# A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITON

**A PROJECT REPORT**

***Submitted by***


**NIRANJANA [19CS095]**

**GOWTHAM [19CS510]**

**KAVI RUBAN [19CS061]**

**KARTHIKSHARAN [19CS057]**


*in partial fulfilment for the award of the degree*

*of*

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**


**SNS COLLEGE OF TECHNOLOGY, SARAVANAMPATTI**

# ABSTRACT

Deep learning has recently taken a radical turn in the field of machine learning by making it more artificially intelligent, thanks to the advent of Convolutional Neural Networks (CNN). Because of its wide range of applications, deep learning is used in a wide range of industries, including surveillance, health, medicine, sports, robots, and drones. Handwritten Digit Recognition is an example of a computer's capacity to recognise human handwritten digits. Because handwritten numerals aren't flawless and might be generated with a variety of tastes, it's difficult work for the machine. The purpose of this project is to provide a response to a current problem that uses a digit image and recognises the digit contained in the image using the Convolutional Neural Networks idea. The Modified National Institute of Standards and Technology (MNIST) dataset is used to train our model in this research. This dataset was created using the convolutional neural network technique and Keras, a Python library for intensive computation of neural nodes that is supported by the Tensor Flow framework on the backend. We will be able to estimate the handwritten digits in an image using this model. This approach allows us to detect numerous digits.

To guarantee the dataset is free from any redundant orirrelevant variables to the target variable, data implementing the data

classification mission. Thirdly, to decrease the errors of training and validation, and avoid the limitation of datasets, data augmentation has been proposed. Fourthly, to simulate the real-world natural influences that can affect image quality, we propose to add an additive white Gaussian noise with $\sigma = 0.5$ to the MNIST dataset. As a result, our CNN algorithm achieves state-of-the-art results in handwritten digit recognition, with a recognition accuracy of 99.98%, and 99.40% with 50% noise

# TABLE OF CONTENTS


| CHAPTER NO. | TITLE | PAGE NO. |
|---|---|---|
| | **ABSTRACT** | **i** |
| | **LIST OF TABLES** | **ii** |
| | **LIST OF FIGURES** | **iv** |
| | **LIST OF ABBREVIATIONS** | **v** |
| **1.** | **INTRODUCTION** | **1** |
| | 1.1 PROJECT OVERVIEW | **1** |
| | 1.2 PURPOSE | **1** |
| **2.** | **LITERATURE SURVEY** | **2** |
| | 2.1 EXISTING PROBLEM | **2** |
| | 2.2 REFERENCES | **2** |
| | 2.3 PROBLEM STATEMENT DEFINITION | **4** |
| **3.** | **IDEATION AND PROPOSED SOLUTION** | **5** |
| | 3.1 EMPATHY MAP CANVAS | **5** |
| | 3.2 IDEATION AND BRAINSTORMING | **6** |
| | 3.3 PROPOSED SOLUTION | **10** |
| | 3.4 PROBLEM SOLUTION FIT | **11** |
| **4.** | **REQUIREMENT ANALYSIS** | **12** |
| | 4.1 FUNCTIONAL REQUIREMENTS | **12** |
| | 4.2 NON-FUNCTIONAL REQUIREMENTS | **13** |
| **5.** | **PROJECT DESIGN** | **14** |
| | 5.1 DATA FLOW DIAGRAM | **14** |
| | 5.2 SOLUTION & TECHNICAL ARCHITECTURE | **16** |
| | 5.3 USER STORIES | **17** |
| **6.** | **PROJECT PLANNING AND SCHEDULING** | **18** |
| | 6.1 SPRINT PLANNING AND ESTIMATION | **18** |
| | 6.2 SPRINT DELIVERY SCHEDULE | **18** |
| **7.** | **CODING AND SOLUTIONING** | **19** |
| | 7.1 FEATURE 1 | **19** |
| | 7.2 FEATURE 2 | **19** |

| | | |
|---|---|---|
| **8.** | **TESTING** | **20** |
| | 8.1 TEST CASES | **20** |
| | 8.2 USER ACCEPTANCE TESTING | **21** |
| | 8.2.1 Defect Analysis | **21** |
| | 8.2.2 Test Case Analysis | **21** |
| **9.** | **RESULTS** | **22** |
| | 9.1 PERFORMANCE METRICS | **22** |
| **10.** | **ADVANTAGES AND DISADVANTAGES** | **23** |
| **11.** | **CONCLUSION** | **24** |
| **12.** | **FUTURE SCOPE** | **25** |
| **13.** | **APPENDIX** | **26** |
| | 13.1 SOURCE CODE | **26** |
| | 13.2 GITHUB AND PROJECT DEMO | **30** |
| **14.** | **REFERENCES** | **31** |

# LIST OF FIGURES


| FIG NO. | FIGURE NAME | PAGE NO. |
|---|---|---|
| 5.1 | Data Flow Diagram | 14 |
| 5.1.1 | DFD Level 0 | 15 |
| 5.1.2 | DFD Level 1 | 15 |
| 5.1.3 | DFD Level 2 | 16 |

# LIST OF ABBREVIATIONS

| | |
|---|---|
| NN | Neural Network |
| CNN | Convolutional Neural Network |
| MNIST | Modified National Institute of Standard and Technology |
| HTML | Hypertext Markup Language |
| CSS | Cascading Style Sheet |

# CHAPTER 1

# INTRODUCTION

## 1.1 PROJECT OVERVIEW

Machine learning (ML) is a type of artificial intelligence (AI) that allows software applications to become more accurate at predicting outcomes without being explicitly programmed to do so. Machine learning algorithms use historical data as input to predict new output values. Recommendation engines are a common use case for machine learning. Other popular uses include fraud detection, spam filtering, malware threat detection,business process automation (BPA) and Predictive maintenance.

The handwritten digit recognition is the ability of computers to recognize human handwritten digits from various sources, such as images, documents, and so on. It is a hard task for the machine because handwritten digits are not perfect and can be made with many different flavours. The handwritten digit recognition is the solution to this problem which uses the image of a digit and recognizes the digit present in the image.

## 1.2 PURPOSE

Handwritten digit recognition is the process to provide the ability to machines to recognize human handwritten digits. It is not an easy task for the machine because handwritten digits are not perfect, vary from person-to-person, and can be made with many different flavours.

Digit recognition systems are capable of recognizing the digits from different sources like emails, bank cheque, papers, images, etc. 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 2

# LITERATURE SURVEY

## 2.1 EXISTING PROBLEM

The handwritten digits are not always of the same size, width, orientation and justified to margins as they differ from writing of person to person, so the general problem would be while classifying the digits due to the similarity between digits such as 1 and 7, 5 and 6, 3 and 8, 2 and 5, 2 and 7, etc. This problem is faced more when many people write a single digit with a variety of different handwritings. Lastly, the uniqueness and variety in the handwriting of different individuals also influence the formation and appearance of the digits.

## 2.2 REFERENCES

**SnapSolve — A Novel Mathematics Equation Solver using Deep Learning**

*Priyank Shah, Nitiket Shinde, Deep Limbad, Ashwini Save*

Poor resolution of the image will lead to inaccurate text recognition. With the increase in advancements, researchers found deep learning models are providing better efficiency than machine learning models. Deep Learning is providing better accuracy for image segmentation, object detection, speech recognition, and image classification as well so,in this paper implemented CNN model to solve handwritten mathematical equation. This system is solving offline handwritten polynomial equations using Deep Learning CNN model.

**A Novel Learning Algorithm to Optimize Deep Neural Networks: Evolved Gradient Direction Optimizer (EVGO)**

*Ibrahim Karabayir, Oguz Akbilgic, Nihat Tas*

In this article, they propose a novel algorithm, evolved gradient direction optimizer (EVGO), updating the weights of DNNs based on the first-order gradient and a novel hyperplane they introduce. They compare the EVGO algorithm with other gradient- based algorithms, such as gradient descent, RMSProp, Adagrad, momentum, and Adam on the well-known Modified National Institute of Standards and Technology (MNIST) data set

for handwritten digit recognition by implementing deep convolutional neural networks. Furthermore, they present empirical evaluations of EVGO on the CIFAR-10 and CIFAR-100 data sets by using the well-known AlexNet and ResNet architectures.

Finally, they implement an empirical analysis for EVGO and other algorithms to investigate the behavior of the loss functions. The results show that EVGO outperforms all the algorithms in comparison for all experiments. We conclude that EVGO can be used effectively in the optimization of DNNs, and also, the proposed hyperplane may provide a basis for future optimization algorithms.

**Electro – Optical Neural Networks Based on Time – Stretch Method**

*Yubin Zang, Minghua Chen, Sigang Yang, Hongwei Chen*

This paper presents a novel method developed for pre-processing of the triaxial accelerometer output signals recorded during a handwriting. The DWT (Discrete Wavelet Transform) was applied to decompose acceleration of each axis into two components representing the acceleration caused by intended hand motion and the fluctuation of acceleration, respectively. Based on the first mentioned component various features were extracted to recognize handwriting and to characterize a person.

**Parameterization of the Acceleration Signals Recorded During Handwriting**

*Barbara Wilk, Malgorzata Augustyn*

In this paper, they propose a novel method that is capable of transferring the knowledge between any two layers of two neural networks by matching the similarity between the extracted representations. The proposed method is model-agnostic overcoming several limitations of existing knowledge transfer techniques, since the knowledge is transferred between layers that can have different architecture and no information about the complex model is required, apart from the output of the layers employed for the knowledge transfer. Three image datasets are used to demonstrate the effectiveness of the proposed approach, including a large-scale dataset for learning a light-weight model for facial pose

estimation that can be directly deployed on devices with limited computational resources, such as embedded systems for drones.

**Neural Network Knowledge Transfer using Unsupervised Similarity Matching**

*Nikolaos Passalis, Anastasios Tefas*

In this paper, a novel architecture of electro-optical neural networks based on the time-stretch method is proposed and numerically simulated. By stretching time-domain ultrashort pulses, multiplications of large scale weight matrices and vectors can be implemented on light and multiple-layer of feedforward neural network operations can be easily implemented with fiber loops. Via simulation, the performance of a three-layer electro-optical neural network is tested by the handwriting digit recognition task and the accuracy reaches 88% under considerable noise.

## 2.3 PROBLEM STATEMENT DEFINITON

Each person has their own way of handwriting, each handwriting changes with width, orientation, height, etc. It is very difficult for a human to recognize handwritten digits more accurately and quickly. For instance, considering a postal office they receive many numbers of posts with many postal codes. It is a hectic job for the postman to recognize the postal code. Using the handwritten digit recognition system they can recognize the digits easily, hence the work becomes very easy for them.

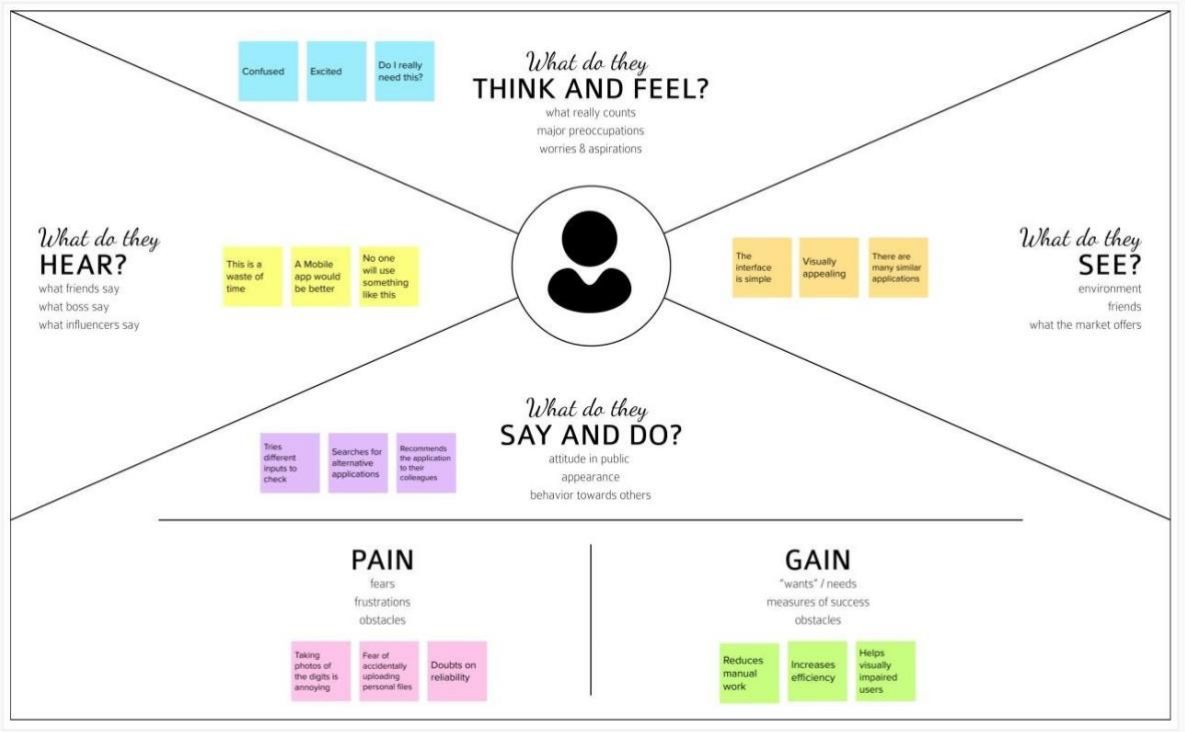
# CHAPTER 3

# IDEATION AND 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.

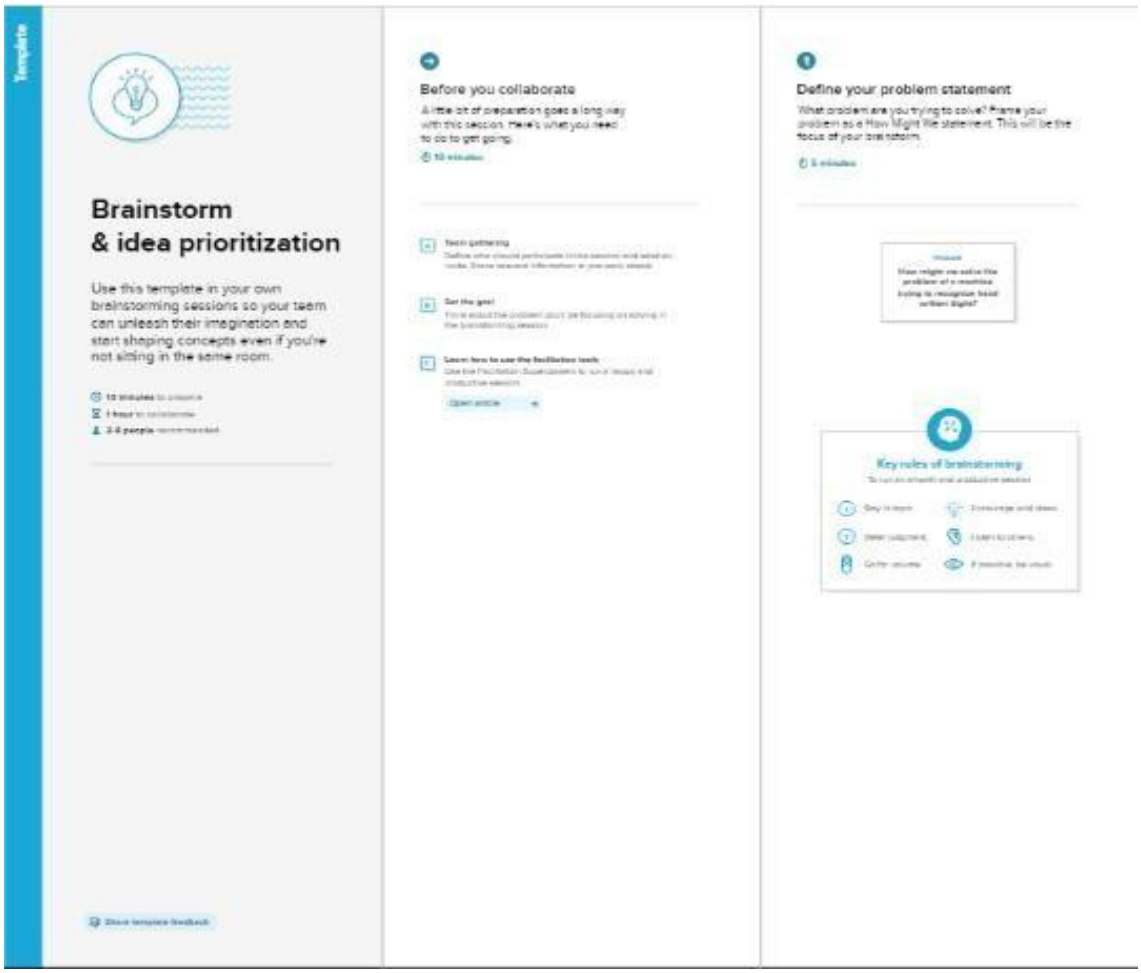
Users think that these types of applications should have a very simple interface and need to be visually appearing for them to use, these can be also given to them as a mobile application for easy access. They feel that the application is very exciting as the digits are recognized but are confused whether they need these types of applications. They start trying to check the application by feeding various inputs get fascinated by the output produced by the application and then start recommending the application to their colleagues. Even though the application reduces the manual work and increase the efficiency of recognizing the digit, users thinks that there are few disadvantages also. Users feel that they may accidentally upload some sensitive files and taking photos of the digits is very annoying.



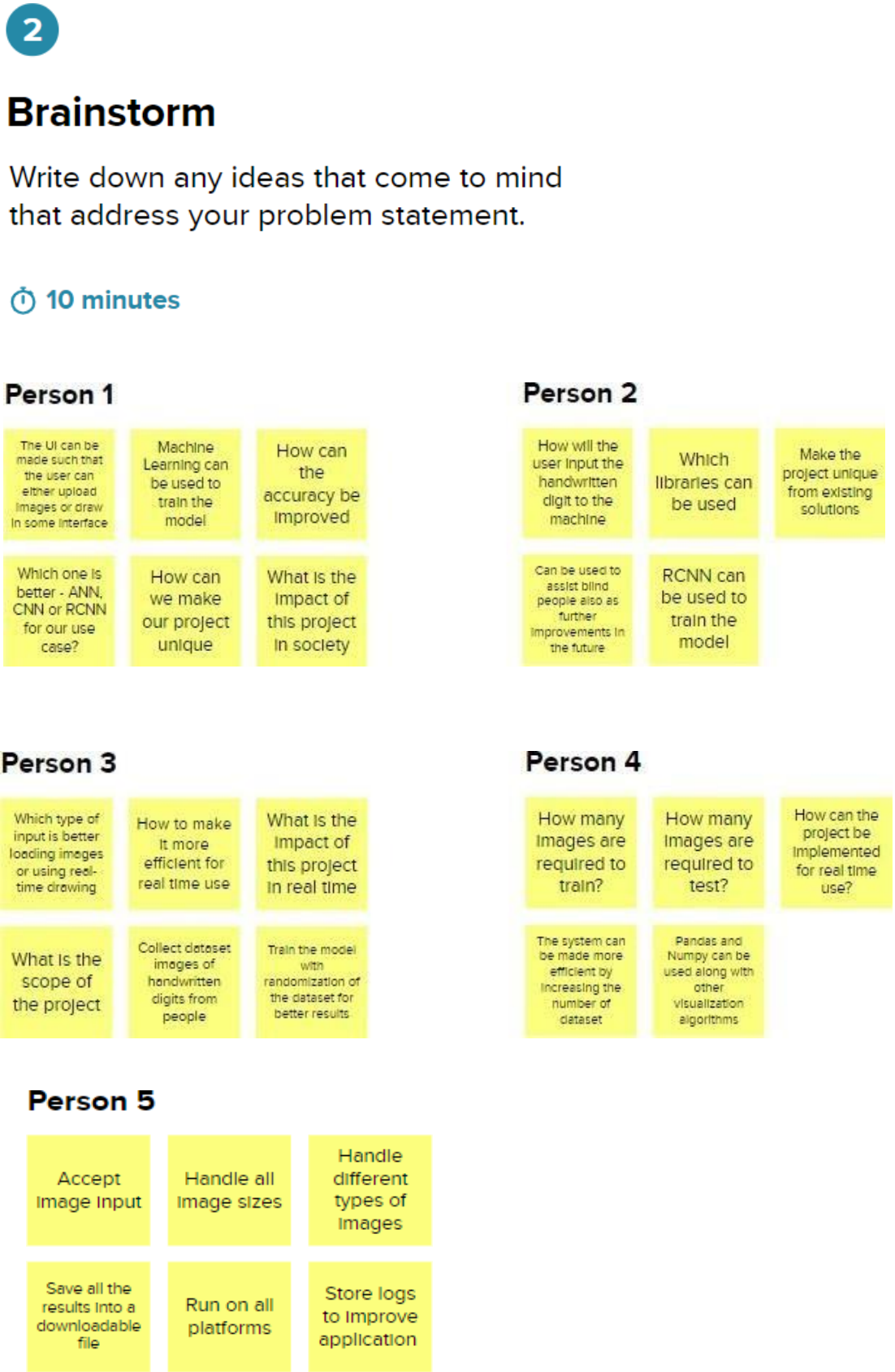
## 3.2 IDEATION AND BRAINSTORMING

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich number of creative solutions.

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



**Step-2: Brainstorm, Idea Listing and Grouping**



3

## Group ideas

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

20 minutes

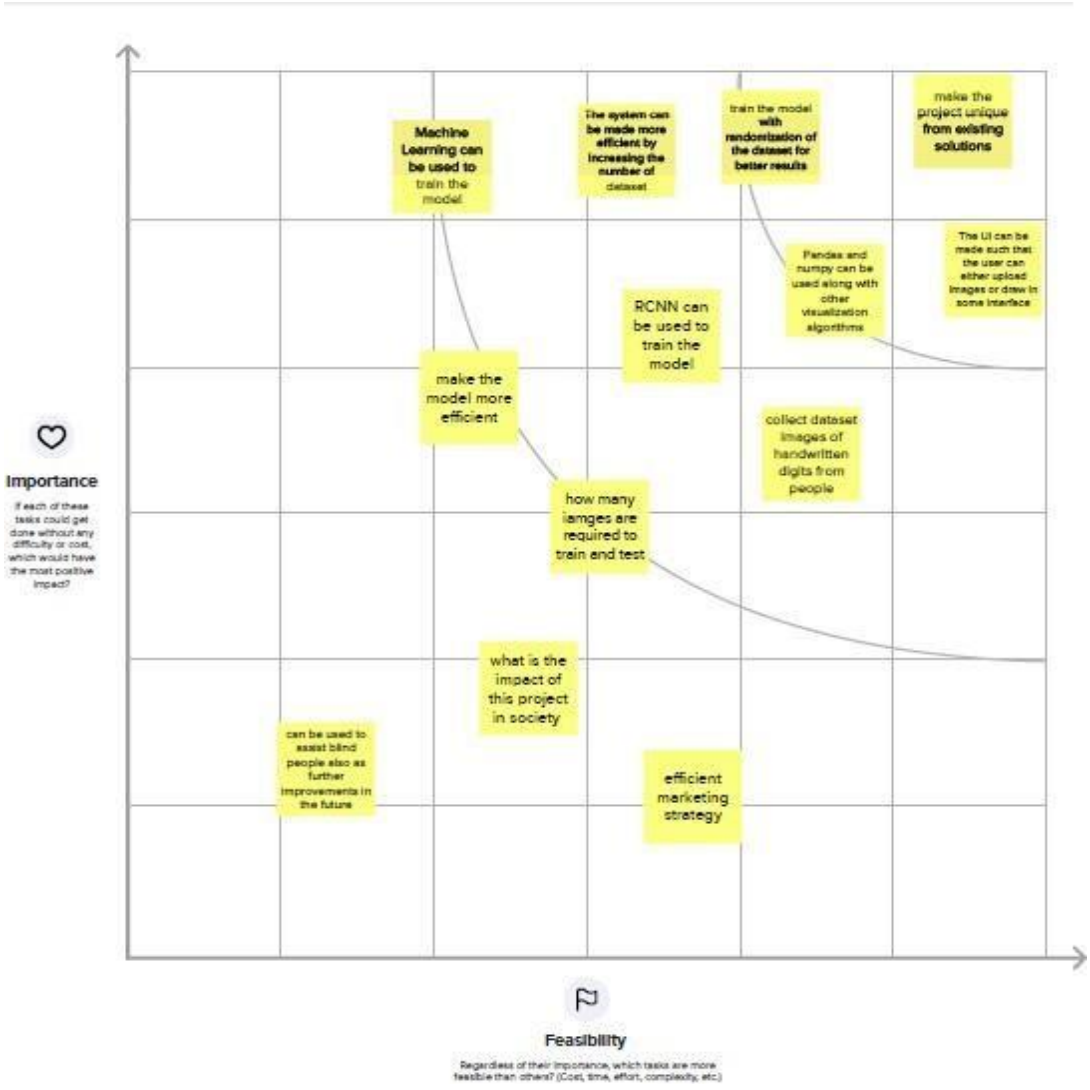
**Step-3: Idea Prioritization**

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

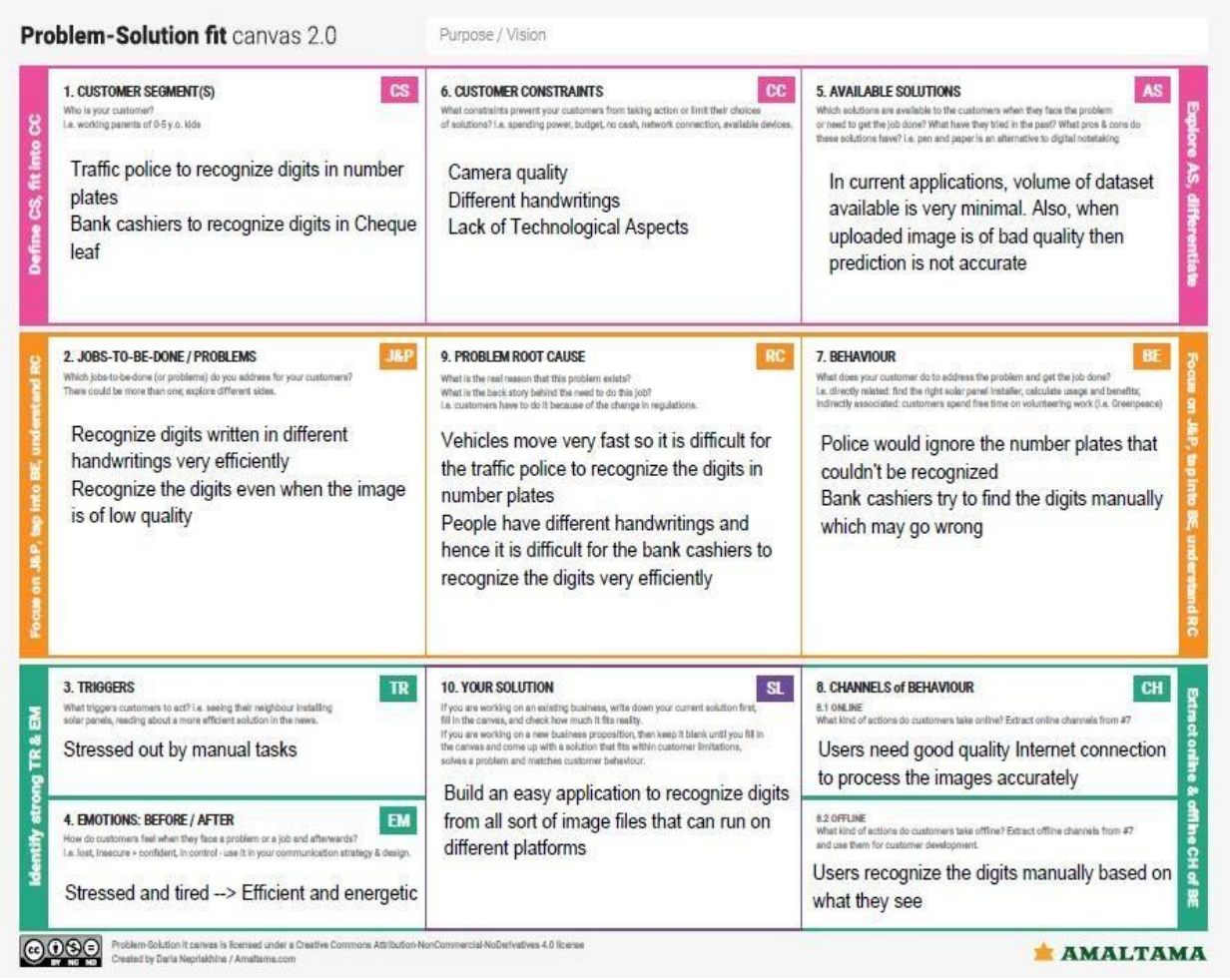
Develop a machine learning model using neural networks and CNN to capture the handwritten digits using the MNIST dataset. This model can recognize various handwritings very efficiently, hence reducing the manual tasks and fastens the prediction process. This model can be used by traffic police to recognize the number plates and bank cashiers to recognize the digits in various cheque leaf. As a future enhancement, the model can be scaled up to various languages recognition.

| S.No. | Parameter | Description |
|---|---|---|
| 1. | Problem Statement (Problem to be solved) | The handwritten digit given by the user should be recognized by the user |
| 2. | Idea / Solution description | To develop a machine learning model using neural networks and CNN to capture the patterns using the dataset |
| 3. | Novelty / Uniqueness | Model developed can recognize different handwritings very efficiently |
| 4. | Social Impact / Customer Satisfaction | Application reduces the manual tasks and fastens the prediction |
| 5. | Business Model (Revenue Model) | This application can be used by the traffic police to recognize the number plates<br><br>This application can also be used by the bank cashiers to recognize the digits in various cheque leaf's |
| 6. | Scalability of the Solution | This application can be used to recognize various language digits and also scaled up to accept multiple inputs to process them |

## 3.4 PROBLEM SOLUTION FIT

The Problem-Solution Fit means that we have found a problem with our customer and that the solution we have realized for it actually solves the customer's problem. It helps entrepreneurs, marketers and corporate innovators identify behavioural patterns and recognize what would work and why. Few purposes of Problem-Solution Fit are:

- It can be used to solve complex problems in a way that fits the state of our customers
- Succeed faster and increase our solution adoption by tapping into existing mediums and channels of behaviour
- Sharpen our communication and marketing strategy with the right triggers and messaging
- Increase touch-points with our company by finding the right problem-behaviour fit and building trust by solving frequent annoyances, or urgent or costly problems
- Understand the existing situation in order to improve it for our target group



# CHAPTER 4

# REQUIREMENT ANALYSIS

## 4.1 FUNCTIONAL REQUIREMENTS

| FR.NO | FUNCTIONAL REQUIREMENTS | SUB REQUIREMENTS |
|---|---|---|
| FR-1 | Model Creation | Get access the MNIST dataset |
| | | Analyse the dataset |
| | | Define a CNN model |
| | | Train and Test the Model |
| FR-2 | Application Development | Create a website to let the user recognize handwritten digits. |
| | | Create a home page to upload images |
| | | Create a result page to display the results |
| | | Host the website to let the users use it from anywhere |
| FR-3 | Input Image Upload | Let users upload images of various formats. |
| | | Let users upload images of various size |
| | | Prevent users from uploading unsupported image formats |
| | | Pre-Process the image to use it on the model |
| | | Create a database to store all the input images |
| FR-4 | Display Results | Display the result from the model |
| | | Display input image |
| | | Display accuracy the result |
| | | Display other possible predictions with their respective accuracy |

## 4.2 NON-FUNCTIONAL REQUIREMENTS

| FR.NO | NON-FUNCTIONAL REQUIREMENTS | DESCRIPTION |
|---|---|---|
| NFR-1 | Usability | The application must be usable in all devices |
| NFR-2 | Security | The application must protect user uploaded image |
| NFR-3 | Reliability | The application must give an accurate result as much as possible |
| NFR-4 | Performance | The application must be fast and quick to load up |
| NFR-5 | Availability | The application must be available to use all the time |
| NFR-6 | Scalability | The application must scale along with the user base |

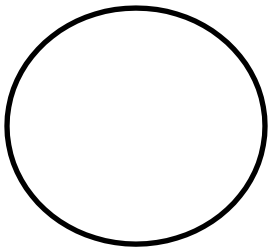
# CHAPTER 5

# PROJECT DESIGN

## 5.1 DATA FLOW DIAGRAM

Data flow diagram is used to describe how the information is processed and stored and identifies how the information flows through the processes. Data flow diagram illustrates how the data is processed by a system in terms of inputs and outputs. The data flow diagram also depicts the flow of the process and it has various levels. The initial level is context level which describes the entire system functionality and the next level describes eachand every sub module in the main system as a separate process or describes all the process involved in the system separately. Data flow diagram are made up of number of symbols,

Square presenting external entities, which are sources or destination of data



Circle representing processes, which take data as input, do something to it and output it

Arrow representing the data flows, which can either be electronic data or physical items

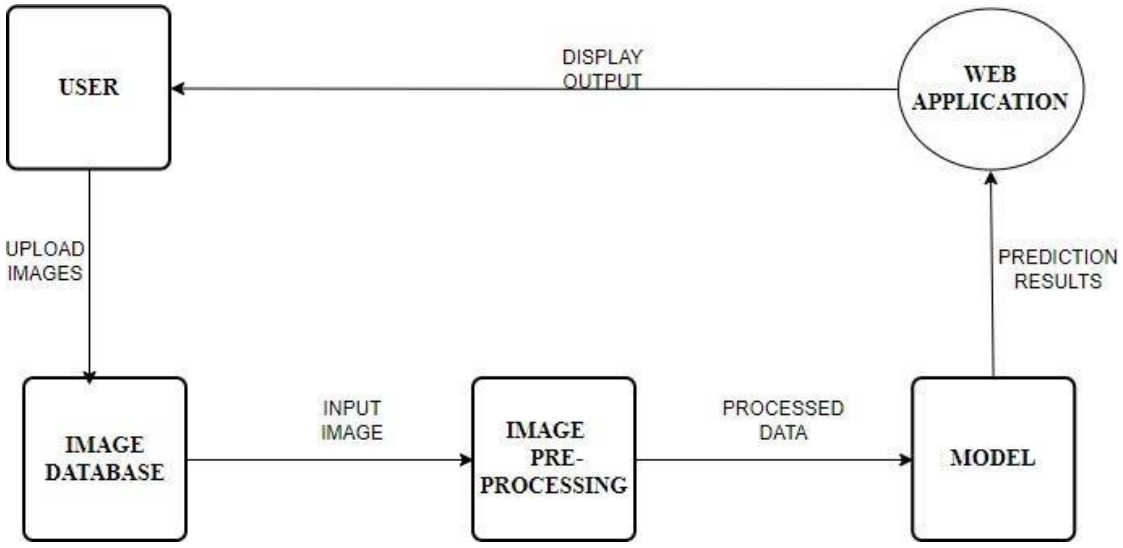
Parallel lines representing data stores, including electronic stores such as databases or XML files and physical stores

### 5.1.1 DFD LEVEL 0

The users of the system upload an image in the web application to recognize the handwritten digits in it. The image is feed into a model for recognition and the answer is sent back to the web application.

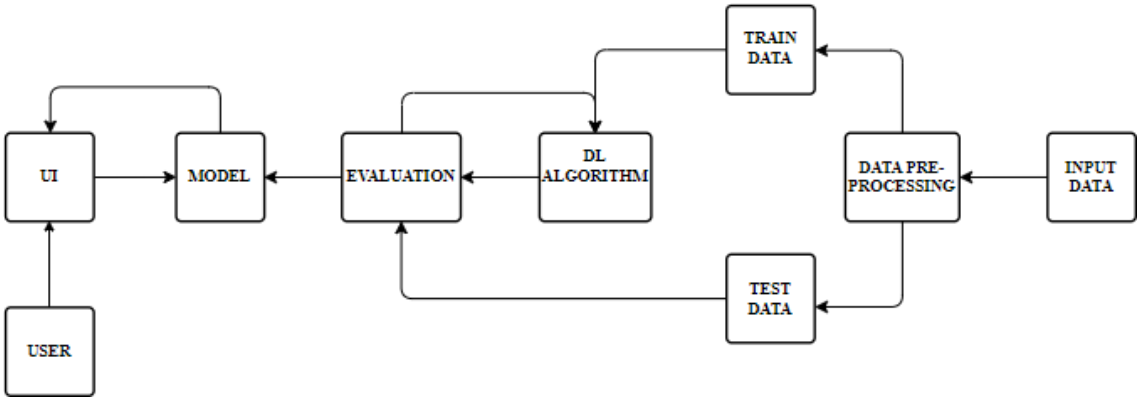

### 5.1.2 DFD LEVEL 1

The image uploaded by the user is initially stored in the image database, then image is pre-processed for recognition. The processed data is sent into the model to predict the result. Finally, the output is displayed in the web application.

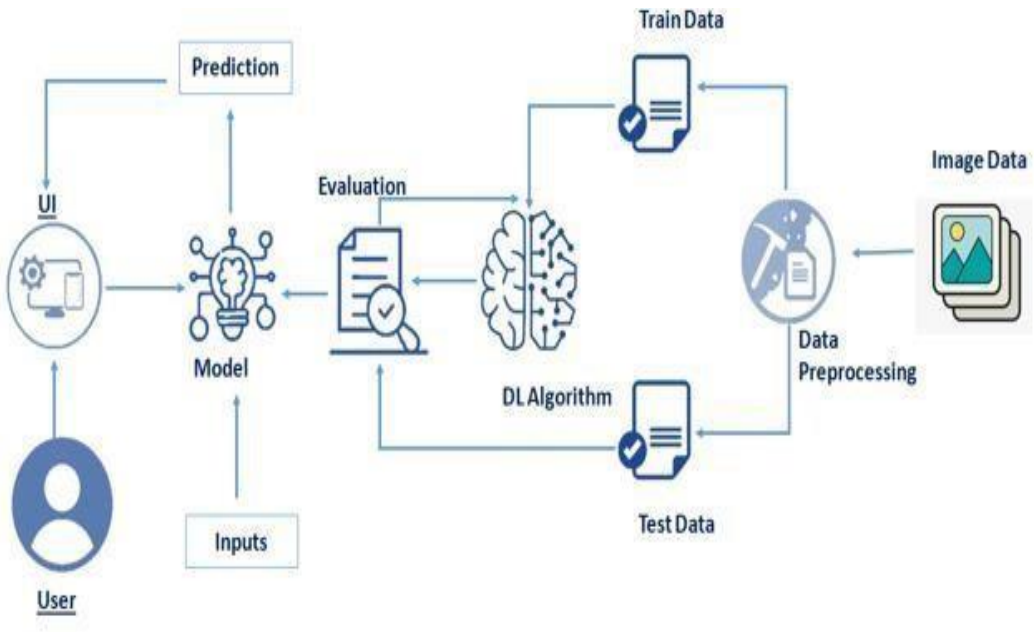


### 5.1.3 DFD LEVEL 2

CNN Model is first trained with the MNIST dataset, then processed image is sent into the model which passes through various layers present in the CNN Model for further processing then the digit is recognized.




## 5.2 SOLUTION AND TECHNICAL ARCHITECTURE



## 5.3 USER STORIES

| USER TYPE | FUNCTIONAL REQUIREMENTS | USER STORY NUMBER | USER STORY/ TASK | ACCEPTA-NCE CRITERIA | PRIO RITY | RELEASE |
|---|---|---|---|---|---|---|
| Customer (Mobile user) | Accessing the Application | USN-1 | As a user, I should be able to access the application from any device a any time | I can access the application using the browser on any device | High | Sprint-4 |
| | Uploading Image | USN-2 | As a user, I should be able to upload images to predict the digits | I can upload images | High | Sprint-3 |
| | Viewing the Results | USN-3 | As a user, I should be able to view the results | Digits are predicted and displayed | High | Sprint-3 |
| | | USN-4 | As a user, I should be able to see other close predictions | The accuracy of other values must be displayed | Medi um | Sprint-4 |
| | Usage Instruction | USN-5 | As a user, I should have a usage instruction to know how to use the application | The usage instruction is displayed on the home page | Medi um | Sprint-4 |

# CHAPTER 6

# PROJECT PLANNING AND SCHEDULING

## 6.1 SPRINT PLANNING AND ESTIMATION

| Sprint | User Story Number | User Story / Task | Story Points | Priority | Team Members |
|---|---|---|---|---|---|
| Sprint - 1 | USN-1 | Get the dataset | 3 | High | Gowtham Niranjana Kavi ruba Karthiksharan |
| | USN-2 | Explore the data | 2 | Medium | |
| | USN-3 | Data Pre-Processing | 3 | High | |
| | USN-4 | Prepare training and testing data | 3 | High | |
| Sprint - 2 | USN-5 | Create the model | 3 | High | Kaviruban gowtham |
| | USN-6 | Train the model | 3 | High | |
| | USN-7 | Test the model | 3 | High | |
| Sprint - 3 | USN-8 | Improve the model | 2 | Medium | Kaviruban Niranjana Gowtham karthishara |
| | USN-9 | Save the model | 3 | High | |
| | USN-10 | Build the Home Page | 3 | High | |
| | USN-11 | Setup a database to store input images | 2 | Medium | |
| Sprint - 4 | USN-12 | Build the results page | 3 | High | Kaviruban Niranjana Gowtham Karthiksharan |
| | USN-13 | Integrate the model with the application | 3 | High | |
| | USN-14 | Test the application | 3 | High | |

## 6.2 SPRINT DELIVERY SCHEDULE

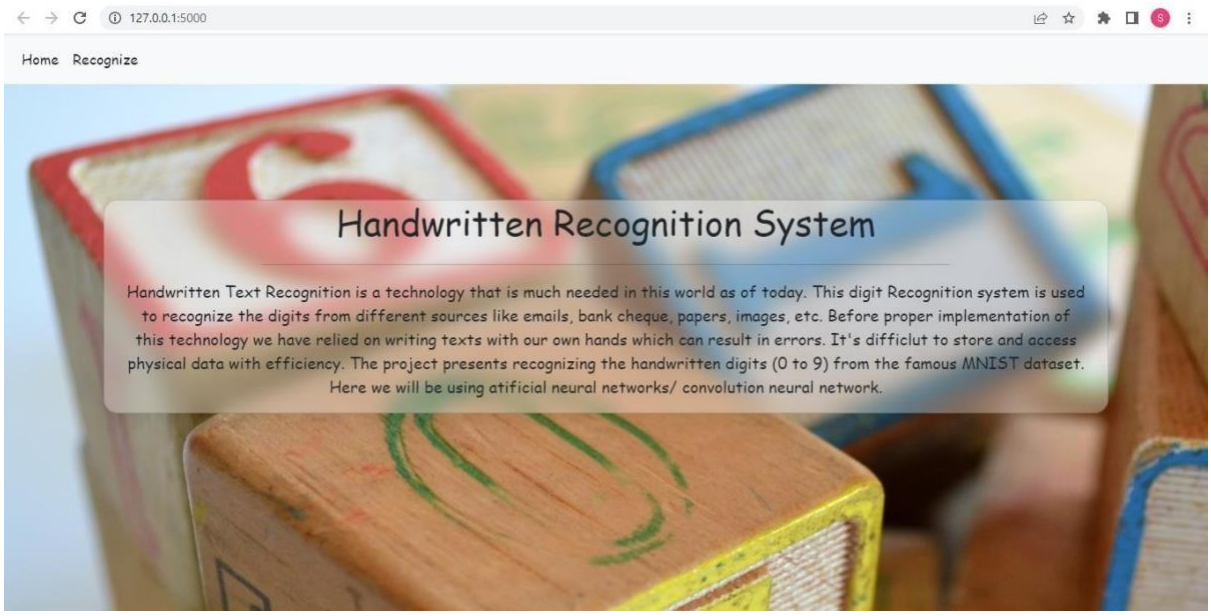
| Sprint | Total Story Points | Duration | Sprint Start Date | Sprint End Date (Planned) | Story Points Completed (as on Planned End Date) | Sprint Release Date (Actual) |
|---|---|---|---|---|---|---|
| | | | | | | |

| | | | | | | |
|---|---|---|---|---|---|---|
| Sprint-1 | 11 | 6 Days | 24 Oct 2022 | 29 Oct 2022 | 11 | 29 Oct 2022 |
| Sprint-2 | 9 | 6 Days | 31 Oct 2022 | 05 Nov 2022 | 9 | 05 Nov 2022 |
| Sprint-3 | 10 | 6 Days | 07 Nov 2022 | 12 Nov 2022 | 10 | 12 Nov 2022 |
| Sprint-4 | 9 | 6 Days | 14 Nov 2022 | 19 Nov 2022 | 9 | 19 Nov 2022 |

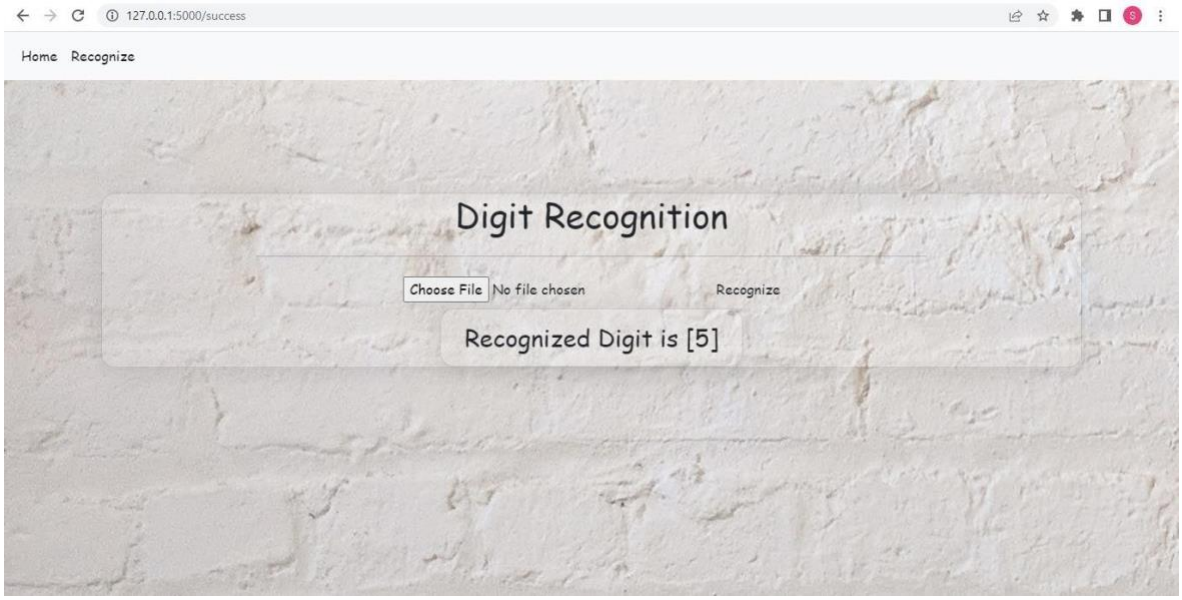
# CHAPTER 7

# CODING AND SOLUTIONING

## 7.1 FEATURE 1




## 7.2 FEATURE 2



# CHAPTER 8

# TESTING

## 8.1 TEST CASES

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

| | | | | | | |
|---|---|---|---|---|---|---|
| TC_007 | Functional | Model | Check if the model can handle complex | The model should predict the number in the | The model fails to identify the | pass |

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

## 8.2 USER ACCEPTANCE TESTING

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

### 8.2.1 TEST CASE ANALYSIS

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

# CHAPTER 9

# RESULTS

## 9.1 PERFORMANCE METRICS

Script: locust.py

### Request Statistics

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

### Response Time Statistics

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

# CHAPTER 10

# ADVANTAGES AND DISADVANTAGES

## 10.1 ADVANTAGES

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

## 10.2 DISADVANTAGES

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

# CHAPTER 11

# CONCLUSION

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

# CHAPTER 12

# FUTURE SCOPE

In the future, application can be improved with following features:

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

# CHAPTER 13

# APPENDIX

## 13.1 SOURCE CODE

**index.html**

```
<!-- Home Page -->
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstra
p.min.css" rel="stylesheet">
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.
bundle.min.js"></script>
    <title>Handwritten Digit Recognizer</title>
    <style>
        body
        {
            background-image: url("{{ url_for('static',
filename='img.jpg') }}");
            width: 100vw;
            height: 100vh;
            position: fixed;
            font-family: cursive;
        }
        .content
        {
            top: 20%;
            font-size: large;
            background: rgba(255, 255, 255, 0.31);
            border-radius: 16px;
            box-shadow: 0 4px 30px rgba(0, 0, 0, 0.1);
            backdrop-filter: blur(10.7px);
            -webkit-backdrop-filter: blur(10.7px);
        }
    </style>
</head>
<body>

    <!-- Navigation Bar -->
```

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
    <div class="container-fluid">
```

```
          <div class="collapse navbar-collapse" id="navbarText">
            <ul class="navbar-nav me-auto mb-2 mb-lg-0">
              <li class="nav-item">
                <a class="nav-link active" href="#">Home</a>
              </li>
              <li class="nav-item">
                <a class="nav-link active" href="{{
url_for('web_load') }}">Recognize</a>
              </li>
            </ul>
          </div>
        </div>
    </nav>
    <!-- Content -->
    <div class="card content container justify-content-center align-
items-center">
        <h1 class="text-center">Handwritten Recognition System</h1>
        <hr width="70%">
        <p class="container text-center">
            Handwritten Text 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 emails, bank cheque, papers,
            images, etc. Before proper implementation of this
            technology we have relied on writing texts with our own
            hands which can result in errors. It's difficult to
            store and access physical data with efficiency. The
            project presents recognizing the handwritten digits (0
            to 9) from the famous MNIST dataset. Here we will be
            using artificial neural networks/convolution neural
            network.
        </p>
    </div>
</body>
</html>
```

**web.html**

```
<!-- Prediction Page -->
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstra
p.min.css" rel="stylesheet">
```

```
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.
bundle.min.js"></script>
    <title>Prediction</title>
    <style>
      body
        {
            background-image: url("{{ url_for('static',
filename='img2.jpg') }}");
            width: 100vw;
            height: 100vh;
            position: fixed;
            font-family: cursive;
        }
        .content,#outputDig
        {
          top: 20%;
          font-size: large;
          background: rgba(255, 255, 255, 0.09);
          border-radius: 16px;
          box-shadow: 0 4px 30px rgba(0, 0, 0, 0.1);
          backdrop-filter: blur(0px);
          -webkit-backdrop-filter: blur(0px);
        }
        #outputDig
        {
          width: fit-content;
        }
    </style>
</head>
<body>
  <!-- Navigation Bar -->
    <nav class="navbar navbar-expand-lg navbar-light bg-light">
        <div class="container-fluid">
          <div class="collapse navbar-collapse" id="navbarText">
            <ul class="navbar-nav me-auto mb-2 mb-lg-0">
              <li class="nav-item">
                <a class="nav-link active" href="{{ url_for('home')
}}">Home</a>
              </li>
              <li class="nav-item">
                <a class="nav-link active" href="#">Recognize</a>
              </li>
            </ul>
          </div>
        </div>
    </nav>
```

```
    <!-- Content -->
    <div class="card content container justify-content-center align-
items-center">
      <h1 class="text-center">Digit Recognition</h1>
      <hr width="70%">
      <div class="but container d-flex justify-content-evenly">
        <form action="/success" method="POST"
enctype="multipart/form-data">
          <input type="file" class="btn btn-default" id="file"
name="file">
          <input type="submit" class="btn btn-default"
value="Recognize">
        </form>
      </div>

      <div class="card container" id="outputDig">
        <div class="card-body">
          <h3 class="card-text fw-italics" id="outp">Recognized
Digit is {{name}}</h3>
        </div>
      </div>

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

**app.py**

```
#flask - it is used to run/serve application
from flask import Flask, render_template, request
from keras.utils import np_utils #used for one-hot encoding
from tensorflow.keras.datasets import mnist #mnist dataset
import tensorflow as tf
import numpy as np
from tensorflow.keras.models import load_model
from PIL import Image

app=Flask(__name__)
model=load_model(r'model.h5')

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

@app.route('/success', methods = ['POST'])
def success():
    if request.method == 'POST':
        f = request.files['file']
```

```
        f.save(f.filename)
        #convert image to required format
        img=Image.open(f).convert("L")

        #resizing of input image
        img = img.resize((28, 28))

        #converting to image
        img2arr = np.array(img)

        #reshaping according to our requirement
        img2arr = img2arr.reshape(1, 28, 28, 1)

        #Predicting the Test set results
        y_pred=model.predict(img2arr)
        # print(y_pred)
        # print(np.argmax(y_pred, axis=1))
        return render_template("web.html", name = np.argmax(y_pred,
axis=1))

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

if_name_=="__main__":
    app.run(debug=True)
```

# CHAPTER 14

# REFERENCES

[1] Priyank Shah, Nitiket Shinde, Deep Limbad, Ashwini Save, “SnapSolve — A Novel Mathematics Equation Solver using Deep Learning”

[2] Ibrahim Karabayir, Oguz Akbilgic, Nihat Tas, “A Novel Learning Algorithm to Optimize Deep Neural Networks: Evolved Gradient Direction Optimizer (EVGO)”

[3] Yubin Zang, Minghua Chen, Sigang Yang, Hongwei Chen, “Electro – Optical Neural Networks Based on Time – Stretch Method”

[4] Barbara Wilk, Malgorzata Augustyn, “Parameterization of the Acceleration Signals Recorded During Handwriting”

[5] Nikolaos Passalis, Anastasios Tefas, “Neural Network Knowledge Transfer using Unsupervised Similarity Matching”

[6] Aparna.A ,.I.Muthumani, “Optical Character Recognition for Handwritten Cursive English characters”, International Journal of Computer Science and Information Technologies, Vol. 5 (1) , 2014, 847-848.

[7] K. Gaurav and Bhatia P. K., DzAnalytical Review of Preprocessing Techniques for Offline Handwritten Character Recognitiondz, ´nd International Conference on Emerging Trends in Engineering & Management, ICETEM, 2013.

[8] Salvador España-Boquera, Maria J. C. B., Jorge G. M. and Francisco Z. M., DzImproving Offline Handwritten Text Recognition with Hybrid HMM/ANN Modelsdz, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 33, No. 4, April 2011.

[9] U. Pal, T. Wakabayashi and F. Kimura, DzHandwritten numeral recognition of six popular scripts,dz Ninth International conference on Document Analysis and Recognition ICDAR 07, Vol.2, pp.749-753, 2007.

[10] Anita Pal & Dayashankar Singh, DzHandwritten English Character RecognitionUsing Neural,dz Network International Journal of Computer Science & Communication. Vol. 1, No. 2, July-December 2010, pp. 141-144.