HANDWRITTEN DIGIT RECOGNITION SYSTEM

ARTIFICIAL INTELLIGENCE (AI DOMAIN)

TEAM ID: PNT2022TMID11500

A PROJECT REPORT

Submitted by

K. B. Hemapriya (910619106022)

R. S. Bhuvaneshwari (910619106009)

V. S. Chandra (910619106011)

N. L. Karthiga Mai (910619106025)

R. Nandhini (910619106033)

Department of Electronics and Communication Engineering

K.L.N. College of Engineering

Pottapalayam, Sivaganga District

Tamilnadu - 630612

Team Mentor	**Evaluator**	**Industry Mentor**
Dr. T. R. Muthu	Dr. S. Subha	Ms. Pradeepthi

November 2022

ANNA UNIVERSITY: CHENNAI 600025

BONAFIDE CERTIFICATE

Certified that this project report "HANDWRITTEN DIGIT RECOGNITION SYSTEM USING CNN" is the bonafide work of

"**HEMAPRIYA K B (910619106022),**

BHUVANESHWARI R S (910619106009),

CHANDRA V S (910619106011),

KARTHIGA MAI N L (910619106025),

NANDHINI R (910619106033)",

Who carried out the project work under our supervision.

SIGNATURE	**SIGNATURE**
FACULTY MENTOR	**FACULTY EVALUATOR**
DR. T. R. MUTHU	DR. S. SUBHA
ASSISTANT PROFESSOR (Sr.Gr)	ASSISTANT PROFESSOR
ELECTRONICS AND COMMUNICATION ENGINEERING	ELECTRONICS AND COMMUNICATION ENGINEERING
K.L.N COLLEGE OF ENGINEERING	K.L.N COLLEGE OF ENGINEERING

SIGNATURE

DR. V. KEJALAKSHMI

HEAD OF THE DEPARTMENT

ELECTRONICS AND COMMUNICATION ENGINEERING

K.L.N COLLEGE OF ENGINEERING

ABSTRACT

Handwriting recognition is one of the compelling research works going on because every individual in this world has their own style of writing. The recognition of handwritten content is the process of converting non-intelligent information such as images into machine edit-able text. Because of the progress in the field of science and technology, everything is being digitalized to reduce human effort. Hence, there comes a need for handwritten digit recognition in many real-time applications. MNIST data set is widely used for this recognition process, and it has 70000 handwritten digits. We use Deep learning algorithm like Convolutional Neural Networks to train these images and identify the digits. Web application is created where the user can upload an image of a handwritten digit. This image is analyzed by the model and the detected result is returned on to UI.

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

INTRODUCTION TO HANDWRITTEN DIGIT RECOGNITION

‘Digits’ are a part of our everyday life, be it License plate on our cars or bike, the price of a product, speed limit on a road, or details associated with a bank account. In the case of a text which is unclear, it is easier to guess the digits in comparison to the alphabets. Machine Learning and Deep Learning are reducing human efforts in almost every field. Moreover, a solution achieved using ML and DL can power various applications at the same time, thereby reducing human effort and increasing the flexibility to use the solution. One such solution is a handwritten digit recognition system that can be used in postal mail sorting, bank check processing, form data entry, etc.

1.1 PROJECT OVERVIEW

The Recognition of Handwritten Digit can be done by several classification methods using Machine Learning, which have been developed and used for this purpose, such as K-Nearest Neighbors, SVM Classifier, Random Forest Classifier, etc., but these methods, whilst having the accuracy of 92%, are not adequate for real-world purposes.

In current years, the research community has been gaining significant interest in deep learning-based strategies to remedy a range of supervised, unsupervised and reinforced getting to know problems. One of the most regularly occurring and broadly used strategies is Convolution neural networks (CNN's), a kind of neural networks which can extract relevant features robotic-ally from enter information. Here, we will learn about the implementation of well-known MNIST facts set to predict and recognize handwritten digits the use of deep gaining knowledge of

techniques and Machine Learning algorithms. This is not a new topic and the MNIST information set is nonetheless very common and essential to take a look at and affirm new algorithms after quite a few decades. The project requires a lot of libraries such as primary ML libraries, deep mastering libraries, EDA (Exploratory Data Analysis) and tensor-flow the place tensor-flow is used as back-end with keras at some stage in the development process.

1.2 PURPOSE

The purpose of Hand-written digit recognition is to recognize and predict digits from an extensive variety of sources such as emails, documents, images, letters, etc. This has been a problem of lookup for decades. Some areas of lookup consist of verification of signatures, processing of bank checks, interpretation of postal addresses from envelopes and many extra are turn out to be less difficult and extra handy through digit recognition methods.

The handwritten digit recognition is the ability of computers to recognize human handwritten digits. It is a hard task for the machine because handwritten digits are not perfect and can be made with many different flavors. 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. CNN is primarily used in object recognition by taking images as input and then classifying them in a certain category. Handwritten digit recognition is one of that kind.

CHAPTER 2

LITERATURE SURVEY

As early notable attempt in the area of handwritten digit recognition can be done in many areas by various approaches such as K-Nearest Neighbors, SVM Classifier, Random Forest Classifier, etc., In the pattern recognition field, growing interest has been shown in recent years for multiple classifier systems and particularly for bagging, boosting and random sub-spaces. Those methods aim at inducing an ensemble of classifiers by producing diversity at different levels. Following this principle, Breiman has introduced in 2001 another family of methods called random forest. Our work aims at studying those methods in a strictly pragmatic approach, in order to provide rules on parameter settings for practitioners. For that purpose, we have experimented the forest-RI algorithm, considered as the random forest reference method, on the MNIST handwritten digits database.

Using Random Forest Classifier, the accuracy of recognition is about 93%. Handwritten digits recognition is becoming a greater demand as one of the computer vision techniques. Computer vision has many applications for interfacing human and a machine. A blob-based classification of image is another algorithm used to solve handwritten digits recognition problems. The algorithm recognizes digits by classifying them as blobs with and without stem. In this method the mathematical morphology was used to construct the classification models. This is a technique of recognition of digits by identification of blobs and stems. The problem with this method is that it is not able to recognize the broken digits of large gap and digits with extra stems. Multiple perception layers with neural network is another model used in handwritten digits recognition. In their study, the authors used MNIST dataset for handwritten digits recognition, the dataset was

trained with gradient descent back-propagation algorithm and the verified with the feed-forward algorithm.

A Decision tree classification is one of the simplest machine learning models used in handwritten digits recognition and a comparative study performed on decision tree classification and random forests. The random forest performed superior than decision tree classification. The Decision Tree Classifier seems to have a success rate of 85% on the test dataset given to it. The system itself may have some confusion when analyzing similar digits. This occurs because the digits themselves are handwritten and because the data has been gathered from over 250 different writers and some of them write different digits in a similar manner.

2.1 EXISTING PROBLEM

Handwritten Digit Recognition is an interesting machine learning problem in which we have to identify the handwritten digits through various classification algorithms. There are a number of ways and algorithms to recognize handwritten digits such as SVM (Support Vector Machine), MLP (Multi-Layered Perception), K-NN, Decision Trees, Random Forests, etc.

- Complexity in recognizing digits.
- More number of computations required.
- Less Accuracy.
- Poor Performance on Model Building.

2.2 REFERENCES

- Novel Deep Neural Network Model for Handwritten Digit Classification and Recognition by Ayush Kumar Agrawal and Vineet Kumar Awasthi in the year 2021.

- Handwritten Digit Recognition System Based on LRM and SVM Algorithm by Hafiz Ahamed, Syed Md Ishraq Alam in the year 2019.
- Handwriting Recognition Using SVM by Kavya N, Madhushree N, Maheshwari S, Manasa H C, Anandhi G in the year 2019.
- Handwritten Digit Recognition with KNN and SVM Classifier by Shubham Ingole and Dhruv Sonone in the year 2018.
- A Robust End-to-End System to Solve the Handwritten Digit String Recognition Problem in Real Complex Scenarios by Arthur Flor De Sousa Neto, Byron Leite Dantas Bezerra (Member, IEEE), Estanislau Baptista Lima and Alejandro Héctor Toselli in the year 2020.

2.3 PROBLEM STATEMENT DEFINITION

Handwriting recognition has been the main subject of research for almost the last forty years. This research work analyzes the behaviour of classification techniques (CNN) in a large handwriting dataset (MNIST) to predict a digit. Machine-learning techniques, particularly when applied to Neural Networks like CNN or ANN, have played an increasingly important role in the design of these recognition systems. Several methods have been developed in handwritten digit recognition and these methods have been classified into categories: knowledge-based methods, feature-based methods, template-based methods and appearance-based methods. Errors in Digit recognition cause severe problems like digits written on a bank cheque if recognized erroneously could result in unfortunate consequences.

Problem Statement

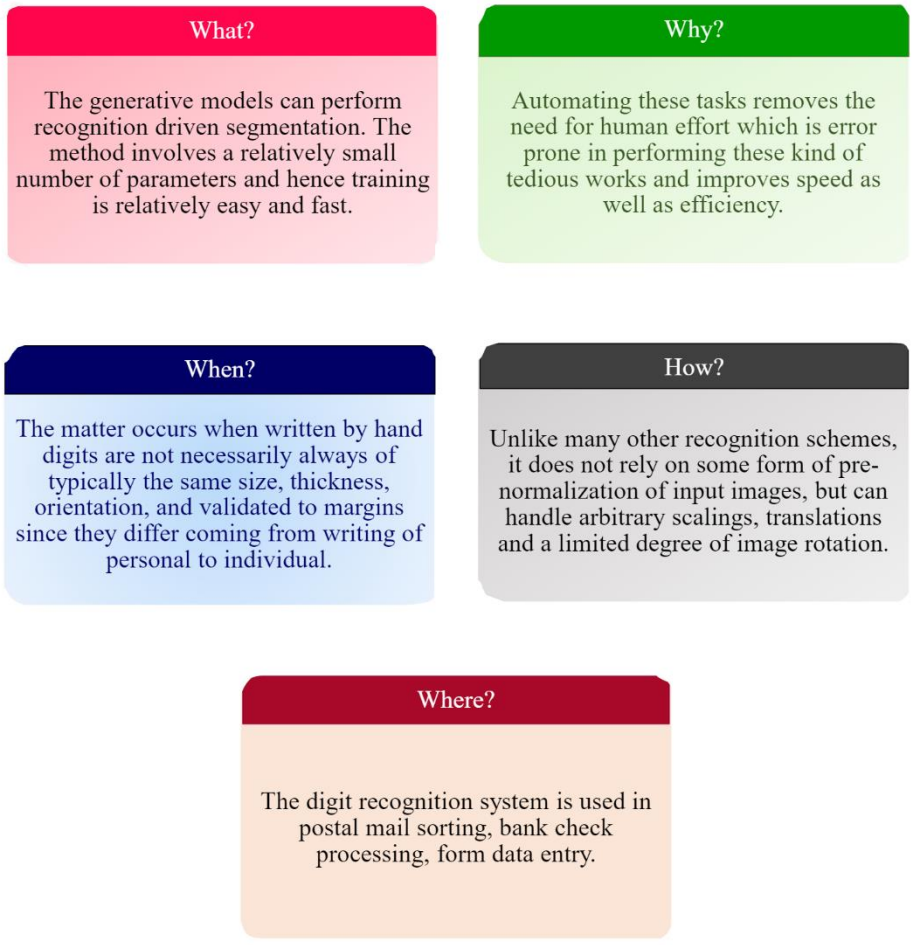


Figure 2.3.1 Problem Statement

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 behaviors and attitudes. It is a useful tool to helps tears 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.

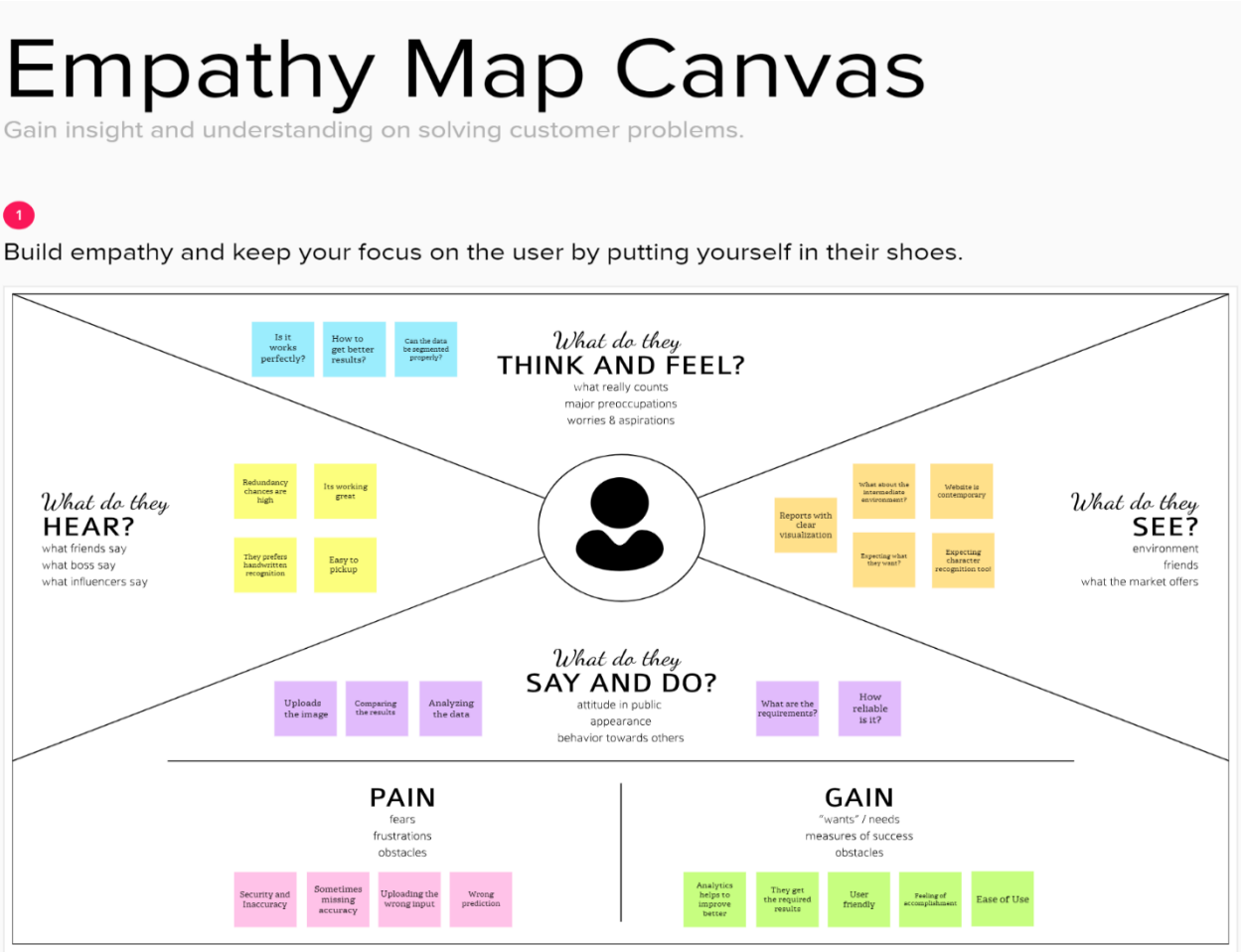


Figure 3.1.1 Empathy Map Canvas

3.2 IDEATION & BRAINSTORMING

Brainstorming is a group problem-solving method that involves the spontaneous contribution of creative ideas and solutions. This technique requires intensive, freewheeling discussion in which every member of the group is encouraged to think aloud and suggest as many ideas as possible based on their diverse knowledge.

Ideation is the process of forming ideas from conception to implementation, most often in a business setting. Ideation is expressed via graphical, written, or verbal methods, and arises from past or present knowledge, influences, opinions, experiences, and personal convictions.

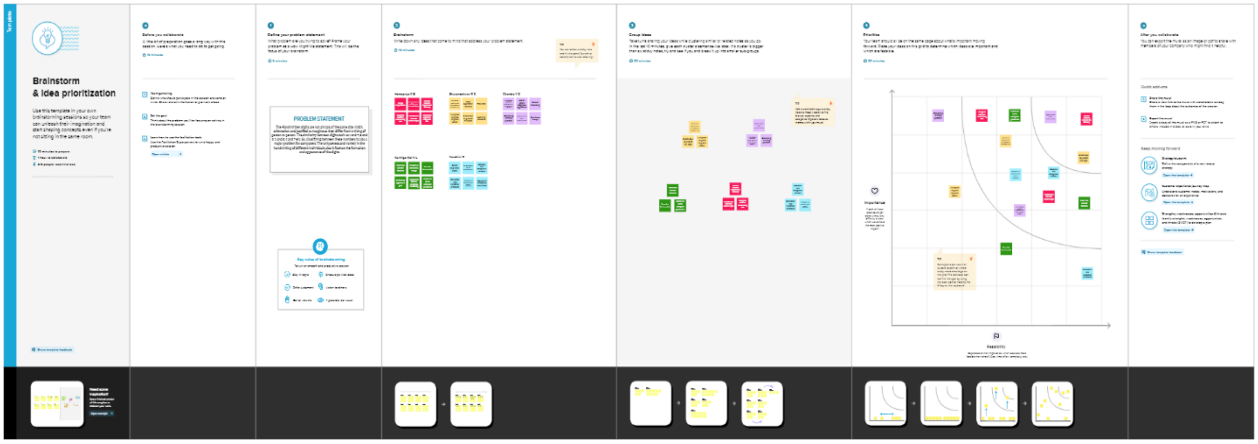


Figure 3.2.1 Ideation & Brainstorming

3.3 PROPOSED SOLUTION

After doing a thorough investigation into the problem, you would have a solid grasp of how the problem occurred. Thus, you need to propose a practical solution or suggest several approaches to understanding and rectifying the issue. The below table contains Problem Statement, Idea/Solution description, Novelty / Uniqueness, Social Impact/Customer Satisfaction, Business Model (Financial Benefit) and Scalability of Solution.

Table 3.3.1 Proposed Solution

S.NO.	PARAMETER	DESCRIPTION
1.	Problem Statement (problem to be solved)	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. The similarity between digits such as 1 and 7, 5 and 6, 3 and 8, 2 and 7 etc. So, classifying between these numbers is also a major problem for computers. The uniqueness and variety in the handwriting of different individuals also influence the formation and appearance of the digits.
2.	Idea/ Solution Description	1. It is the capability of a computer to fet the mortal handwritten integers from different sources like images, papers, touch defences. 2. It allows user to translate all those signature and notes into electronic words in a text document format and this data only requires far less physical space than the storage of the physical copies.
3.	Novelty/ Uniqueness	1. Artificial Intelligence developed the app called Handwritten digit Recognizer. 2. It converts the written word into digital approximations and utilizes complex algorithms to identify characters before churning out a digital approximation.
4.	Social Impact/ Customer Satisfaction	1. To receive and interpret intelligible handwritten input from sources such as paper documents, photographs, touch-screens and other devices. 2.The main purpose of this application is to convert handwritten digits into machine readable formats.
5.	Business Model (Financial Benefit)	1.AI can generate revenue through consumer cases like reading text from stylus writing or camera applications to academic research on how handwriting recognition works in humans. 2.It minimizing the human labour and enhancing the solution versatility.
6.	Scalability of Solution	1. We described the data generation pipeline and presented a series of techniques to generate better data. 2. The experimental results showed that models can

		be improved by using the generated data, and we obtained the best model by combining the generated data and small amount of real images.

3.4 PROBLEM SOLUTION FIT

The Problem-Solution Fit simply means that you have found a problem with your customer and that the solution you have realized for it actually solves the customer's problem. It helps entrepreneurs, marketers and corporate innovators identify behavioral patterns and recognize what would work and why.

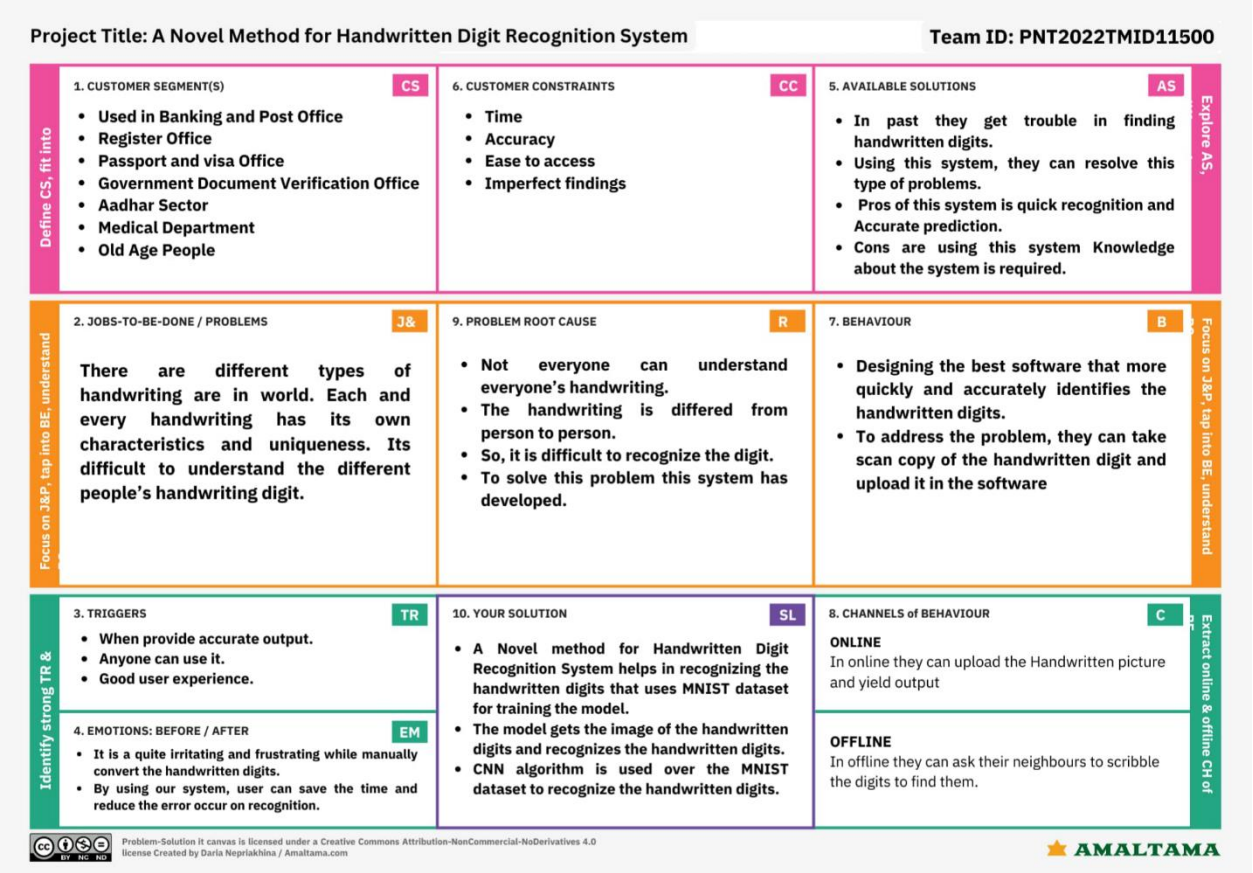


Figure 3.4.1 Problem Solution Fit

CHAPTER 4

REQUIREMENT ANALYSIS

Requirements analysis is nearly self-explanatory: it's the process of defining the expectations of stakeholders on a project. It analyzes, documents, validates and manages all of the identified requirements while considering the possibility that there are conflicting requirements among stakeholders.

Solution Requirements (Functional & Non-functional)

4.1 Functional Requirements:

Functional Requirements define what a product must do, what its features and functions are. They are product features or functions that developers must implement to enable users to accomplish their tasks. Generally, functional requirements describe system behavior under specific conditions. Following are the functional requirements of the proposed solution.

Table 4.1.1 Functional Requirements

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIn
FR-2	User Confirmation	Login via registered Username and Password
FR-3	Uploading images	Able to input the handwritten images into the application
FR-4	Recognizing digits	Display the recognized digits from the input images tothe user

4.2 Non-functional Requirements:

Nonfunctional Requirements, not related to the system functionality, rather define how the system should perform. Here, we'll just briefly describe the most typical nonfunctional requirements. Following are the non-functional requirements of the proposed solution.

Table 4.2.1 Non-functional Requirements

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The software is very easy to use and reduces the learning work. To recognize the digits from bank cheque, papers, numeric entry in forms etc. The application needs to respond smoothly so that the user can use the application effectively and need to be a user-friendly application.
NFR-2	**Security**	The handwritten digit recognition can be used by banking sector where it can be used to maintain the security pin numbers, it can be also used for blind peoples by using sound output. Ensure the security by authenticating the users using their username and password.
NFR-3	**Reliability**	The application does not show any error during the recognition of the digits from the uploaded images. This software will work reliably for low resolution images and not for graphical images.
NFR-4	**Performance**	Needs to respond fast and provide the output even for the complex handwritings. Handwritten characters in the input image will be recognized with an accuracy of about 90% and more.
NFR-5	**Availability**	Need to available for all users at any time and can be able to input the handwritten images to the application easily. This system will retrieve the handwritten text regions only if the image contains written text in it.
NFR-6	**Scalability**	It can be able to handle N numbers of users at the same time with faster response and recognize the digits effectively. It contains thousands of handwritten digits that have been used in the development of programs.

CHAPTER 5

PROJECT DESIGN

Project design is a major first step towards a successful project. A project design is a strategic organization of ideas, materials and processes for the purpose of achieving a goal.

5.1 DATA FLOW DIAGRAMS

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

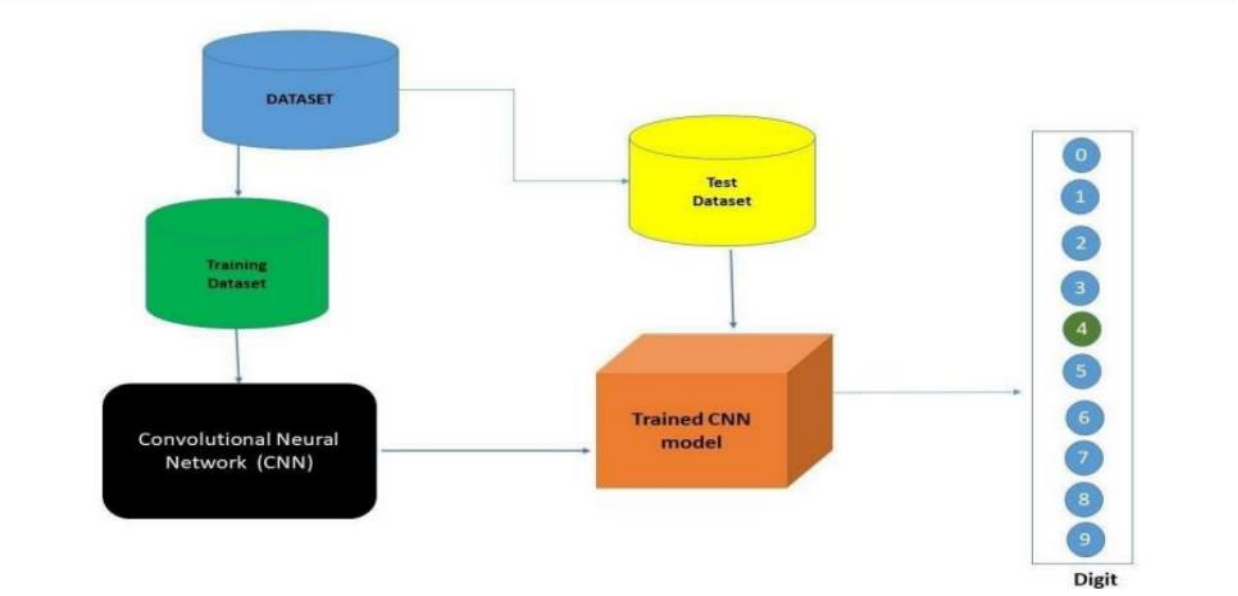


Figure 5.1.1 Data Flow Diagram

5.2 SOLUTION & TECHNICAL ARCHITECTURE

Technical Architecture:

Technical Architecture (TA) is a form of IT architecture that is used to design computer systems. It involves the development of a technical blueprint with regard

to the arrangement, interaction, and interdependence of all elements so that system-relevant requirements are met.

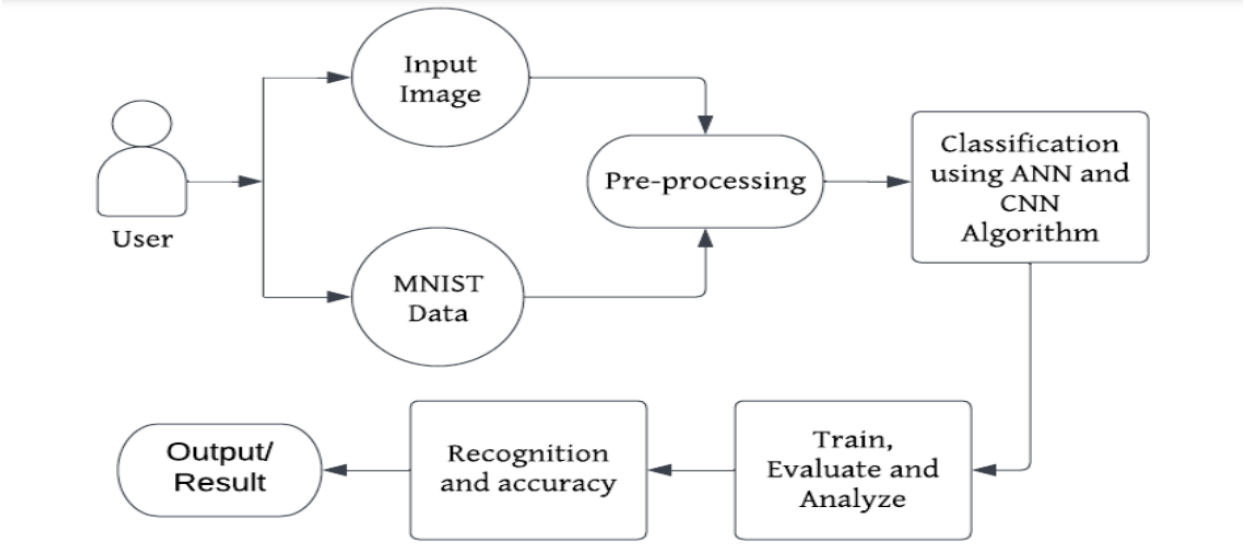


Figure 5.2.1 Technology Stack (Architecture & Stack)

The components and technologies used in Technology architecture are listed below.

Table 5.2.2 Components & Technologies

S. No	Component	Description	Technology
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	Understanding and loading the data, Feature Extraction, Subsampling Layer	Java / Python
3.	Application Logic-2	Creating the model and train the model	IBM Watson STT service
4.	Application Logic-3	Evaluating the model and predicting the result	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, Cloud Foundry, Kubernetes, etc.

Table 5.2.3 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	To optimize image processing efficacy and response rate for visual ability.	3 – tier, Micro-services
4.	Availability	The features for handwritten digit recognition have been introduced. These features are based on shape analysis of the digit image and extract slant or slope information. They are effective in obtaining good recognition accuracy.	Distributed servers, IBM cloud
5.	Performance	The standard implementations of neural networks achieve an accuracy of ~ (98–99) percent in correctly classifying the handwritten digits.	Number of requests per sec, use of Cache, use of CDN's

5.3 USER STORIES

A user story is the smallest unit of work in an agile framework. It's not a feature, but an end goal that the user has when using the software. The user story will convey what the user wants to achieve and states it in a simple, non-technical way. In Agile projects, user stories are organized in a backlog, which is an ordered list of product functions.

Table 5.3.1 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Home	USN-1	As a user, I can view the guide and awareness to use this application.	I can view the awareness to use this application and its limitations.	Low	Sprint-1
		USN-2	As a user, I'm allowed to view the guided video to use the interface of this application.	I can gain knowledge to use this application by a practical method.	Low	Sprint-1
		USN-3	As a user, I can read the instructions to use this application.	I can read instructions also to use it in a user-friendly method.	Low	Sprint-2
	Recognize	USN-4	As a user, in this prediction page I get to choose the image.	I can choose the image from our local system and predict the output.	High	Sprint-2
	Predict	USN-6	As a user, I'm Allowed to upload and choose the image to be uploaded	I can upload and choose the image from the system storage and in any virtual storage.	Medium	Sprint-3
		USN-7	As a user, I will train and test the input to get the maximum accuracy of output.	I can be able to train and test the application until it gets maximum accuracy of the result.	High	Sprint-4
		USN-8	As a user, I can access the MNIST data set	I can access the MNIST data set to produce the accurate result.	Medium	Sprint-3
Customer (Web user)	Dashboard	USN-9	As a user, I can view the guide to use the web app.	I can view the awareness of this application and its limitations.	Low	Sprint-1

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
	Recognize	USN-10	As a user, I can use the web application virtually anywhere.	I can use the application portably anywhere.	High	Sprint-1
		USN-11	As it is an open source, can use it cost freely.	I can use it without any payment to be paid for it to access.	Medium	Sprint-2
		USN-12	As it is a web application, it is installation free	I can use it without the installation of the application or any software.	Medium	Sprint-4
	Predict	USN-13	As a user, I will train and test the input to get the maximum accuracy of output.	I can be able to train and test the application until it gets maximum accuracy of the result.	High	Sprint-4
Customer Care Executive		USN-14	As a user, I can use the web application virtually anywhere.	I can use the application portably anywhere and get the accurate result.	Medium	Sprint-2
Administrator		USN-15	As a user, I can use the web/mobile application virtually anywhere.	I can use the application portably anywhere and get the accurate result.	High	Sprint-2

CHAPTER 6

PROJECT PLANNING & SCHEDULING

Project Planning (Product Backlog, Sprint Planning, Stories, Story points)

The process of planning primarily deals with selecting the appropriate policies and procedures in order to achieve the objectives of the project. Scheduling converts the project action plans for scope, time cost and quality into an operating timetable.

6.1 SPRINT PLANNING & ESTIMATION

During sprint planning, we break the stories down into tasks, estimate those tasks, and compare the task estimates against our capacity. It's that, not points, that keep us from overcommitting in this sprint. No need to change the estimate.

Table 6.1.1 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	10	High	Hemapriya, Chandra
Sprint-1	Login	USN-2	As a user, I can log into the application by entering email & password.	10	High	Nandhini, Hemapriya
Sprint-1	Data Collection & Pre-processing	USN-3	As a user, I can upload any kind of image with the pre-processing step is involved in it.	10	High	Karthiga mai, Hemapriya
Sprint-2	Upload input Image	USN-4	As a user, I can able to input the images of digital documents, handwritten documents or images to the application.	8	Medium	Bhuvaneshwari, Chandra
Sprint-2	Building the ML model	USN-5	As a user, I will get an application with ML model which provides high accuracy of recognized handwritten digit.	9	High	Hemapriya, Karthiga mai
Sprint-2		USN-6	As a user, I can pass the handwritten digit image for recognizing the digit.	5	Medium	Nandhini, Karthiga mai
Sprint-3	Building the UI Application	USN-7	As a user, I will upload the handwritten digit image to the application by clicking an upload button.	8	Medium	Bhuvaneshwari, Hemapriya
Sprint-3		USN-8	As a user, I can know the details of the fundamental usage of the application.	2	Low	Chandra, Nandhini
Sprint-3		USN-9	As a user, I can see the predicted / recognized digits in the application.	10	High	Hemapriya, Nandhini
Sprint-4	Train and deploy the model in IBM Cloud	USN-10	As a user, I can access the web application and make the use of the product from anywhere.	20	High	Chandra, Hemapriya
Sprint-4	Recognize the digit	USN-11	As a user I can able to get the recognised digit as output from the images of digital documents or images.	10	Medium	Karthiga mai, Bhuvaneshwari

6.2 SPRINT DELIVERY SCHEDULE

A Sprint is a set period where an agile team works to complete a specific set of development tasks.

Table 6.2.1 Sprint Delivery Schedule

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

Velocity:

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

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{6} = 3.33$$

Burndown Chart:

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

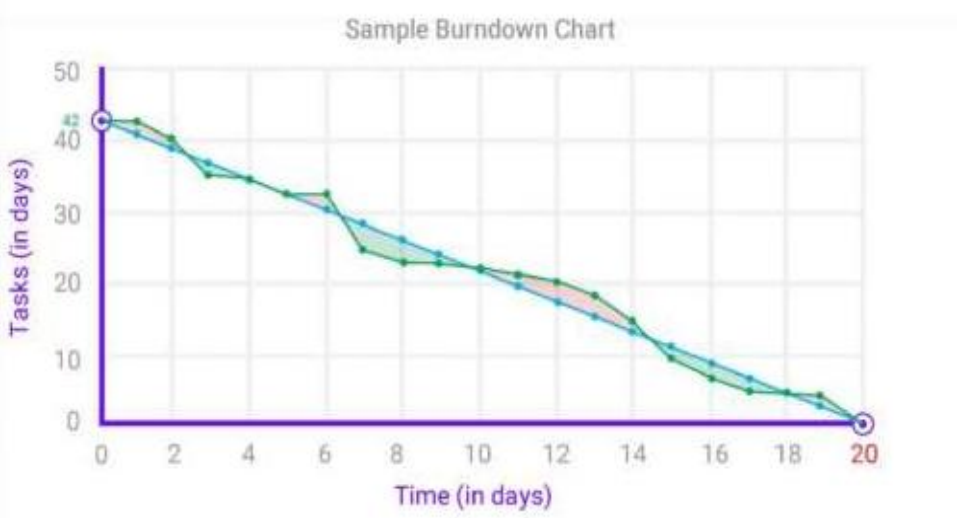


Figure 6.2.2 Burndown Chart

6.3 REPORTS FROM JIRA

JIRA provides different types of reports within a project. It helps to analyze the Progress, Issues, Showstoppers and Timeliness of any Project.

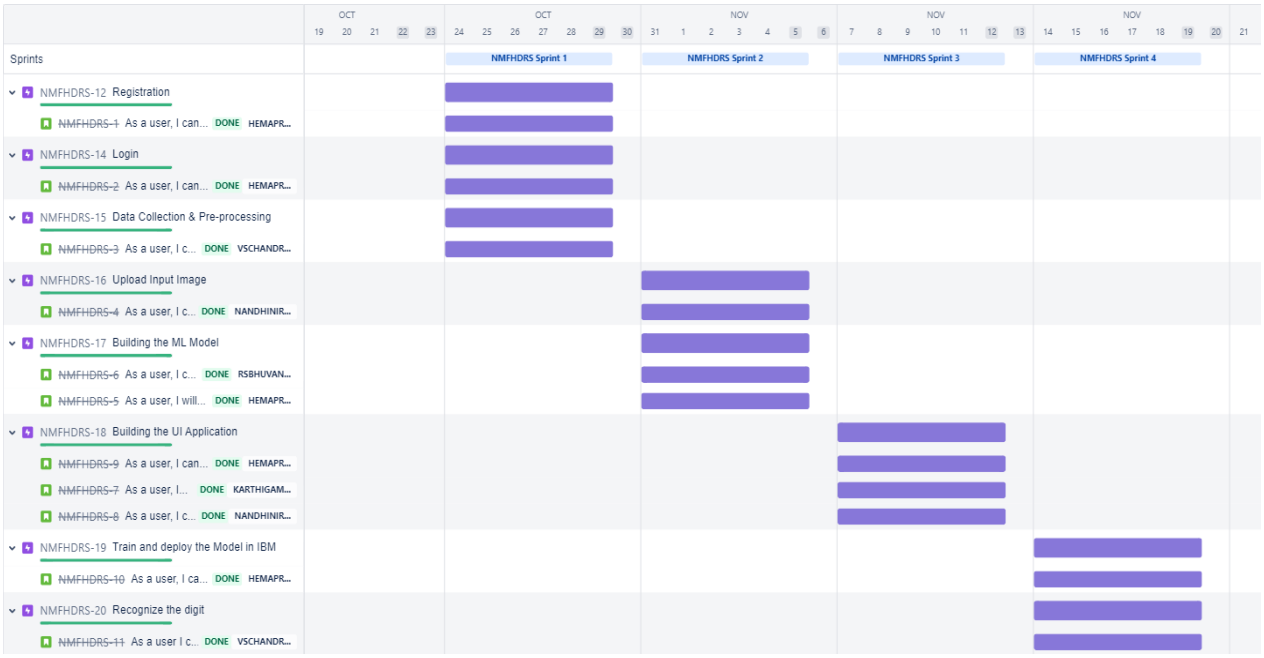


Figure 6.3.1 JIRA Road Map

CHAPTER 7

CODING & SOLUTIONING

7.1 FEAUTRE 1

UNDERSTANDING THE DATA

Importing the required libraries which are required for the model to run. The dataset for this model is imported from the Keras module.

The dataset contains ten classes: Digits from 0-9. Each digit is taken as a class.

Import the necessary packages

```
In [1]:
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from keras.utils import np_utils
from tensorflow.keras.datasets import mnist
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, Dense, Flatten
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.models import load_model
from PIL import Image, ImageOps
```

Load data

```
In [2]:
(X_train, y_train), (X_test, y_test) = mnist.load_data()
```

Data Analysis

```
In [3]:
print(X_train.shape)
print(X_test.shape)
```

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

```
In [4]:
X_train[0]
```

```
Out[4]: array([[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
        [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
        [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
        [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
        [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
        [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   3,
         18,  18,  18, 126, 136, 175,  26, 166, 255, 247, 127,   0,   0,
          0,   0],
        [  0,   0,   0,   0,   0,   0,   0,   0,  30,  36,  94, 154, 170,
        253, 253, 253, 253, 253, 225, 172, 253, 242, 195,  64,   0,   0,
          0,   0],
        [  0,   0,   0,   0,   0,   0,   0,  49, 238, 253, 253, 253, 253,
        253, 253, 253, 253, 251,  93,  82,  82,  56,  39,   0,   0,   0,
          0,   0],
        [  0,   0,   0,   0,   0,   0,   0,  18, 219, 253, 253, 253, 253,
        253, 198, 182, 247, 241,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
        [  0,   0,   0,   0,   0,   0,   0,   0,  80, 156, 107, 253, 253,
        205,  11,   0,  43, 154,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
```

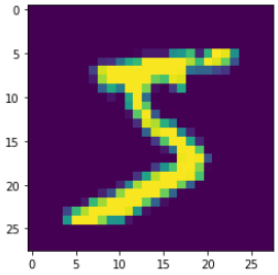
```
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0]], dtype=uint8)
```

In [5]:
```
y_train[0]
```

Out[5]: 5

In [6]:
```
plt.imshow(X_train[0])
```

Out[6]:



DATA PREPROCESSING

Data preprocessing is an iterative process for the transformation of the raw data into understandable and useable forms. Raw datasets are usually characterized by incompleteness, inconsistencies, lacking in behavior, and trends while containing errors. The preprocessing is essential to handle the missing values and address inconsistencies.

Data Pre-Processing

In [7]:
```
X_train = X_train.reshape(60000, 28, 28, 1).astype('float32')
X_test = X_test.reshape(10000, 28, 28, 1).astype('float32')
```

In [8]:
```
number_of_classes = 10
Y_train = np_utils.to_categorical(y_train, number_of_classes)
Y_test = np_utils.to_categorical(y_test, number_of_classes)
```

In [9]:
```
Y_train[0]
```

Out[9]: array([0., 0., 0., 0., 0., 1., 0., 0., 0., 0.], dtype=float32)

CREATE THE MODEL

Creating the model and adding the input, hidden, and output layers to it.

The Sequential model is a linear stack of layers. You can create a Sequential model by passing a list of layer instances to the constructor.

Create model

In [10]:
```
model = Sequential()
model.add(Conv2D(64, (3, 3), input_shape=(28, 28, 1), activation="relu"))
model.add(Conv2D(32, (3, 3), activation="relu"))
model.add(Flatten())
model.add(Dense(number_of_classes, activation="softmax"))
```

In [11]:
```
model.compile(loss='categorical_crossentropy', optimizer="Adam", metrics=["accuracy"])
```

In [12]:
```
model.summary()
```

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

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

 flatten (Flatten)           (None, 18432)             0

 dense (Dense)               (None, 10)                184330

=================================================================
Total params: 203,434
Trainable params: 203,434
Non-trainable params: 0
_________________________________________________________________
```

TRAIN THE MODEL

To start the training of the model we can simply call the model.fit() function of Keras. It takes the training data, validation data, epochs, and batch size as the parameter.

The training of model takes some time. After successful model training, we can save the weights and model definition in the ‘mnist.h5’ file.

Train the model

In [12]:
```
model.fit(X_train, Y_train, batch_size=32, epochs=5, validation_data=(X_test,Y_test))
```

```
Epoch 1/5
1875/1875 [==============================] - 16s 5ms/step - loss: 0.2158 - accuracy: 0.9518 - val_loss: 0.0964 - val_accuracy: 0.9707
Epoch 2/5
1875/1875 [==============================] - 9s 5ms/step - loss: 0.0682 - accuracy: 0.9794 - val_loss: 0.0674 - val_accuracy: 0.9805
Epoch 3/5
1875/1875 [==============================] - 9s 5ms/step - loss: 0.0478 - accuracy: 0.9844 - val_loss: 0.0852 - val_accuracy: 0.9759
Epoch 4/5
1875/1875 [==============================] - 9s 5ms/step - loss: 0.0336 - accuracy: 0.9893 - val_loss: 0.1202 - val_accuracy: 0.9719
Epoch 5/5
1875/1875 [==============================] - 9s 5ms/step - loss: 0.0270 - accuracy: 0.9914 - val_loss: 0.1036 - val_accuracy: 0.9777
```

Out[12]:

TEST THE MODEL

Firstly, we are slicing the x_test data until the first four images. In the next step we the printing the predicted output.

Test the model

In [13]:
```
metrics = model.evaluate(X_test, Y_test, verbose=0)
print("Metrics (Test Loss & Test Accuracy): ")
print(metrics)
```

```
Metrics (Test Loss & Test Accuracy):
[0.1035672277212143, 0.9776999950408936]
```

In [14]:
```
prediction = model.predict(X_test[:4])
print(prediction)
```

```
1/1 [==============================] - 0s 177ms/step
[[6.43197941e-15 8.71634543e-21 7.98728167e-11 7.08215517e-12
  2.27718335e-18 1.36703092e-15 2.37176042e-22 1.00000000e+00
  4.51405352e-13 4.25453591e-13]
 [4.56659687e-15 1.54588287e-10 1.00000000e+00 1.20107971e-13
  1.86926159e-19 3.90255250e-20 1.16102319e-11 4.27834925e-23
  7.33884963e-17 1.86307852e-23]
 [1.37352282e-10 9.99961138e-01 3.40877750e-06 1.50240779e-12
  1.99599867e-07 1.10004057e-05 6.72304851e-11 7.78906983e-09
  2.42337919e-05 3.74607870e-13]
 [1.00000000e+00 5.39840355e-16 1.03082355e-10 4.23198737e-17
  8.17481194e-10 2.49619574e-12 1.66041558e-09 5.06253395e-17
  3.02219919e-13 5.55243709e-08]]
```

In [15]:
```
print(numpy.argmax(prediction, axis=1))
print(Y_test[:4])
```

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

SAVE THE MODEL

The model is saved with .h5 extension as follows:

An H5 file is a data file saved in the Hierarchical Data Format (HDF). It contains multidimensional arrays of scientific data.

Save the model

In [16]:
```
model.save("model.h5")
```

TEST THE SAVED MODEL

Test the saved model

In [22]:
```
model=load_model("model.h5")
```

In [23]:
```
img = Image.open("sample.png").convert("L")
img = img.resize((28, 28))
img2arr = np.array(img)
img2arr = img2arr.reshape(1, 28, 28, 1)
results  = model.predict(img2arr)
results = np.argmax(results,axis = 1)
results = pd.Series(results,name="Label")
print(results)
```

```
1/1 [==============================] - 0s 435ms/step
0    8
Name: Label, dtype: int64
```

7.2 FEATURE 2

RETURNING THE PREDICTION ON UI:

```
import os
import random
import string
from pathlib import Path
import numpy as np
from tensorflow.keras.models import load_model
from PIL import Image, ImageOps


def random_name_generator(n):
    return ''.join(random.choices(string.ascii_uppercase + string.digits, k=n))

def recognize(image):
    model=load_model(Path("./model/model.h5"))

    img = Image.open(image).convert("L")
    img_name = random_name_generator(10) + '.jpg'

    if not os.path.exists(f"./static/data/"):
        os.mkdir(os.path.join('./static/', 'data'))
    img.save(Path(f"./static/data/{img_name}"))

    img = ImageOps.grayscale(img)
    img = ImageOps.invert(img)
    img = img.resize((28, 28))
```

```
    img2arr = np.array(img)
    img2arr = img2arr / 255.0
    img2arr = img2arr.reshape(1, 28, 28, 1)

    results  = model.predict(img2arr)
    best = np.argmax(results,axis = 1)[0]

    pred = list(map(lambda x: round(x*100, 2), results[0]))

    values = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
    others = list(zip(values, pred))

    best = others.pop(best)

    return best, others, img_name
```

BUILDING FLASK APP

The flask file ‘app.py’ which is a web framework written in python for server-side scripting.

Main Function:

This function runs your app in a web browser. Lastly, we run our app on the localhost.

```python
from flask import Flask,render_template,request
from recognizer import recognize

app=Flask(__name__)

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


@app.route('/predict',methods=['POST'])
def predict():
    if request.method=='POST':
      image = request.files.get('photo', '')
      best, others, img_name = recognize(image)
      return render_template("predict.html", best=best, others=others, img_name=img_name)


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

7.3 STORING THE APP IN IBM CLOUD STORAGE SERVICE

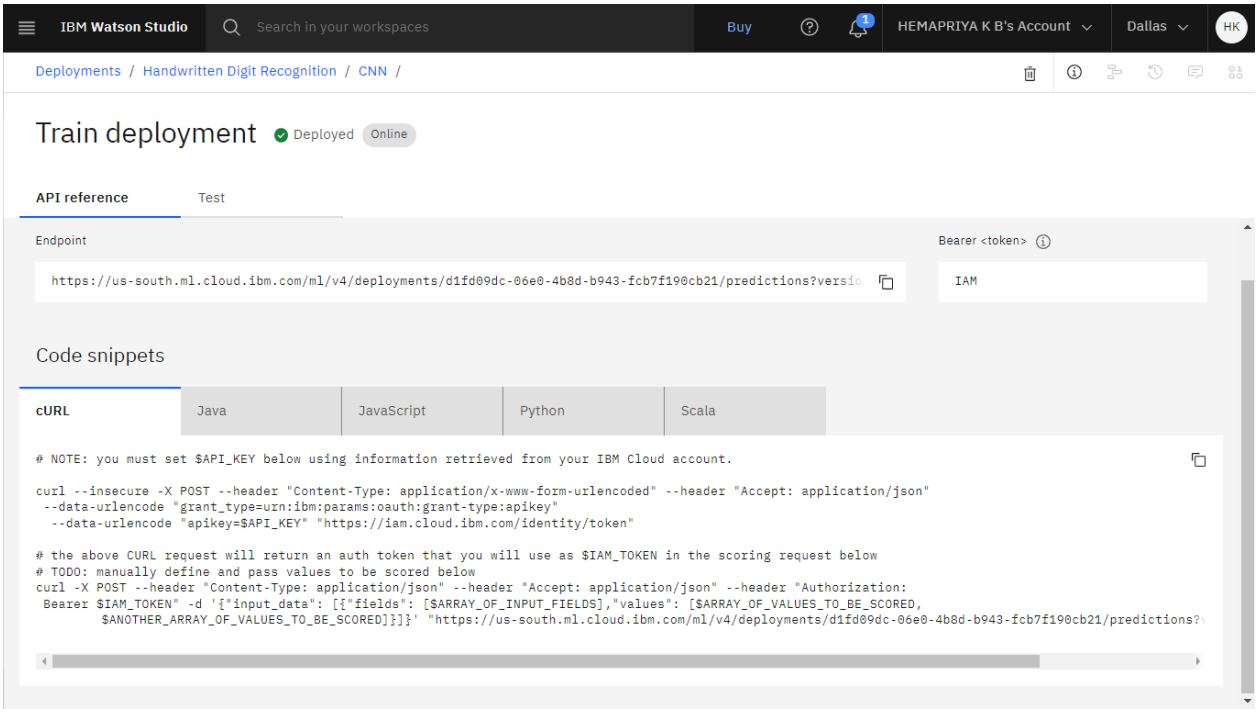


Figure 7.3.1 App Deployment in Cloud

PUBLISHING WEBPAGE ON LOCAL SERVER

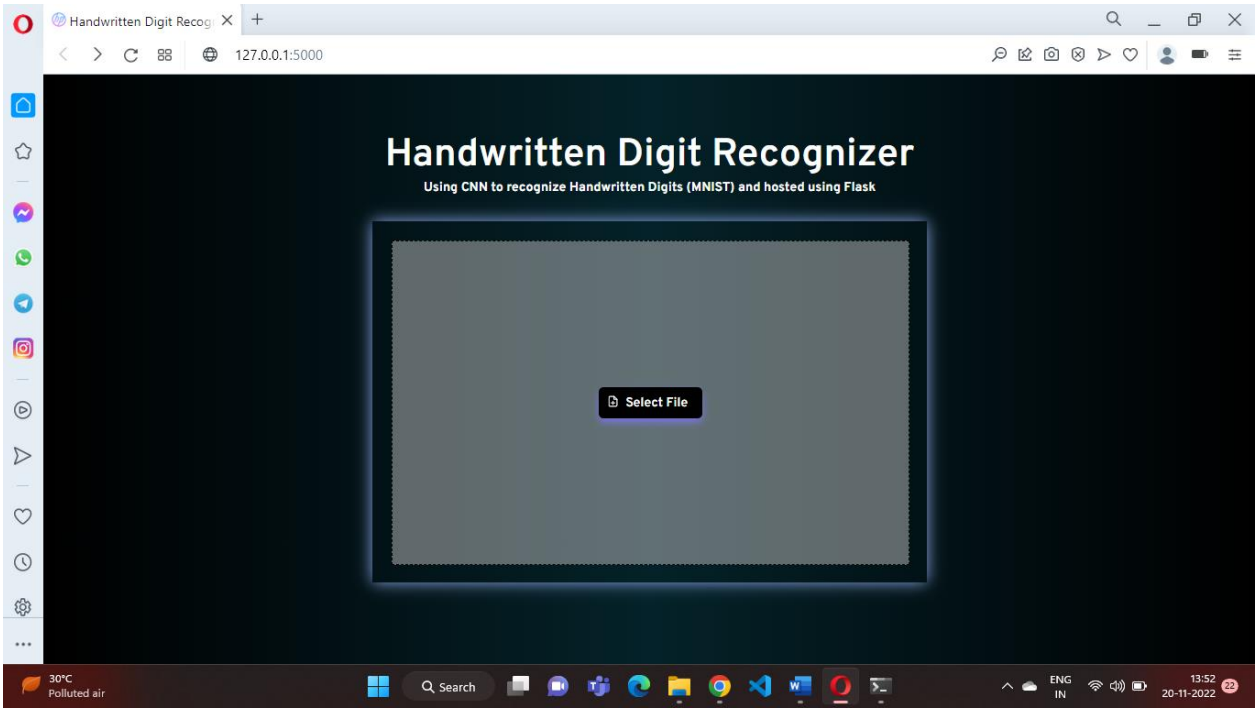


Figure 7.3.2 Home page

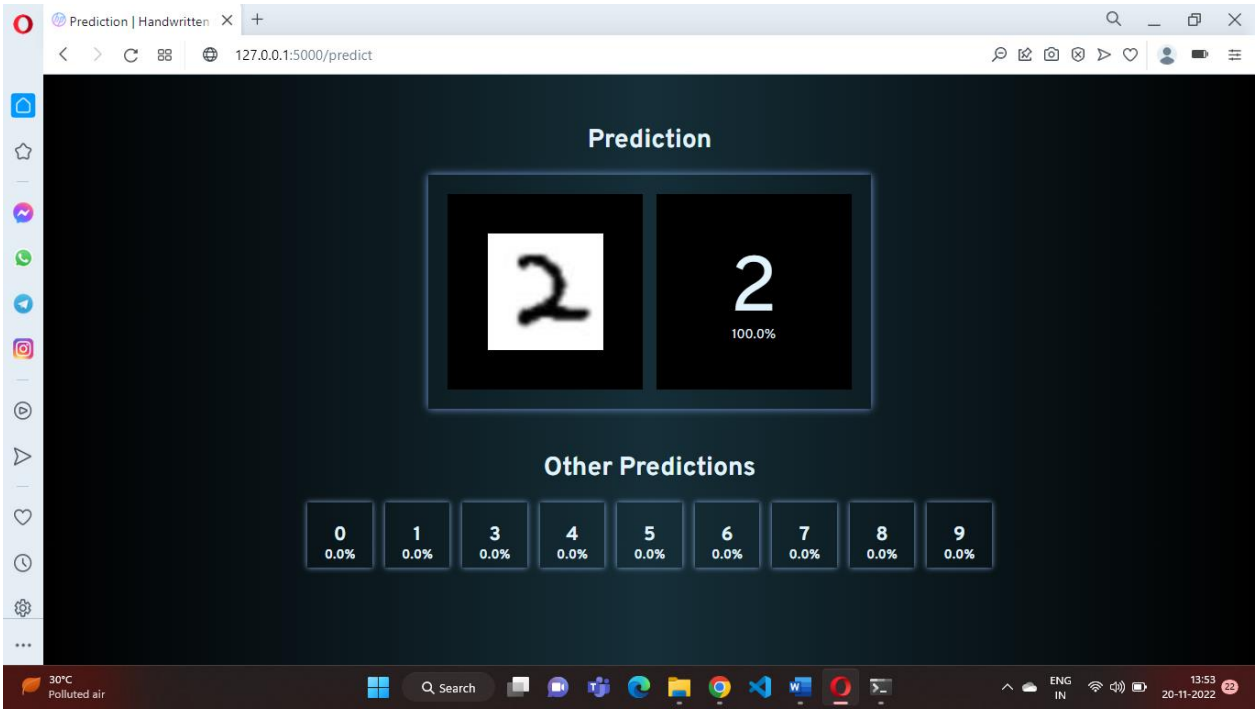


Figure 7.3.3 Prediction for Number 2

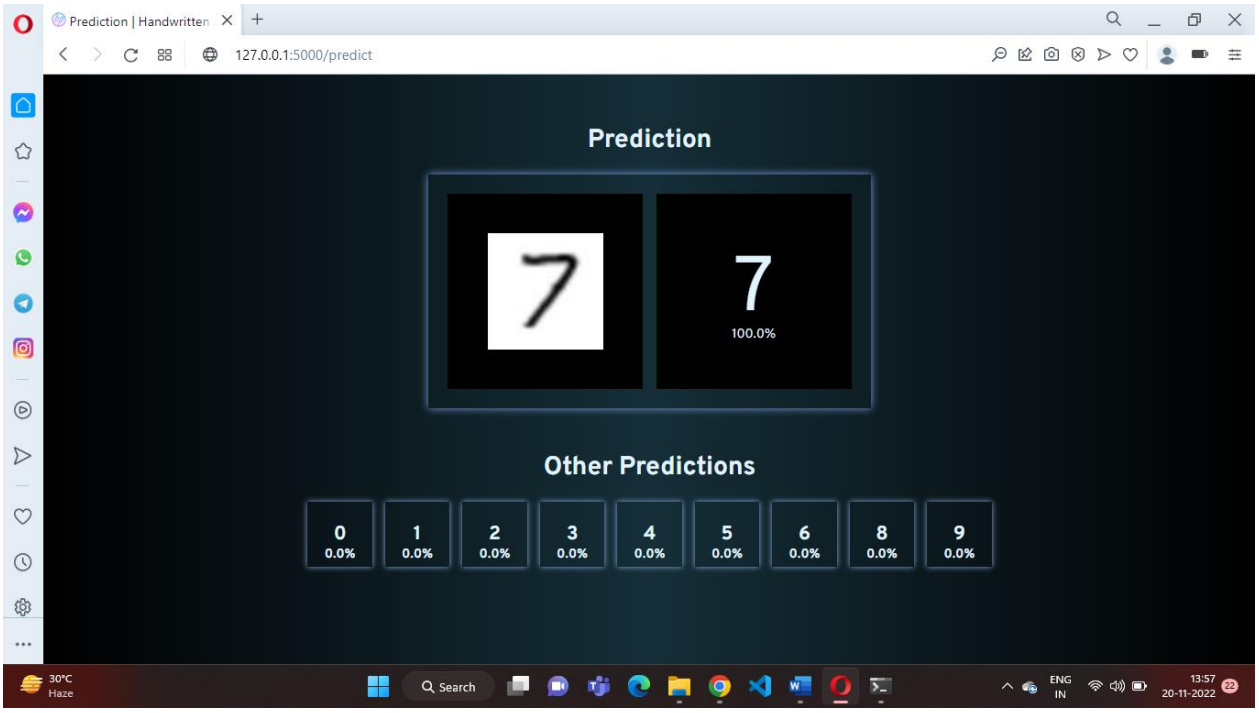


Figure 7.3.4 Prediction for Number 7

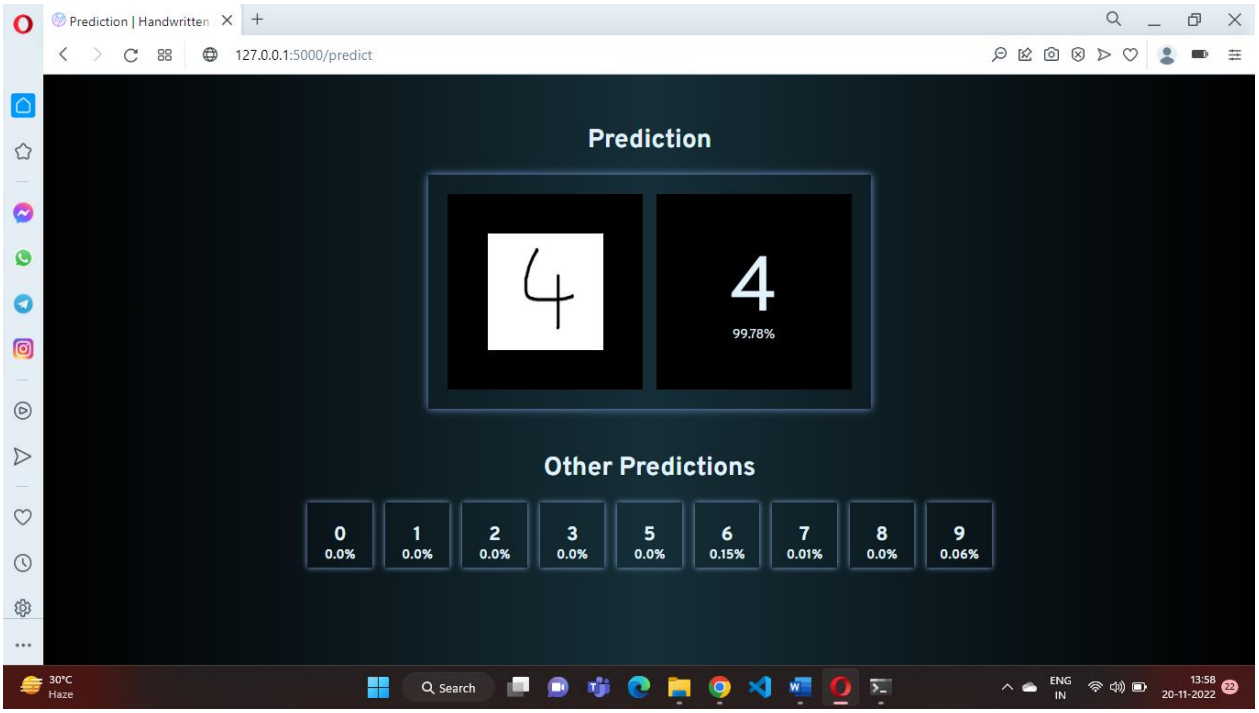


Figure 7.3.5 Prediction for Number 4

CHAPTER 8

TESTING

8.1 TEST CASES

UI testing is a testing type that helps testers ensure that all the fields, labels, buttons, and other items on the screen function as desired. It involves checking screens with controls, like toolbars, colors, fonts, sizes, icons, and others, and how these respond to the user input.

Table 8.1.1 Test Cases Report

				Date	3-Nov-22							
				Team ID	PNT2022TMID11500							
				Project Name	Handwritten Digit Recognition System							
				Maximum Marks	4 marks							
Test case ID	**Feature Type**	**Component**	**Test Scenario**	**Pre-Requisite**	**Steps To Execute**	**Test Data**	**Expected Result**	**Actual Result**	**Status**	**Commnets**	**TC for Automation(Y/N)**	**BUG ID**	**Executed By**
LoginPage_TC_OO1	Functional	Home Page	Verify user is able to see the Home page to upload the images	Not required	1.Click on Select file Button 2.Upload the image 3.Verify Select file button is displayed or not	https://127.0.0.1:5000	Should display the Webpage	Working as expected	Pass	Steps are clear to follow	N	NA	R.S.Bhuvaneshwari
LoginPage_TC_OO2	UI	Home Page	Verify the UI elements is Responsive	Not required	1.Click on Select file Button 2.Upload the image 3.Verify Select file button is displayed or not	https://127.0.0.1:5000	Webpage should be responsive	Working as expected	Pass	Steps are clear to follow	N	NA	N.L.Karthiga mai
LoginPage_TC_OO3	Functional	Home page	Verify whether the uploading image is a valid data	Not required	1.Check the input image is valid data or not. 2.Verify whether the result is displayed or not. 3.Back to the home page to predict next digit.	http://127.0.0.1:5000/predict	User should able to access the Webpage	Working as expected	Pass	Steps are clear to follow	N	NA	R.Nandhini
LoginPage_TC_OO4	Functional	Predict page	Verify user is able to see the predict result	Not required	1.Check the input image is valid data or not. 2.Verify whether the result is displayed or not. 3.Back to the home page to predict next digit.		Webpage should be responsive	Working as expected	Pass	Steps are clear to follow	N	NA	V.S.Chandra
LoginPage_TC_OO4	Functional	Predict page	Testing the website with various images	Not required	1.Check the input image is valid data or not. 2.Verify whether the result is displayed or not. 3.Back to the home page to predict next digit.		User should able to see the result	Working as expected	Pass	Steps are clear to follow	N	NA	K.B.Hemapriya

8.2 USER ACCEPTANCE TESTING

User Acceptance Testing (UAT), also known as beta or end-user testing, is defined as testing the software by the user or client to determine whether it can be accepted or not. This is the final testing performed once the functional, system and regression testing are completed.

1) Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the [A Novel method for Handwritten Digit Recognition System] project

at the time of the release to User Acceptance Testing (UAT).

2) Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved.

Table 8.2.1 Defect Analysis

Resolution	Severity1	Severity2	Severity3	Severity4	Subtotal
By Design	10	4	2	3	19
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	10	2	4	20	36
Not Reproduced	0	0	1	0	1
Skipped	0	0	0	0	0
Won't Fix	0	0	1	1	2
Totals	23	9	11	25	68

3) Test Case Analysis

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

Table 8.2.2 Test Case Analysis

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	10	0	0	10
Client Application	50	0	0	50
Security	5	0	0	5
Outsource Shipping	3	0	0	3
Exception Reporting	10	0	0	10
Final Report Output	4	0	0	4
Version Control	4	0	0	4

CHAPTER 9

RESULTS

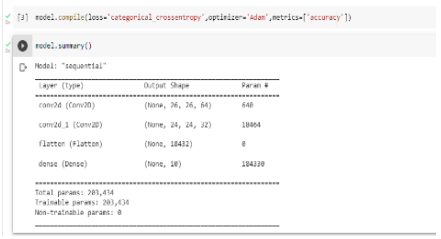
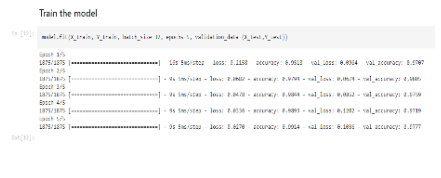
9.1 PERFORMANCE METRICS

Performance testing is a non-functional software testing technique that determines how the stability, speed, scalability, and responsiveness of an application holds up under a given workload.

Model Performance Testing:

This test will load/validate the behavior of the business components within the database. Two main requirements for this type of test are the number of transactions/ hours per user journey and appropriate think times.

Table 9.1.1 Performance Testing

S.No.	Parameter	Values	Screenshot
1.	Model Summary	Total params: 203,434 Trainable params: 203,434 Non-trainable params: 0	
2.	Accuracy	Training Accuracy - 0.9914 Validation Accuracy - 0.9777	

Model Summary

Keras provides a way to summarize a model. The summary is textual and includes information about: The layers and their order in the model. The output shape of each layer. The number of parameters (weights) in each layer.

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

model.summary()

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

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

 flatten (Flatten)           (None, 18432)             0

 dense (Dense)               (None, 10)                184330

=================================================================
Total params: 203,434
Trainable params: 203,434
Non-trainable params: 0
_________________________________________________________________
```

Figure 9.1.2 Summarize Model

Accuracy

Machine learning model accuracy is the measurement used to determine which model is best at identifying relationships and patterns between variables in a dataset based on the input, or training, data.

```
Train the model

In [12]: model.fit(X_train, Y_train, batch_size=32, epochs=5, validation_data=(X_test,Y_test))

Epoch 1/5
1875/1875 [==============================] - 16s 5ms/step - loss: 0.2158 - accuracy: 0.9518 - val_loss: 0.0964 - val_accuracy: 0.9707
Epoch 2/5
1875/1875 [==============================] - 9s 5ms/step - loss: 0.0682 - accuracy: 0.9794 - val_loss: 0.0674 - val_accuracy: 0.9805
Epoch 3/5
1875/1875 [==============================] - 9s 5ms/step - loss: 0.0478 - accuracy: 0.9844 - val_loss: 0.0852 - val_accuracy: 0.9759
Epoch 4/5
1875/1875 [==============================] - 9s 5ms/step - loss: 0.0336 - accuracy: 0.9893 - val_loss: 0.1202 - val_accuracy: 0.9719
Epoch 5/5
1875/1875 [==============================] - 9s 5ms/step - loss: 0.0270 - accuracy: 0.9914 - val_loss: 0.1036 - val_accuracy: 0.9777
Out[12]:
```

Figure 9.1.3 Training and Validation Accuracy

CHAPTER 10

ADVANTAGES & DISADVANTAGES

ADVANTAGES

- Since all the digits differs greatly, CNN is good in recognition and classification.
- High Level of accuracy while comparing to other algorithms and methodology.
- Fast in Classification Process.
- Useful in automation of bank check processing.
- Useful in postal mail sorting.
- Useful in form data entry etc.

DISADVANTAGES

- The main difficulty in the handwritten digits recognition different handwritten style which is a very personal behavior where there are a lot of models for numbers based on the angles, length of the segments, stress on some parts of numbers, etc.
- It has huge number of features, so the classifying process is challenging part.
- The Web server has some delay due to Python-Flask Environment was implemented.
- Difficulty in prediction, if the digits are connected or overlapped while writing.

CHAPTER 11

CONCLUSION

The project of the undertaking is just to create a model which can recognize the digits using MNIST datasets however it can be prolonged to letters and then a person’s handwriting. It can be used by countless organization, schools, banks and even for family activities. Handwritten digit focus will be beneficial for government bodies or any different organization to identify citizenship identification range which helps in automation. Likewise, license card quantity of any individual can be diagnosed thru this system. Similarly, it can be used for academic reason the place student can learn and recognize the real-world solution making use of this system. Similarly, postal addresses, bank cheque digit consciousness can be made less complicated thru automation the usage of this system.

CHAPTER 12

FUTURE SCOPE

Future studies might consider using the architecture of the convolution network which gave the best result on the MNIST database and the proposed recognition system is implemented on handwritten digits. Such more system can be designed for handwritten characters recognition, object recognition, image segmentation, handwriting recognition, text language recognition, and future studies also might consider on hardware implementation on online digit recognition system with more performance and efficiency with live results from live testing case scenarios.

CHAPTER 13

APPENDIX

GITHUB LINK:

https://github.com/IBM-EPBL/IBM-Project-30829-1660190740

PROJECT DEMO LINK:

https://youtu.be/-IsCfqiilDo