n",
        "print(X_test.shape)"
    ]
},
{
    "cell_type": "code",
    "execution_count": 4,
    "id": "3c9d9bdc",
    "metadata": {},
```

```

"outputs": [
{
  "data": {
    "text/plain": [
      "array([[ 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,\n",
      "         0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,\n",
      "         0,  0],\n",
      "        [ 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,\n",
      "         0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,\n",
      "         0,  0],\n",
      "        [ 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,\n",
      "         0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,\n",
      "         0,  0],\n",
      "        [ 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,\n",
      "         0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,\n",
      "         0,  0],\n",
      "        [ 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,\n",
      "         0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,\n",
      "         0,  0],\n",
      "        [ 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  3,\n",
      "        18, 18, 18, 126, 136, 175, 26, 166, 255, 247, 127,  0,  0,\n",
      "         0,  0],\n",
      "        [ 0,  0,  0,  0,  0,  0,  0,  0, 30, 36, 94, 154, 170,\n",
      "        253, 253, 253, 253, 253, 225, 172, 253, 242, 195, 64,  0,  0,\n",
      "         0,  0],\n",
      "        [ 0,  0,  0,  0,  0,  0,  0, 49, 238, 253, 253, 253, 253,\n",
      "        253, 253, 253, 253, 251, 93, 82, 82, 56, 39,  0,  0,  0,\n",
      "         0,  0],\n",
      "        [ 0,  0,  0,  0,  0,  0,  0, 18, 219, 253, 253, 253, 253,\n",
      "        253, 198, 182, 247, 241,  0,  0,  0,  0,  0,  0,  0,  0,\n",
      "         0,  0],\n",
      "        [ 0,  0],\n",
    ]
  }
}

```

" [ 0, 0, 0, 0, 0, 0, 0, 0, 80, 156, 107, 253, 253,\n",  
" 205, 11, 0, 43, 154, 0, 0, 0, 0, 0, 0, 0, 0,\n",  
" 0, 0],\n",  
" [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 14, 1, 154, 253,\n",  
" 90, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,\n",  
" 0, 0],\n",  
" [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 139, 253,\n",  
" 190, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,\n",  
" 0, 0],\n",  
" [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 11, 190,\n",  
" 253, 70, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,\n",  
" 0, 0],\n",  
" [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 35,\n",  
" 241, 225, 160, 108, 1, 0, 0, 0, 0, 0, 0, 0, 0,\n",  
" 0, 0],\n",  
" [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,\n",  
" 81, 240, 253, 253, 119, 25, 0, 0, 0, 0, 0, 0, 0,\n",  
" 0, 0],\n",  
" [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,\n",  
" 0, 45, 186, 253, 253, 150, 27, 0, 0, 0, 0, 0, 0,\n",  
" 0, 0],\n",  
" [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,\n",  
" 0, 0, 16, 93, 252, 253, 187, 0, 0, 0, 0, 0, 0,\n",  
" 0, 0],\n",  
" [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,\n",  
" 0, 0, 0, 0, 249, 253, 249, 64, 0, 0, 0, 0, 0,\n",  
" 0, 0],\n",  
" [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,\n",  
" 0, 46, 130, 183, 253, 253, 207, 2, 0, 0, 0, 0, 0,\n",  
" 0, 0],\n",  
" [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 39,\n",

```

"    148, 229, 253, 253, 253, 250, 182,  0,  0,  0,  0,  0,  0,\n",
"    0,  0],\n",
"  [  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0, 24, 114, 221,\n",
"    253, 253, 253, 253, 201, 78,  0,  0,  0,  0,  0,  0,  0,\n",
"    0,  0],\n",
"  [  0,  0,  0,  0,  0,  0,  0,  0,  0, 23, 66, 213, 253, 253,\n",
"    253, 253, 198, 81,  2,  0,  0,  0,  0,  0,  0,  0,  0,\n",
"    0,  0],\n",
"  [  0,  0,  0,  0,  0,  0, 18, 171, 219, 253, 253, 253, 253,\n",
"    195, 80,  9,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,\n",
"    0,  0],\n",
"  [  0,  0,  0,  0, 55, 172, 226, 253, 253, 253, 253, 244, 133,\n",
"    11,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,\n",
"    0,  0],\n",
"  [  0,  0,  0,  0, 136, 253, 253, 253, 212, 135, 132, 16,  0,\n",
"    0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,\n",
"    0,  0],\n",
"  [  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,\n",
"    0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,\n",
"    0,  0],\n",
"  [  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,\n",
"    0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,\n",
"    0,  0],\n",
"  [  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,\n",
"    0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,\n",
"    0,  0]], dtype=uint8)"
]
},
"execution_count": 4,
"metadata": {},
"output_type": "execute_result"

```

```
}  
],  
"source": [  
  "X_train[0]"  
]  
},  
{  
  "cell_type": "code",  
  "execution_count": 5,  
  "id": "b6957877",  
  "metadata": {},  
  "outputs": [  
    {  
      "data": {  
        "text/plain": [  
          "5"  
        ]  
      },  
      "execution_count": 5,  
      "metadata": {},  
      "output_type": "execute_result"  
    }  
  ],  
  "source": [  
    "y_train[0]"  
  ]  
},  
{  
  "cell_type": "code",  
  "execution_count": 6,  
  "id": "dfa5630f",
```



```
"metadata": {},
"outputs": [
  {
    "data": {
      "text/plain": [
        "<matplotlib.image.AxesImage at 0x2150688afd0>"
      ]
    },
    "execution_count": 6,
    "metadata": {},
    "output_type": "execute_result"
  },
  {
    "data": {
      "image/png": "
      "text/plain": [
        "<Figure size 640x480 with 1 Axes>"
      ]
    },
    "metadata": {},
    "output_type": "display_data"
  }
],
"source": [
  "plt.imshow(X_train[0])"
],
"cell_type": "code",
"execution_count": 7,
"id": "661e7f60",
```

```

"metadata": {},
"outputs": [],
"source": [
    "X_train = X_train.reshape(60000, 28, 28, 1).astype('float32')\n",
    "X_test = X_test.reshape(10000, 28, 28, 1).astype('float32')\n"
]
},
{
    "cell_type": "code",
    "execution_count": 8,
    "id": "116f1613",
    "metadata": {},
    "outputs": [],
    "source": [
        "number_of_classes = 10\n",
        "Y_train = np_utils.to_categorical(y_train, number_of_classes)\n",
        "Y_test = np_utils.to_categorical(y_test, number_of_classes)\n"
    ]
},
{
    "cell_type": "code",
    "execution_count": 9,
    "id": "e7f511b4",
    "metadata": {},
    "outputs": [
        {
            "data": {
                "text/plain": [
                    "array([0., 0., 0., 0., 0., 1., 0., 0., 0., 0.], dtype=float32)"
                ]
            }
        ]
    },

```

```

    "execution_count": 9,
    "metadata": { },
    "output_type": "execute_result"
  }
],
"source": [
  "Y_train[0]"
],
},
{
  "cell_type": "code",
  "execution_count": 10,
  "id": "42543896",
  "metadata": { },
  "outputs": [],
  "source": [
    "model = Sequential()\n",
    "model.add(Conv2D(64, (3, 3), input_shape=(28, 28, 1), activation=\"relu\"))\n",
    "model.add(Conv2D(32, (3, 3), activation=\"relu\"))\n",
    "model.add(Flatten())\n",
    "model.add(Dense(number_of_classes, activation=\"softmax\"))"
  ]
},
{
  "cell_type": "code",
  "execution_count": 11,
  "id": "dc220044",
  "metadata": { },
  "outputs": [],
  "source": [
    "model.compile(loss='categorical_crossentropy', optimizer=\"Adam\", metrics=[\"accuracy\"])"
  ]
}

```

```

]
},
{
  "cell_type": "code",
  "execution_count": 12,
  "id": "2dcf8aa8",
  "metadata": {},
  "outputs": [
    {
      "name": "stdout",
      "output_type": "stream",
      "text": [
        "Epoch 1/5\n",
        "1875/1875 [=====] - 142s 75ms/step - loss: 0.2317 - accuracy: 0.9532 - val_loss: 0.1107 - val_accuracy: 0.9688\n",
        "Epoch 2/5\n",
        "1875/1875 [=====] - 151s 81ms/step - loss: 0.0680 - accuracy: 0.9798 - val_loss: 0.0706 - val_accuracy: 0.9813\n",
        "Epoch 3/5\n",
        "1875/1875 [=====] - 152s 81ms/step - loss: 0.0481 - accuracy: 0.9849 - val_loss: 0.0873 - val_accuracy: 0.9776\n",
        "Epoch 4/5\n",
        "1875/1875 [=====] - 148s 79ms/step - loss: 0.0388 - accuracy: 0.9880 - val_loss: 0.0985 - val_accuracy: 0.9752\n",
        "Epoch 5/5\n",
        "1875/1875 [=====] - 147s 79ms/step - loss: 0.0309 - accuracy: 0.9898 - val_loss: 0.0913 - val_accuracy: 0.9784\n"
      ]
    }
  ],
  "data": {
    "text/plain": [
      "<keras.callbacks.History at 0x21505e0e8b0>"
    ]
  }
}

```

```

    ]
  },
  "execution_count": 12,
  "metadata": {},
  "output_type": "execute_result"
}
],
"source": [
  "\n",
  "model.fit(X_train, Y_train, batch_size=32, epochs=5, validation_data=(X_test,Y_test))"
]
},
{
  "cell_type": "code",
  "execution_count": 13,
  "id": "7a586305",
  "metadata": {},
  "outputs": [
    {
      "name": "stdout",
      "output_type": "stream",
      "text": [
        "Metrics (Test Loss & Test Accuracy): \n",
        "[0.09134303778409958, 0.9783999919891357]\n"
      ]
    }
  ],
  "source": [
    "\n",
    "metrics = model.evaluate(X_test, Y_test, verbose=0)\n",
    "print(\"Metrics (Test Loss & Test Accuracy): \")\n"
  ]
}

```

```

"print(metrics)"
]
},
{
  "cell_type": "code",
  "execution_count": 14,
  "id": "dcd0388f",
  "metadata": {},
  "outputs": [
    {
      "name": "stdout",
      "output_type": "stream",
      "text": [
        "1/1 [=====] - 0s 119ms/step\n",
        "[[6.25965448e-16 1.25216477e-20 1.75363088e-13 5.50589695e-12\n",
        " 4.62221321e-20 6.49635625e-17 1.15485593e-20 1.00000000e+00\n",
        " 4.76414740e-16 2.77069153e-14]\n",
        "[2.25897328e-08 1.58694990e-09 9.99999762e-01 9.70034941e-10\n",
        " 6.94937509e-15 5.10383218e-18 2.60728310e-08 2.12245768e-15\n",
        " 2.35813729e-07 2.87017610e-17]\n",
        "[1.98433185e-07 9.99914408e-01 1.51130564e-07 6.95965008e-10\n",
        " 5.53215577e-05 1.78290074e-11 2.69697098e-10 1.13093584e-07\n",
        " 2.97780425e-05 1.28232136e-09]\n",
        "[9.99998212e-01 2.94243232e-16 3.05269632e-09 3.60318481e-12\n",
        " 1.85465250e-14 1.47418258e-11 5.10268938e-09 2.74148109e-13\n",
        " 7.40127681e-10 1.79003518e-06]]\n"
      ]
    }
  ],
  "source": [
    "prediction = model.predict(X_test[:4])\n",

```

```
"print(prediction)"
]
},
{
  "cell_type": "code",
  "execution_count": 15,
  "id": "2ae4edc1",
  "metadata": {},
  "outputs": [
    {
      "name": "stdout",
      "output_type": "stream",
      "text": [
        "[7 2 1 0]\n",
        "[[0. 0. 0. 0. 0. 0. 0. 1. 0. 0.]\n",
        "[0. 0. 1. 0. 0. 0. 0. 0. 0. 0.]\n",
        "[0. 1. 0. 0. 0. 0. 0. 0. 0. 0.]\n",
        "[1. 0. 0. 0. 0. 0. 0. 0. 0. 0.]]\n"
      ]
    }
  ],
  "source": [
    "print(numpy.argmax(prediction, axis=1))\n",
    "print(Y_test[:4])"
  ]
},
{
  "cell_type": "code",
  "execution_count": null,
  "id": "a475f89a",
  "metadata": {},
```

```
"outputs": [],
"source": []
}
],
"metadata": {
  "kernel_spec": {
    "display_name": "Python 3 (ipykernel)",
    "language": "python",
    "name": "python3"
  },
  "language_info": {
    "codemirror_mode": {
      "name": "ipython",
      "version": 3
    },
    "file_extension": ".py",
    "mimetype": "text/x-python",
    "name": "python",
    "nbconvert_exporter": "python",
    "pygments_lexer": "ipython3",
    "version": "3.9.13"
  }
},
"nbformat": 4,
"nbformat_minor": 5
}
```



## CHAPTER 8

### TESTING

Test case ID	Feature Type	Component	Test Scenario	Expected Result	Actual Result	Status
HP_TC_001	UI	Home Page	Verify UI elements in the Home Page	The Home page must be displayed properly	Working as expected	PASS
HP_TC_002	UI	Home Page	Check if the UI elements are displayed properly in different screen sizes	The Home page must be displayed properly in all sizes	The UI is not displayed properly in screen size 2560 x 1801 and 768 x 630	FAIL
HP_TC_003	Functional	Home Page	Check if user can upload their file	The input image should be uploaded to the application successfully	Working as expected	PASS
HP_TC_004	Functional	Home Page	Check if user cannot upload unsupported files	The application should not allow user to select a non image file	User is able to upload any file	FAIL
HP_TC_005	Functional	Home Page	Check if the page redirects to the result page once the input is given	The page should redirect to the results page	Working as expected	PASS

BE_TC_001	Functional	Backend	Check if all the routes are working properly	All the routes should properly work	Working as expected	PASS
M_TC_001	Functional	Model	Check if the model can handle various image sizes	The model should rescale the image and predict the results	Working as expected	PASS
M_TC_002	Functional	Model	Check if the model predicts the digit	The model should predict the number	Working as expected	PASS
M_TC_003	Functional	Model	Check if the model can handle complex input image	The model should predict the number in the complex image	The model fails to identify the digit since the model is not built to handle such data	FAIL
RP_TC_001	UI	Result Page	Verify UI elements in the Result Page	The Result page must be displayed properly	Working as expected	PASS
RP_TC_002	UI	Result Page	Check if the input image is displayed properly	The input image should be displayed properly	The size of the input image exceeds the display container	FAIL
RP_TC_003	UI	Result Page	Check if the result is displayed properly	The result should be displayed properly	Working as expected	PASS
RP_TC_004	UI	Result Page	Check if the other predictions are displayed properly	The other predictions should be displayed properly	Working as expected	PASS

## 8.2 USER ACCEPTANCE TESTING

### 8.2.1 DEFECT TESTING

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

### 8.2.2 TEST CASE ANALYSIS

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

### 9.1 PERFORMANCE METRICS

#### Locust Test Report

During: 11/12/2022, 7:05:40 AM - 11/12/2022, 7:14:47 AM

Target Host: http://127.0.0.1:5000/

Script: locust.py

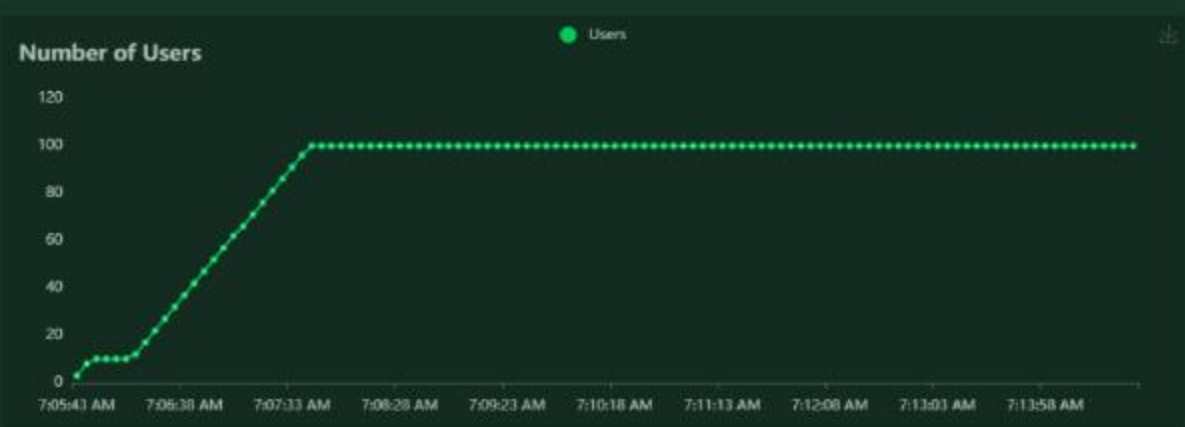
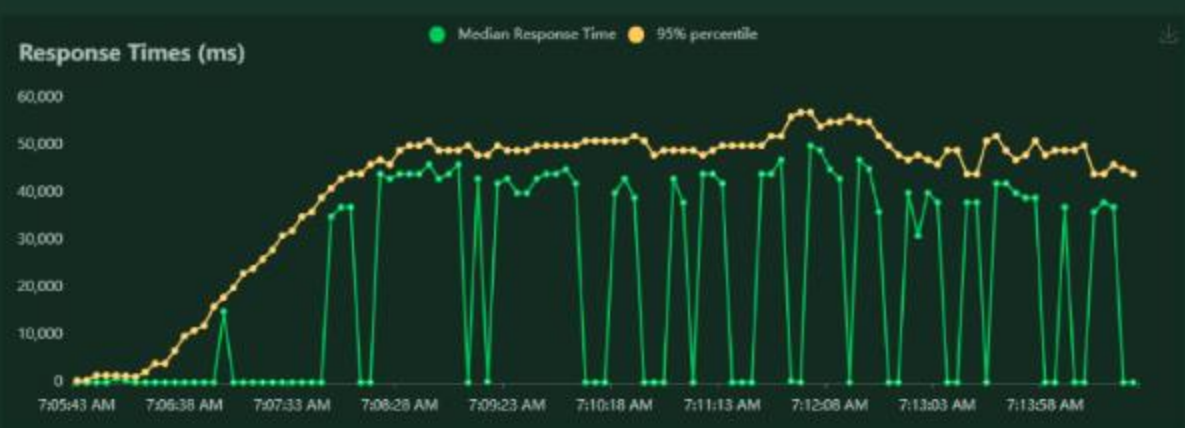
#### Request Statistics

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

#### Response Time Statistics

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

Charts



## **CHAPTER 10**

### **ADVANTAGES & 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 machine 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. During testing, the model achieved a 99.61% recognition rate. The proposed project is scalable and can easily handle a huge number of users. Since it is a web application, it is compatible with any device that can run a browser. This project is extremely useful in real-world scenarios such as recognizing number plates of vehicles, processing bank cheque amounts, numeric entries in forms filled up by hand (tax forms) and so on. There is so much room for improvement, which can be implemented in subsequent versions.

## **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.



# APPENDIX

## SOURCE CODE

```
import numpy as np
import os
from PIL import Image
from flask import Flask, request, render_template, url_for
from werkzeug.utils import secure_filename, redirect
#from gevent.pywsgi import WSGIServer
from keras.models import load_model
from keras.preprocessing import image
from flask import send_from_directory

UPLOAD_FOLDER = r'C:\Users\nithy\handwirtten\data'

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

model = load_model(r"C:\Users\nithy\mnistCNN.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(debug=True, threaded=False)
```

## STYLE.CSS CODING

```
#clear_button{
  margin-left: 15px;
  font-weight: bold;
  color: rgb(0, 174, 255);
}

#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;
  position: relative;
  color: rgb(0, 32, 112);
  background-color: skyblue;
  padding-top: 1%;
  padding-bottom: 1%;
  font-weight: bold;
  font-family: 'Bookman', 'URW Bookman L', serif;
}

#team_id{
  text-align: right;
  font-size: 25px;
  padding-right: 3%;
}

#predict_button{
  margin-right: 15px;
```

```
    color: rgb(0, 255, 72);
    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; */
    background: #eee;
    border: 1px solid #888;
    margin-top: 20px;
    margin-bottom: 20px;
}

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

.heading{
    font-family: "American Typewriter", serif;
```

```

    font-weight: 700;
    font-size: 2rem;
    display: inline;
}

.leftside{
    text-align: center;
    margin: 0 auto;
    margin-top: 2%;
    /* padding-left: 10%; */
}

#frame{
    margin-right: 10%;
}

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

h1{
    text-align: center;
    color: aliceblue;
    padding: 100px 50px 65px 100px;
}

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

```

## INDEX.HTML

<html>

<head>

<title>A Novel Method For Handwritten Digit Recognition</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&display=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.css"
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://cdn.jsdelivr.net/npm/popper.js/1.14.7/umd/popper.min.js"
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"></script>
```

```
<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.min.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><center>NAALAIYA THIRAN<br>
  IBM Guided project</center></h1>
  <div class="container p-3 my-3 bg-cyan text-white">
    <p style = "text-align:center"> Handwritten Digit Recognition System</p>
      <p style = "text-align:center"> Team ID - PNT2022TMID41196</P>
    <p style = "text-align:center"> Team batch Members - Register Number</P>
    <p style = "text-align:center"> 612819106023    MUNIVEL M</P>
    <p style = "text-align:center"> 612819106010    GOKUL SURYA G</P>
    <p style = "text-align:center"> 612819106031    PREETHI G</P>
    <p style = "text-align:center"> 612819106047    SRIPRIYA G</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">&nbsp; Clear &nbsp;</button>
        </div>
      </form>
    </div>
  </section>

</body>

</html>

```

## PERDICT.HTML

```

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

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

```

```

#rectangle{
width:400px;
height:150px;
background-color: #000000;
border-radius: 15px;
position:absolute;
box-shadow: 0px 0px 10px 5px white;
top:25%;
left:50%;
transform:translate(-50%,-50%);
}

#head{
text-align: center;
font-size: 30px;
margin: 0 auto;
padding: 3% 5%;
font-family: Arial, Helvetica, sans-serif;
color: white;
}

#num{
font-size: 50px;
}

</style>

<body>

<div id="rectangle">
<h1 id="head">Predicted Number : <br><center id="num">{ { num} }</center></h1>
</div>

</body>
</html>

```

## GITHUB LINK

<https://github.com/IBM-EPBL/IBM-Project-45334-1660729516.git>

## PROJECT DEMO LINK

<https://github.com/IBM-EPBL/IBM-Project-45334-1660729516/blob/3452f3c9a3cb2a80552931e6ae1b292adca63bb2/Final%20Deliverables/Demo%20video%20link/Project%20Demo%20link.mp4>