PROJECT REPORT

A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

Submitted by

PNT2022TMID53469

Team Leader :	Hemalatha G	2127190801028
Team member :	Deepika R	2127190801015
Team member :	Keerthiga D	2127190801037
Team member :	Narmadha R	2127190801049

Project Report Format

CHAPTER 1
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 recognise 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 2
LITERATURE SURVEY

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 margins since they vary from person to person. Additionally, there would be issues with identifying the numbers because of similarities between numerals like 1 and 7, 5 and 6, 3 and 8, 2 and 5, 2 and 7, etc. Finally, the individuality and variation of each individual's handwriting influence the structure and appearance of the digits.

2.2 REFERENCES

Improved Handwritten Digit Recognition Using Convolutional Neural Networks (CNN) (2020).

Ahlawat, Savita and Choudhary, Amit and Nayyar, Anand and Singh, Saurabh and Yoon, Byungun.

This paper's primary goal was to enhance handwritten digit recognition ability. To avoid difficult pre-processing, expensive feature extraction, and a complex ensemble (classifier combination) method of a standard recognition system, they examined different convolutional neural network variations. Their current work makes suggestions on the function of several hyper-parameters through thorough evaluation utilizing an MNIST dataset. They also confirmed that optimizing hyper-parameters is crucial for enhancing CNN architecture performance. With the Adam optimizer for the MNIST database, they were able to surpass many previously published results with a recognition rate of 99.89%. Through the trials, it is made abundantly evident how the performance of handwritten digit recognition is affected by the number of convolutional layers in CNN architecture. According to the paper, evolutionary algorithms can be explored for optimizing convolutional filter kernel sizes, CNN learning parameters, and the quantity of layers and learning rates.

An Efficient And Improved Scheme For Handwritten Digit Recognition Based On Convolutional Neural Network (2019)

Ali, Saqib and Shaukat, Zeeshan and Azeem, Muhammad and Sakhawat, Zareen and Mahmood, Tariq and others.

This study uses rectified linear units (ReLU) activation and a convolutional neural network (CNN) that incorporates the Deeplearning4j (DL4J) architecture to recognize handwritten digits. The proposed CNN framework has all the necessary parameters for a high level of MNIST digit classification accuracy. The system's training takes into account the time factor as well. The system is also tested by altering the number of CNN layers for additional accuracy verification. It is important to note that the CNN architecture consists of two convolutional layers, the first with 32 filters and a 5x5 window size and the second with 64 filters and a 7x7 window size. In comparison to earlier proposed systems, the experimental findings show that the proposed CNN architecture for the MNIST dataset demonstrates great performance in terms of time and accuracy. As a result, handwritten numbers are detected with a recognition rate of 99.89% and high precision (99.21%) in a short amount of time.

Improved Handwritten Digit Recognition Using Quantum K-Nearest Neighbor Algorithm (2019)

Wang, Yuxiang and Wang, Ruijin and Li, Dongfen and Adu-Gyamfi, Daniel and Tian, Kaibin and Zhu, Yixin.

The KNN classical machine learning technique is used in this research to enable quantum parallel computing and superposition. They used the KNN algorithm with quantum acceleration to enhance handwritten digit recognition. When dealing with more complicated and sizable handwritten digital data sets, their suggested method considerably lowered the computational time complexity of the traditional KNN algorithm. The paper offered a theoretical investigation of how quantum concepts can be applied to machine learning. Finally, they established a fundamental operational concept and procedure for machine learning with quantum acceleration.

Handwritten Digit Recognition Using Machine And Deep Learning Algorithms (2021)

Pashine, Samay and Dixit, Ritik and Kushwah, Rishika.

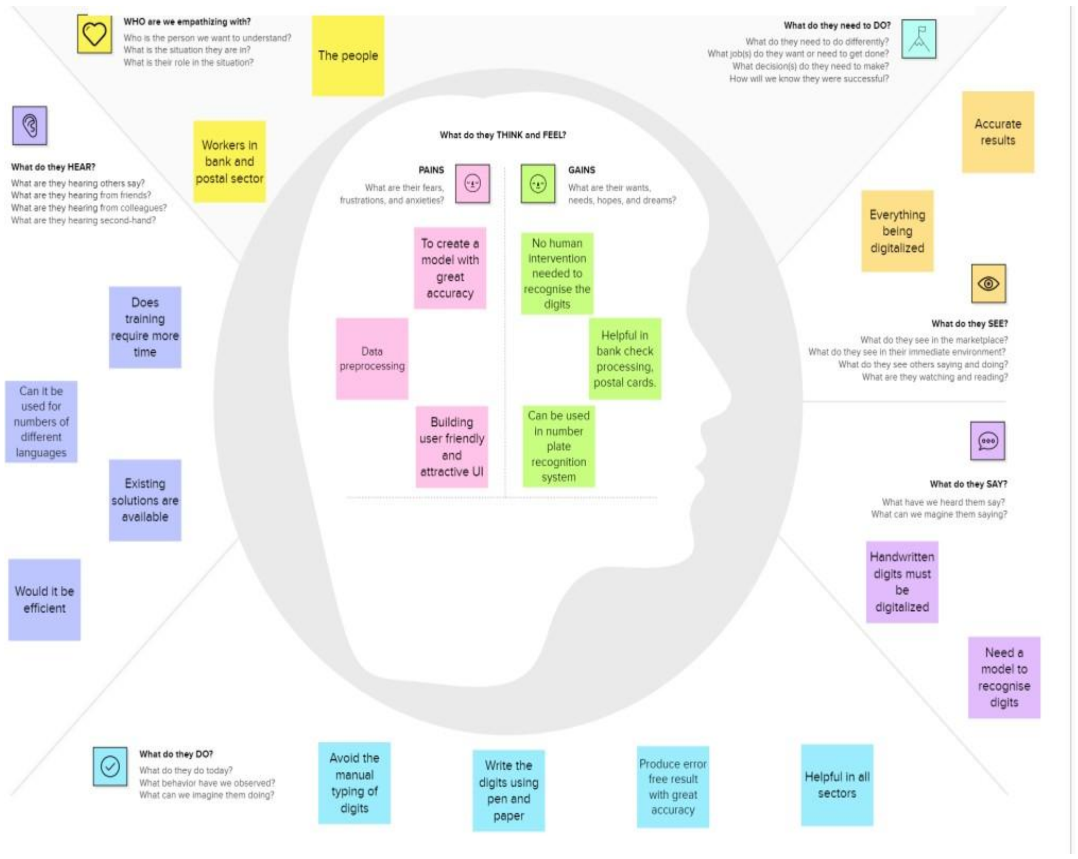
In this study, they developed three deep and machine learning-based models for handwritten digit recognition using MNIST datasets. To determine which model was the most accurate, they compared them based on their individual properties. Support vector machines are among the simplest classifiers, making them faster than other algorithms and providing the highest training accuracy rate in this situation. However, due to their simplicity, SVMs cannot categorize complicated and ambiguous images as accurately as MLP and CNN algorithms can. In their research, they discovered that CNN produced the most precise outcomes for handwritten digit recognition. This led them to the conclusion that CNN is the most effective solution for all types of prediction issues, including those using picture data. Next, by comparing the execution times of the algorithms, they determined that increasing the number of epochs without changing the configuration of the algorithm is pointless due to the limitation of a certain model, and they discovered that beyond a certain number of epochs, the model begins over-fitting the dataset and provides biased predictions.

2.3 PROBLEM STATEMENT DEFINITION

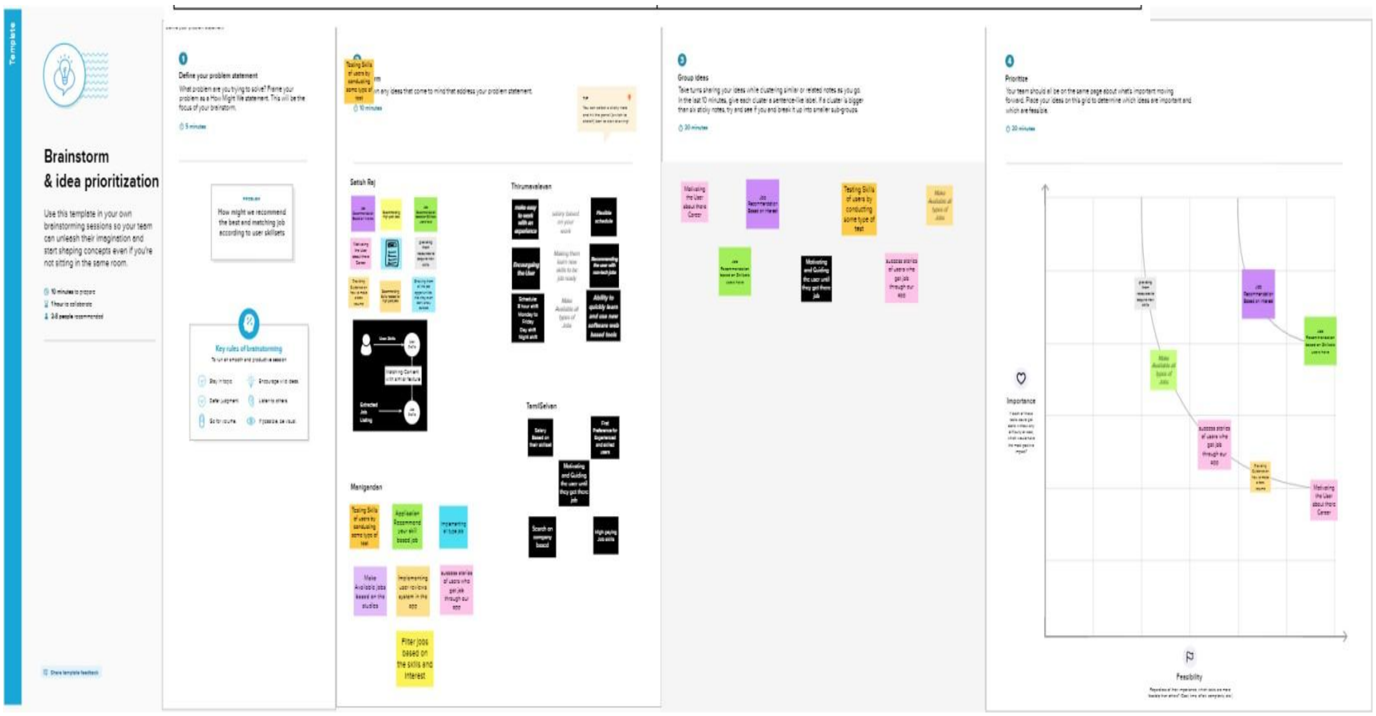
For years, the traffic department has been combating traffic law violators. These offenders endanger not only their own lives, but also the lives of other individuals. Punishing these offenders is critical to ensuring that others do not become like them. Identification of these offenders is next to impossible 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 violations as a result.

CHAPTER 3
IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS


WHO are we empathizing with?
Who is the person we want to understand?
What is the situation they are in?
What is their role in the situation?
The people
Workers in bank and postal sector
What do they need to DO?
What do they need to do differently?
What job(s) do they want or need to get done?
What decision(s) do they need to make?
How will we know they were successful?
Accurate results
Everything being digitalized
What do they HEAR?
What are they hearing others say?
What are they hearing from friends?
What are they hearing from colleagues?
What are they hearing second-hand?
Does training require more time
Can it be used for numbers of different languages
Existing solutions are available
Would it be efficient
What do they THINK and FEEL?
PAINS
What are their fears, frustrations, and anxieties?
GAINS
What are their wants, needs, hopes, and dreams?
To create a model with great accuracy
Data preprocessing
Building user friendly and attractive UI
No human intervention needed to recognise the digits
Helpful in bank check processing, postal cards.
Can be used in number plate recognition system
What do they SEE?
What do they see in the marketplace?
What do they see in their immediate environment?
What do they see others saying and doing?
What are they watching and reading?
What do they SAY?
What have we heard them say?
What can we imagine them saying?
Handwritten digits must be digitalized
Need a model to recognise digits
What do they DO?
What do they do today?
What behavior have we observed?
What can we imagine them doing?
Avoid the manual typing of digits
Write the digits using pen and paper
Produce error free result with great accuracy
Helpful in all sectors

3.2 IDEATION & BRAINSTORMING



Template

Brainstorm & idea prioritization

Use this template in your own 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
3-8 people recommended

Share template feedback

1
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 might we recommend the best and matching job according to user skillsets

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
Testing Skills of users by conducting some type of test
...m
...on any ideas that come to mind that address your problem statement.
10 minutes

Satish Raj

User Skills
Matching Content with similar feature
Extracted Job Listing

Thirumavalavan
make easy to work with an experience
salary based on your work
Flexible schedule
Encouraging the User
Making them learn new skills to be job ready
Recommending the user with non-tech jobs
Schedule 8 hour shift Monday to Friday Day shift Night shift
Make Available all types of Jobs
Ability to quickly learn and use new software web based tools

TamilSelvan
Salary Based on their skillset
First Preference for Experienced and skilled users
Motivating and Guiding the user until they got there job
Search on company based
High paying Job skills

Manigandan
Testing Skills of users by conducting some type of test
Application Recommend your skill based job
Implementing all type job
Make Available jobs based on the studies
Implementing user reviews system in the app
success stories of users who got job through our app
Filter jobs based on the skills and interest

3
Group ideas
Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10 minutes, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.
20 minutes

Motivating the User about there Career
Job Recommendation Based on Interest
Testing Skills of users by conducting some type of test
Make Available all types of Jobs
Job Recommendation based on Skillsets users have
Motivating and Guiding the user until they got there job
success stories of users who got job through our app

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

Importance
Feasibility

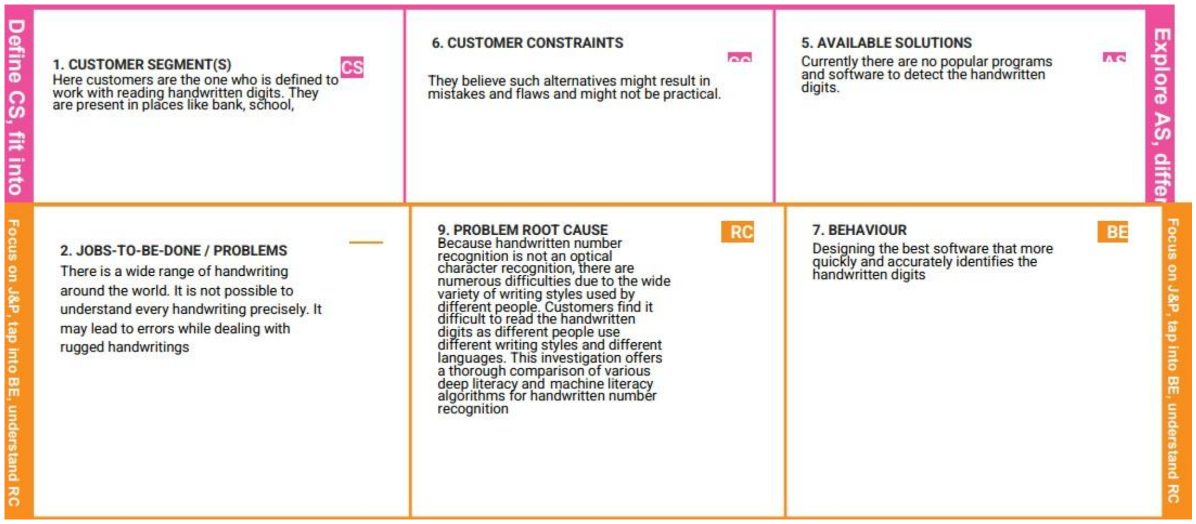
Job Recommendation Based on Interest
Job Recommendation based on Skillsets users have
Make Available all types of Jobs
success stories of users who got job through our app
Motivating the User about there Career

3.3 PROPOSED SOLUTION

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Manually written digits are of a different size, thickness, position and direction. In this manner, various difficulties must be considered to determine the issue of handwritten digit recognition.
2.	Idea / Solution description	To solve this problem, we are going to implement a classification algorithm to recognize the handwritten digits. This algorithm will be effective in order to recognize digits which are of different compositions.
3.	Novelty / Uniqueness	*Strategy for perceiving and arranging transcribed digits. *Can be used offline *Provided more data sets for more accuracy * The uniqueness and assortment in the composition styles of various individuals additionally influence the example and presence of the digits.
4.	Social Impact / Customer Satisfaction	The main social impact of this work is to ensure effective and reliable approaches for recognition of handwritten digits and make banking operations easier and error free. Customers will feel at ease, as it is easy and convenient to use. As the accuracy is acceptable, this can have many applications.

5.	Business Model (Revenue Model)	This novel method for Handwritten Digit Recognition System can be approached by many industries which needs this application including, programmed bank checks, postal locations and tax documents and so on. Humans recognizing the handwritten digits with their naked eye can be difficult at times as it of different sizes, thickness, direction and ca also lead to making errors due to these factors. This is when our proposed solutions comes into help. We provide different data sets which helps in recognition with accuracy, so that human making errors can be avoided respectively.
6.	Scalability of the Solution	Financial and other business organizations such as banks are facing issues in Recognizing written digits such as in cheques etc. This can be handled by our handwritten digit recognition project as they expand into different business domains without impacting performance. Our proposed solution is scalable as it is dynamic and also trained using AI and deep learning Models.

3.4 PROBLEM SOLUTION FIT



Define CS, fit into

1. CUSTOMER SEGMENT(S) CS
Here customers are the one who is defined to work with reading handwritten digits. They are present in places like bank, school,

6. CUSTOMER CONSTRAINTS
They believe such alternatives might result in mistakes and flaws and might not be practical.

5. AVAILABLE SOLUTIONS
Currently there are no popular programs and software to detect the handwritten digits.

Explore AS, differ

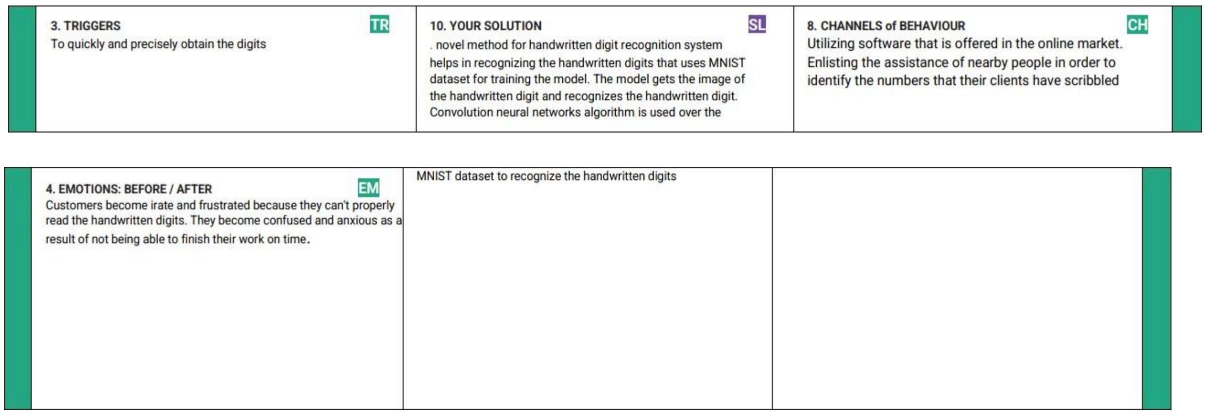
Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS
There is a wide range of handwriting around the world. It is not possible to understand every handwriting precisely. It may lead to errors while dealing with rugged handwritings

9. PROBLEM ROOT CAUSE RC
Because handwritten number recognition is not an optical character recognition, there are numerous difficulties due to the wide variety of writing styles used by different people. Customers find it difficult to read the handwritten digits as different people use different writing styles and different languages. This investigation offers a thorough comparison of various deep literacy and machine literacy algorithms for handwritten number recognition

7. BEHAVIOUR BE
Designing the best software that more quickly and accurately identifies the handwritten digits

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



3. TRIGGERS TR
To quickly and precisely obtain the digits

10. YOUR SOLUTION SL
. novel method for handwritten digit recognition system helps in recognizing the handwritten digits that uses MNIST dataset for training the model. The model gets the image of the handwritten digit and recognizes the handwritten digit. Convolution neural networks algorithm is used over the MNIST dataset to recognize the handwritten digits

8. CHANNELS of BEHAVIOUR CH
Utilizing software that is offered in the online market. Enlisting the assistance of nearby people in order to identify the numbers that their clients have scribbled

4. EMOTIONS: BEFORE / AFTER EM
Customers become irate and frustrated because they can't properly read the handwritten digits. They become confused and anxious as a result of not being able to finish their work on time.

CHAPTER 4
REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

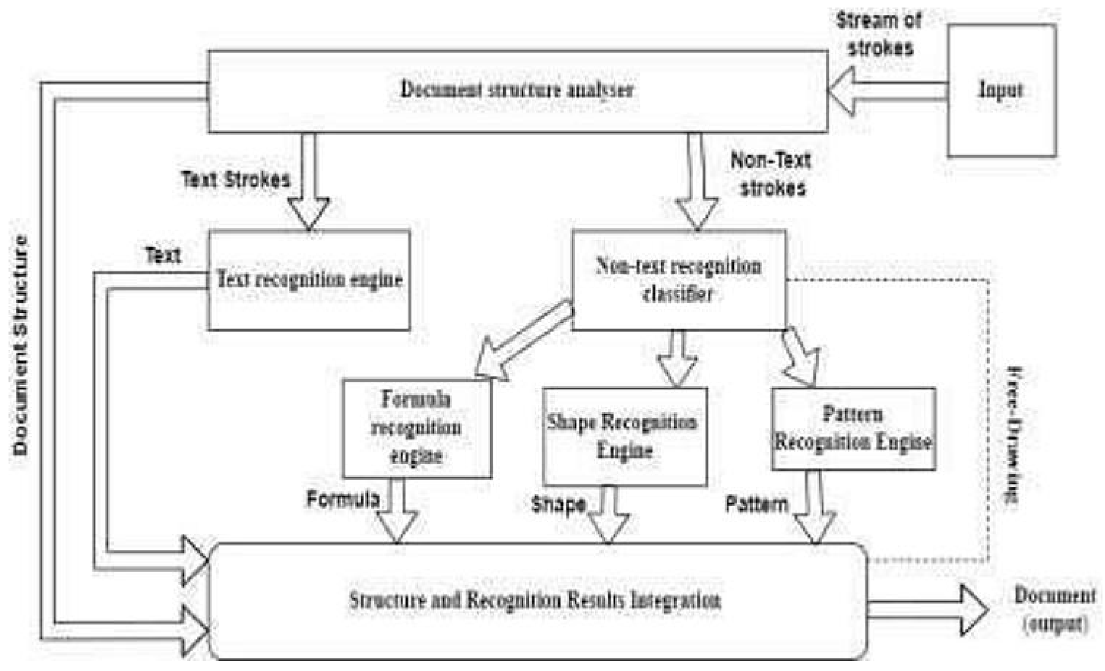
FR No.	**Sub Requirement (Story / Sub-Task)**
FR-1	Image Data: Handwritten digit recognition refers to a computer's capacity to identify human handwritten digits from a variety of sources, such as photographs, documents, touch screens, etc., and categorise them into ten established classifications (0-9). In the realm of deep learning, this has been the subject of countless studies
FR-2	Website: Web hosting makes the code, graphics, and other items that make up a website accessible online. A server hosts every website you've ever visited. The type of hosting determines how much space is allotted to a website on a server. Shared, dedicated, VPS, and reseller hosting are the four basic varieties.
FR-3	Digit Classifier Model: To train a convolutional network to predict the digit from an image, use the MNIST database of handwritten digits. get the training and validation data first.
FR-4	Cloud: The cloud offers a range of IT services, including virtual storage, networking, servers, databases, and applications. In plain English, cloud computing is described as a virtual platform that enables unlimited storage and access to your data over the internet.
FR-5	Modified National Institute of Standards and Technology dataset: The abbreviation MNIST stands for the MNIST dataset. It is a collection of 60,000 tiny square grayscale photographs, each measuring 28 by 28, comprising handwritten single digits between 0 and 9.

4.2 NON FUNCTIONAL REQUIREMENT

FR No.	**Non-Functional Requirement**	**Description**
NFR-1	**Usability**	One of the very significant problems in pattern recognition applications is the recognition of handwritten characters. Applications for digit recognition include filling out forms, processing bank checks, and sorting mail.
NFR-2	**Security**	The system generates a thorough description of the instantiation parameters, which might reveal information like the writing style, in addition to a categorization of the digit. The generative models are capable of segmentation driven by recognition. The procedure uses a relatively.
NFR-3	**Reliability**	The samples are used by the neural network to automatically deduce rules for reading handwritten digits. Furthermore, the network may learn more about handwriting and hence enhance its accuracy by increasing the quantity of training instances. Numerous techniques and algorithms, such as Deep Learning/CNN, SVM, Gaussian Naive Bayes, KNN, Decision Trees, Random Forests, etc., can be used to recognise handwritten numbers
NFR-4	**Performance**	With typed text in high -quality photos, optical character recognition (OCR) technology offers accuracy rates of greater than 99%. However, variances in spacing, abnormalities in handwriting, and the variety of human writing styles result in less precise character identification.
NFR-5	**Availability**	Access to information is restricted to each other.
NFR-6	**Scalability**	The task of handwritten digit recognition, using a classifier, has great importance and use such as – online handwriting recognition on computer tablets, recognize zip codes on mail for postal mail sorting, processing bank check amounts, numeric entries in forms filled up by hand.

CHAPTER 5
PROJECT DESIGN

5.1 DATA FLOW DIAGRAM

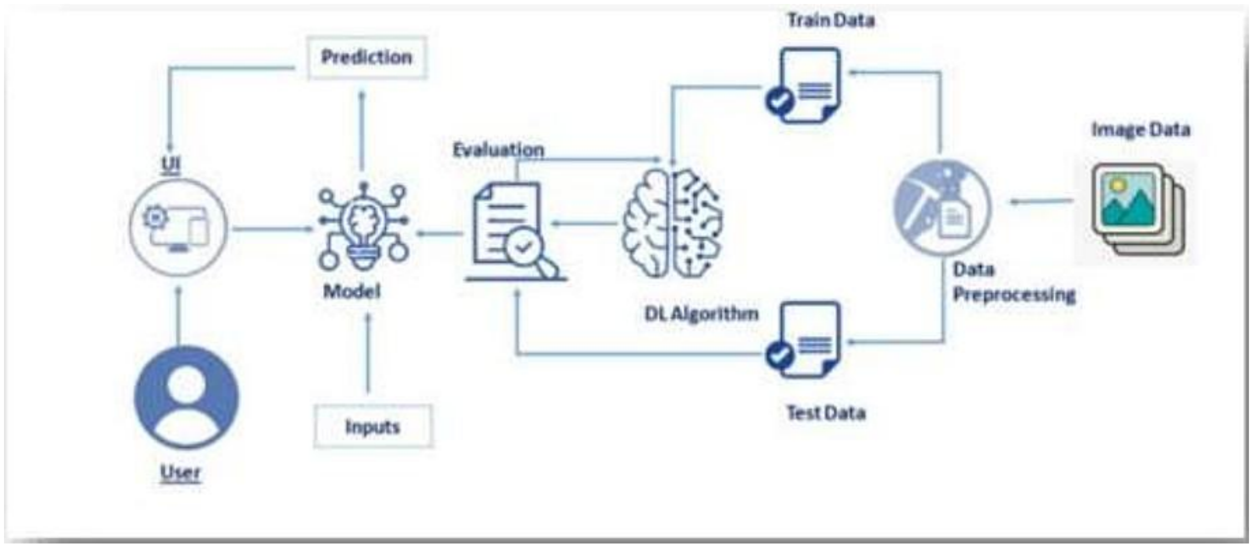


5.2 SOLUTION & TECHNICAL ARCHITECTURE

5.2.1 SOLUTION ARCHITECTURE

PROJECT DESCRIPTION :

Everyone in the world has a unique writing style, handwriting identification is one of the fascinating research projects now being conducted. It is the ability of a computer to automatically recognize and comprehend handwritten numbers or letters. Every aspect of life is being digitized to lessen the need for human labor as a result of advancements in science and technology. Thus, handwritten digit recognition is required in many real-time applications. The MNIST data collection, which contains 70000 handwritten digits, is frequently utilized for this recognition method. In order to train these photos and create a deep learning model, we use artificial neural networks. A web application is developed that allows users to upload pictures of handwritten numbers. The model examines this image and the detected result is returned to the UI.



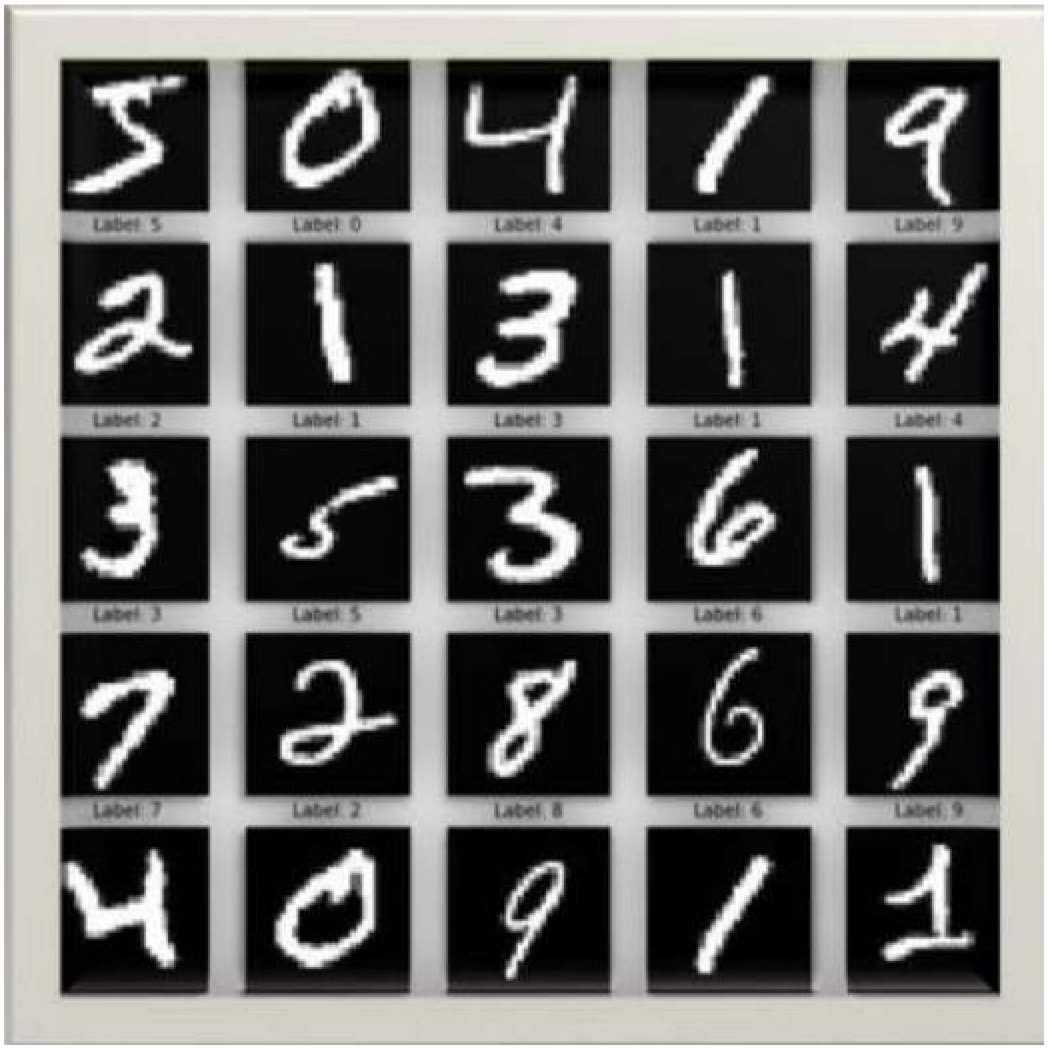
SOLUTION:

MNIST Dataset Description :

Because everyone in the world has a unique writing style, handwriting identification is one of the fascinating research projects now being conducted. It is the ability of a computer to automatically recognise and comprehend handwritten numbers or letters. Every aspect of life is being digitized to lessen the need for human labor as a result of advancements in science and technology. Thus, handwritten digit recognition is required in many real-time applications. The MNIST data collection, which contains 70000 handwritten digits, is frequently utilized for this recognition method. In order to train these photos and create a deep learning model, we use artificial neural networks. A web application is developed that allows users to upload pictures of handwritten numbers. The model examines this picture. The 60,000 training and 10,000 testing labeled handwritten digit images in the MNIST Handwritten Digit Recognition Dataset. Each image has a total of 784 (2828) pixels, or 28 pixels in height and 28 pixels in width. There is just one pixel value assigned to each pixel. It displays the brightness or darkness of that pixel (larger numbers indicate darker pixel). The integer for this pixel value ranges from 0 to 255.

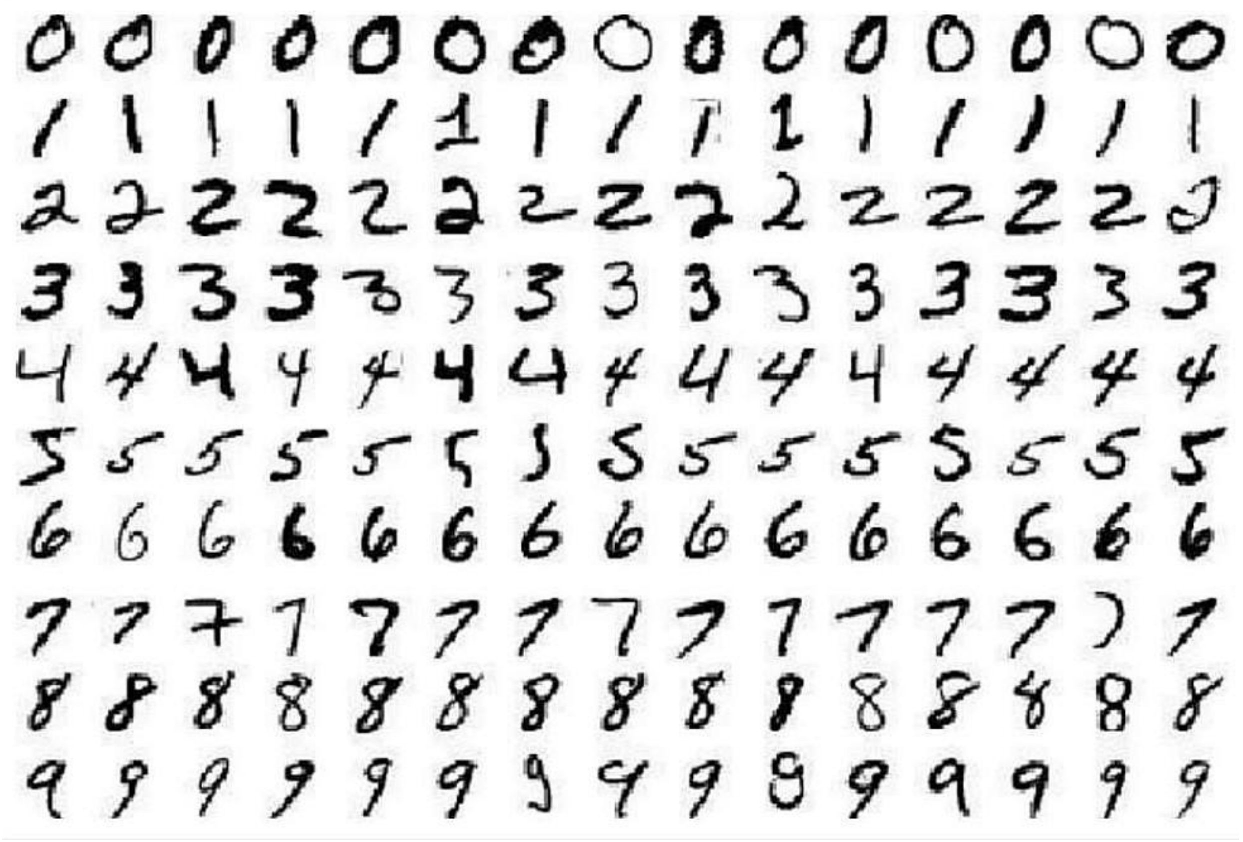
PROCEDURE:

1. Install the TensorFlow library.
2. Prepare the model's dataset.
3. For the purposeof classifying the handwritten digits, create a single layerperceptron model.
4. Plot the accuracy change over time. Analyze the test data to evaluate the model.



● Speculate on the model summary.
● To make the model a multi-layer perceptron, add a hidden layer.
● To avoid overfitting and assess its impact on accuracy, include Dropout.
● Increase the number of hidden layer neurons and assess how accuracy is affected.
● Test the impact of various optimizers on accuracy.
● Increase the hidden layers and assess the accuracy impact.
● Change the batch size and epochs, then assess the impact on accuracy. For the recognition of handwritten digits, the MNIST dataset is frequently used. 10,000

test photos make up the dataset, which includes 60,000 training images. The discipline of image processing relies heavily on artificial neural networks because they most closely resemble the human brain. A significant project done with the use of neural networks is the recognition of handwritten digits using the MNIST dataset. In essence, it recognizes digits that were scribbled and scanned. Our handwritten digit identification system goes a step further in that it can now recognize handwritten numbers typed directly on the screen with the aid of an integrated GUI in addition to detecting them in scanned photos.



APPROACH :

This project will be approached utilizing a three-layered neural network.

● The input layer: The input layer transfers the information from our example systems to the following layer so that the latter can compute its activations.

● The hidden layer: The network's nonlinear ties are provided by hidden units termed activations that make up the hidden layer. Depending on our needs, there

can be a variety of concealed layers.
● The output layer: The nodes in this stratum are referred to as output units. It gives us access to the neural network's final prediction, which may be used to make final predictions. A neural network is a model of the brain's operations. It is made up of numerous layers with a variety of activations; these activations mimic the neurons in our brain. An attempt is made by a neural network to learn a set of parameters from a set of data that might aid in understanding the underlying relationships. Since neural networks are capable of adapting to changing input, the network can produce the best outcome without having to change the output criterion.

.

METHODOLOGY :

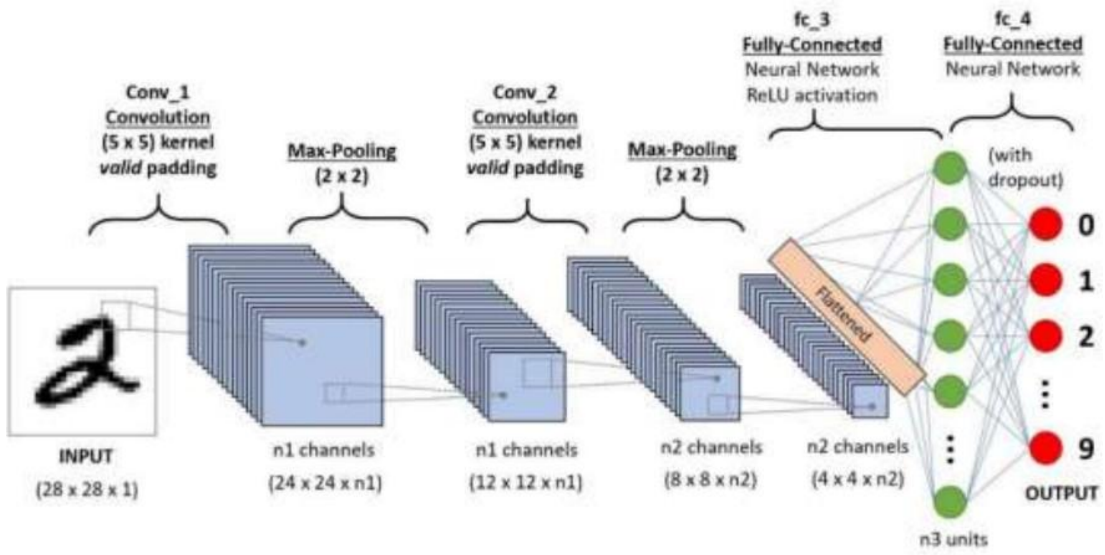
A neural network with one hidden layer and 100 activation units has been put into practice (excluding bias units). The features (X) and labels (Y) were retrieved after the data was loaded from a.mat file. To prevent overflow during computation, features are then scaled into a range of [0,1] by dividing by 255. 10,000 testing cases and 60,000 training examples make up the data. With the training data, feedforward is used to calculate the hypothesis, and backpropagation is then used to lower the error between the layers. To combat overfitting, the regularization parameter lambda is set to 0.1. To identify the model that fits the situation the optimizer runs for 70 times.

ALGORITHM :

Forward Propagation Architecture : A simple process for the CNN module's feature extraction and picture classification is shown below. The network's input layer, hidden layers, and output layer are all displayed in the architecture. The feature extraction phase of the network involves multiple layers and uses convolution and subsampling.

EXPLANATION FOR THE PROPOSED SYSTEM

1. The User layer is the top layer of the architecture. The users who engage with the programme andget the desired outcomesmake up the user layer.
2. The frontend architecture of the application is made upof the following three levels.
3. The application will be created on the open-source JavaScript, CSS, and HTML platform.
4. The localhost, which is displayed in the browser, is where the programme is deployed. The userwill be ableto upload images of the handwritten numbers to the app to have them digitized.
5. The business layer, which consists of logical calculations based on the client's request, sitsbetween the database and view layers.The service interfaceis also included.



6. Training Data and Test Data make up the backend layer's two datasets. The

training set, which consists of 60,000 cases, and the test set, which consists of 10,000 examples, have already been separated into the MNIST database.

7. A convolution neural network is utilized as the training algorithm. This will get the trained model ready to classify the data.

WORKING :

❖ After receiving an input, neural networks change it using a number of hidden layers.
❖ Each group of neurons in a hidden layer is completely linked to every other neuron in the layer above it.
❖ One layer of neurons have perfect independence from one another.
❖ The "output layer" is the final layer to be fully connected.

CONVOLUTION LAYER :

The foundational component of a CNN is the convolutional layer. The parameters of the layer are a set of learnable filters (or kernels) that cover the entire depth of the input volume but have a narrowreceptive field.Each filter is convolved acrossthe width and heightof the input volume during the forward pass, computing the dot product between each filter entry and the input to create a two- dimensional activation map of the filter. As a result, the network picks up filters that turn on when itdetects a certain kind of featureat a particular spatiallocation in the input.

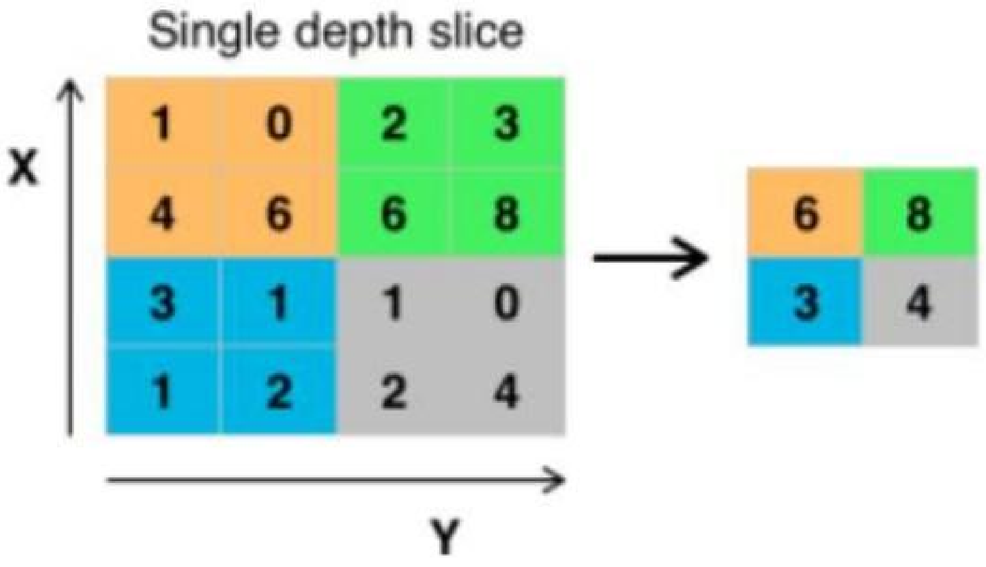
FEATURE EXTRACTION :

All neurons in a feature share the same weights .In this way all neurons detect the same feature atdifferent positions in the input image. Reduce the number of free parameters.

SUBSAMPLING LAYER :

Subsampling, or down sampling, refers to reducing the overall size of a signal. The subsampling layers reduce the spatial resolution of each feature map. Reduce the effect of noises and shift ordistortion invariant is achieved.

POOLING LAYER :



It is common to periodically insert a Pooling layer in-between successive Conv layer in a Convent architecture. Its function is to progressively reduce the spatial size of the representation to reducethe number of parameters and computation in the network, and hence to also control overfitting. The Pooling Layer operates independently on every depth slice of the input and resizes it spatially,usingthe MAX operation.

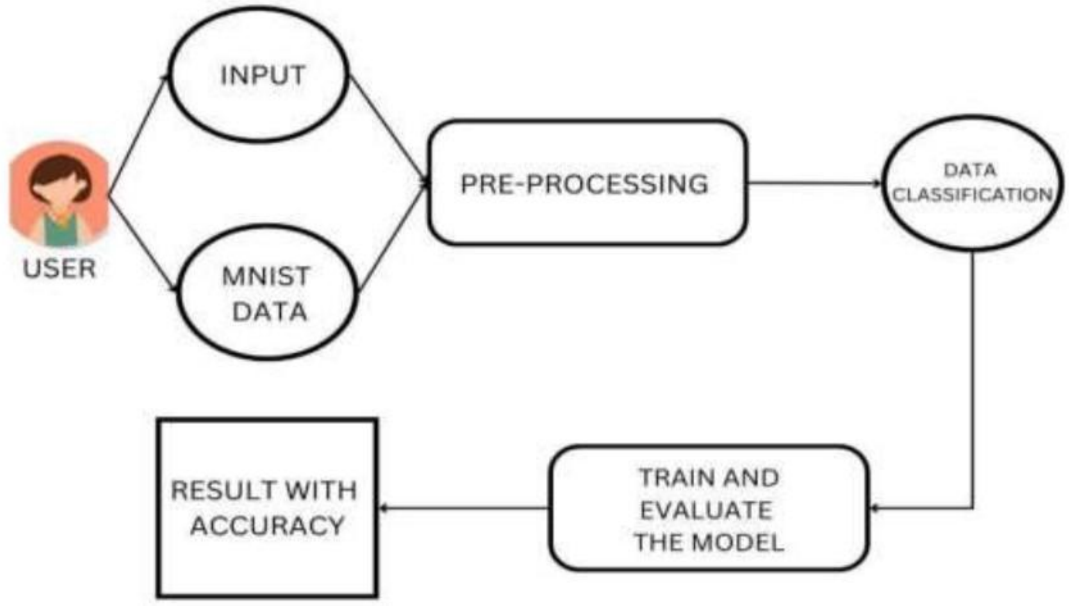
TENSORFLOW :

An open-source machine learning library for both research and production is called TensorFlow. TensorFlow provides developers of all skill levels with APIs for desktop, mobile, web, and cloud applications. To get started, refer to the sections below. We can achieve text output and sound output by scanning the number digit and converting it to PNG format using the python3 command inthe terminal.

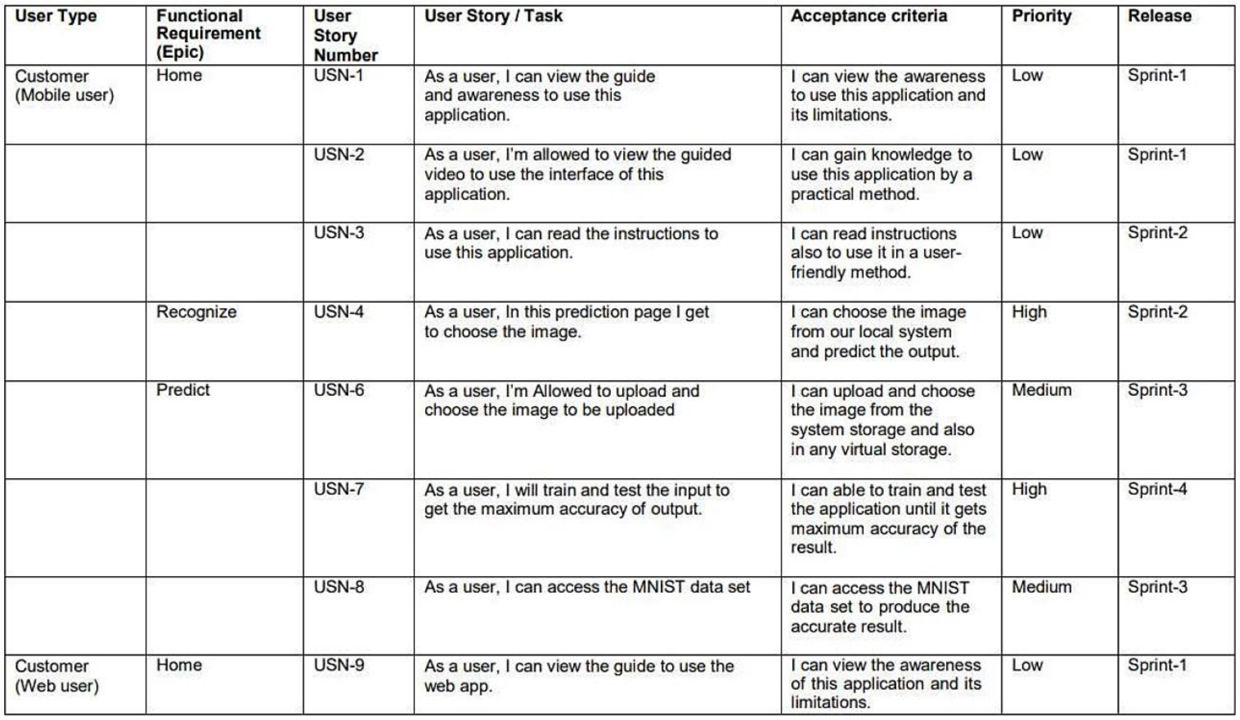
RESULT:

As with any study or project conducted in the fields of machine learning and image processing, wedo not consider our results to be perfect. Because machine learning is a field that is constantly evolving, there is always room for improvement in your approaches. There will always be a brand new concept that more successfully addresses a certain issue. The application was evaluated usingthree models: Multi-Layer Perceptron (MLP), Convolution Neural Network, and (CNN). With each model, we get a different classifier accuracy, showingwhich is better

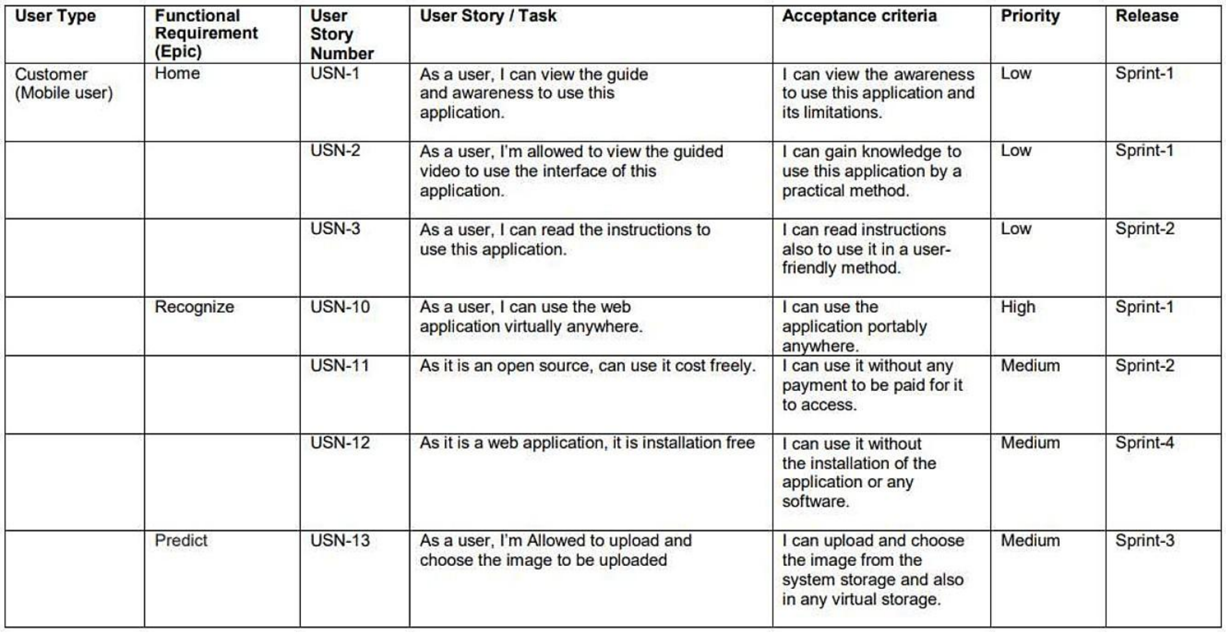
5.2.2 TECHNICAL ARCHITECTURE



5.3 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 also 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 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)	Home	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
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-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'm Allowed to upload and choose the image to be uploaded	I can upload and choose the image from the system storage and also in any virtual storage.	Medium	Sprint-3

CHAPTER 6
PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNING AND ESTIMATION

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data Collection &pre processing	USN-1	As a user, I can upload any kind of image with thepre-processing step is involved in it.	10	High	Keerthiga D Deepika R
Sprint-1		USN-2	As a user, I can uploadthe image in anyresolution	10	High	Keerthiga D Deepika R
Sprint-2	Building the Machine learning model	USN-3	As a user, I will get a application with MLmodel which provides high accuracy of recognized handwritten digit	5	Medium	Hemalatha G Narmadha R

Sprint	**Functional Requirement (Epic)**	**User Story Number**	**User Story / Task**	**Story Points**	**Priority**	**Team Members**
Sprint-2		USN-4	As a user, I can passthe handwritten digitimage for recognizing the digit.	5	Medium	Deepika R Hemalatha G
Sprint-2		USN-5	As a user, I can get the most suitable recognized digit.	10	High	Jaswanth N Narmadha R
Sprint-3	Building User Interface Application	USN-6	As a user, I will upload the handwritten digit image to the application by clicking a upload button.	6	Medium	Deepika R Hemalatha G
Sprint-3		USN-7	As a user, I can knowthe details of thefundament al usage of the application.	4	Medium	Deepika R Hemalatha G Narmadha R
Sprint-3		USN-8	As a user, I can see the predicted / recognized digits in the application.	10	High	Keerthiga DDeepika R Hemalatha G
Sprint-4	Train and deployme nt ofmodel in IBM Cloud.	USN-9	As a user, I can access the webapplicationand make the use of the product from anywhere	20	High	Keerthiga DDeepika R Hemalatha G Narmadha R

6.2 SPRINT DELIVERY SCHEDULE

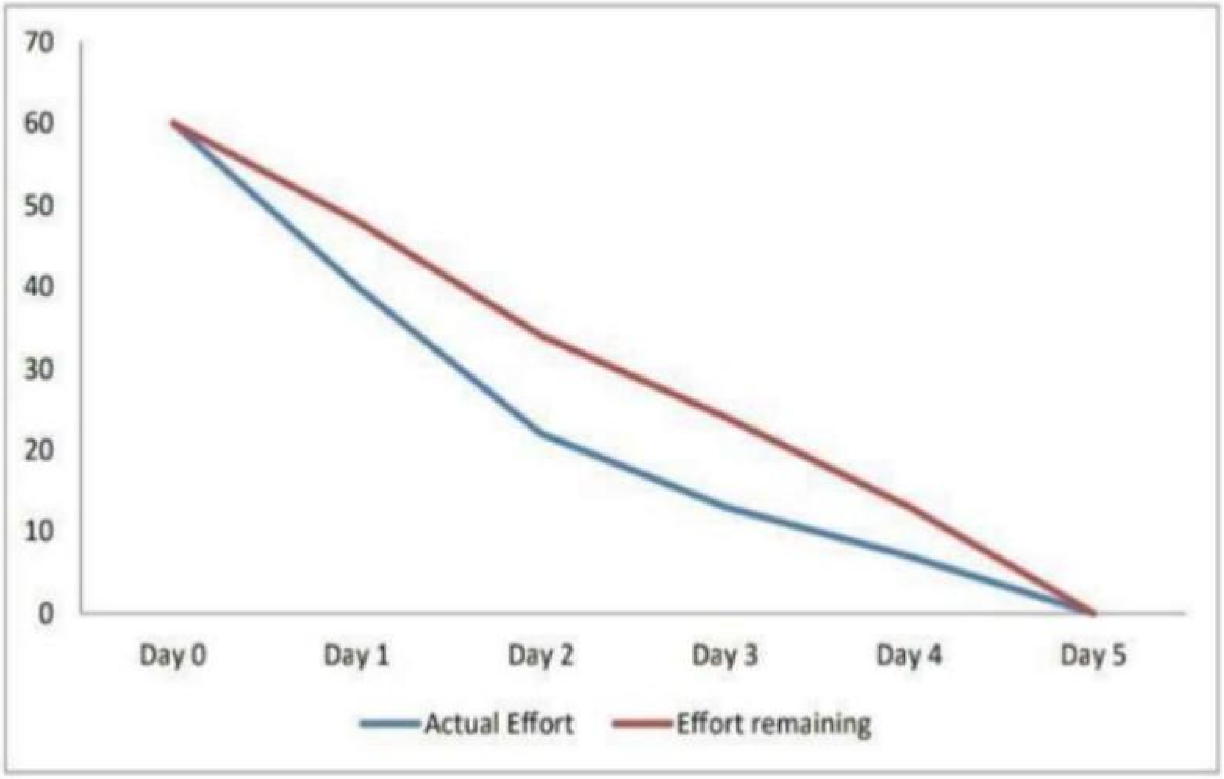
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	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 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let us calculate the team's average velocity (AV) per iteration unit (story points per day)

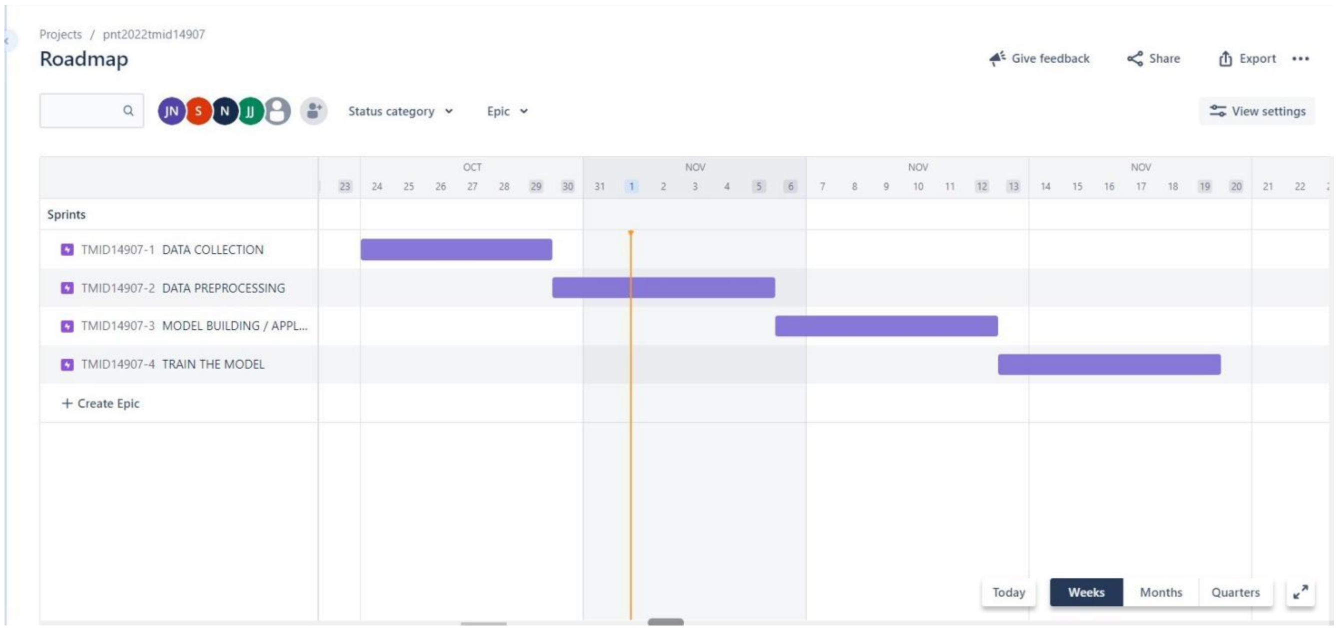
AV=Sprint Duration / Velocity
20/6 = 3.33

Burndown Chart:



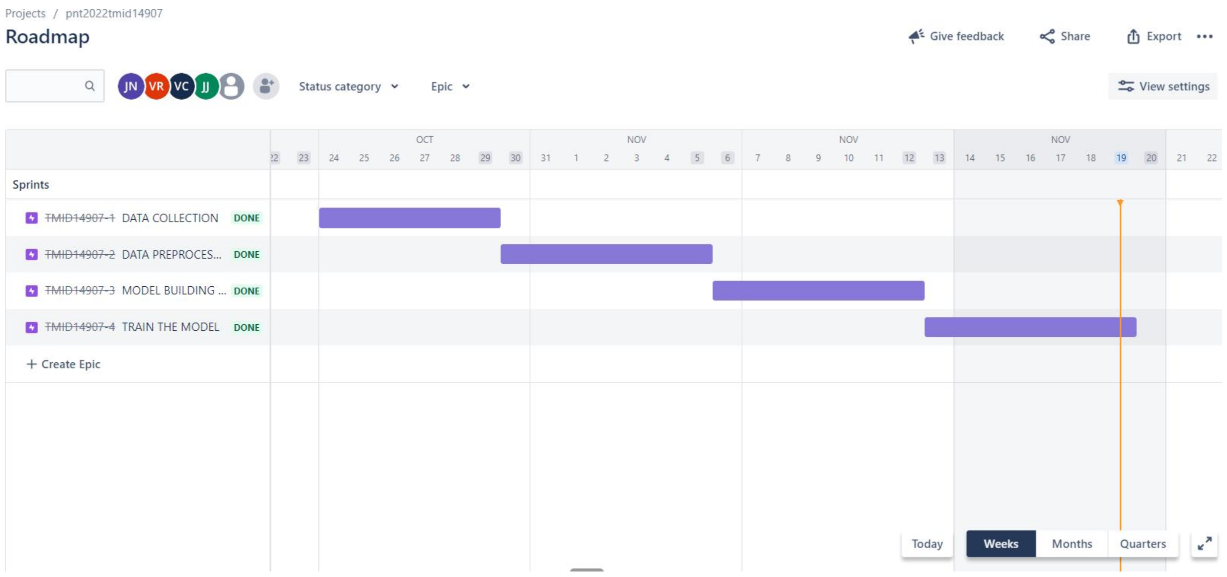
6.3 Reports from JIRA

IMAGE 1:



TMID14907-1 DATA COLLECTION
TMID14907-2 DATA PREPROCESSING
TMID14907-3 MODEL BUILDING / APPL...
TMID14907-4 TRAIN THE MODEL
+ Create Epic

IMAGE 2:



~~TMID14907-1~~ DATA COLLECTION DONE
~~TMID14907-2~~ DATA PREPROCES... DONE
~~TMID14907-3~~ MODEL BUILDING ... DONE
~~TMID14907-4~~ TRAIN THE MODEL DONE
+ Create Epic

CHAPTER 7
CODING & SOLUTIONING

```
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: int) -> str:
	"""
	Generates a random file name.
	Args:
		n (int): Length the of the file name.
	Returns:
		str: The file name.
	"""
	return ''.join(random.choices(string.ascii_uppercase + string.digits, k=n))

def recognize(image: bytes) -> tuple:
	"""
	Predicts the digit in the image.
	Args:
		image (bytes): The image data.
	Returns:
		tuple: The best prediction, other predictions and file name
	"""

	model=load_model(Path("./model/mnistcnn.h5"))

	img = Image.open(image).convert("L")
```

```
# Generate a random name to save the image file.
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}"))

# Convert the Image to Grayscale, Invert it and Resize to get better prediction.
img = ImageOps.grayscale(img)
img = ImageOps.invert(img)
img = img.resize((28, 28))

# Convert the image to an array and reshape the data to make prediction.
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]

# Get all the predictions and it's respective accuracy.
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))

# Get the value with the highest accuracy
best = others.pop(best)

return best, others, img_name
```

7.1 FEATURE 1

Recognition:

i) User can give input as images.

ii) By this digits can be easily recognized and it can be in JPEG or PNG format.

iii) Data accuracy is fully protected.

7.2 FEATURE 2:

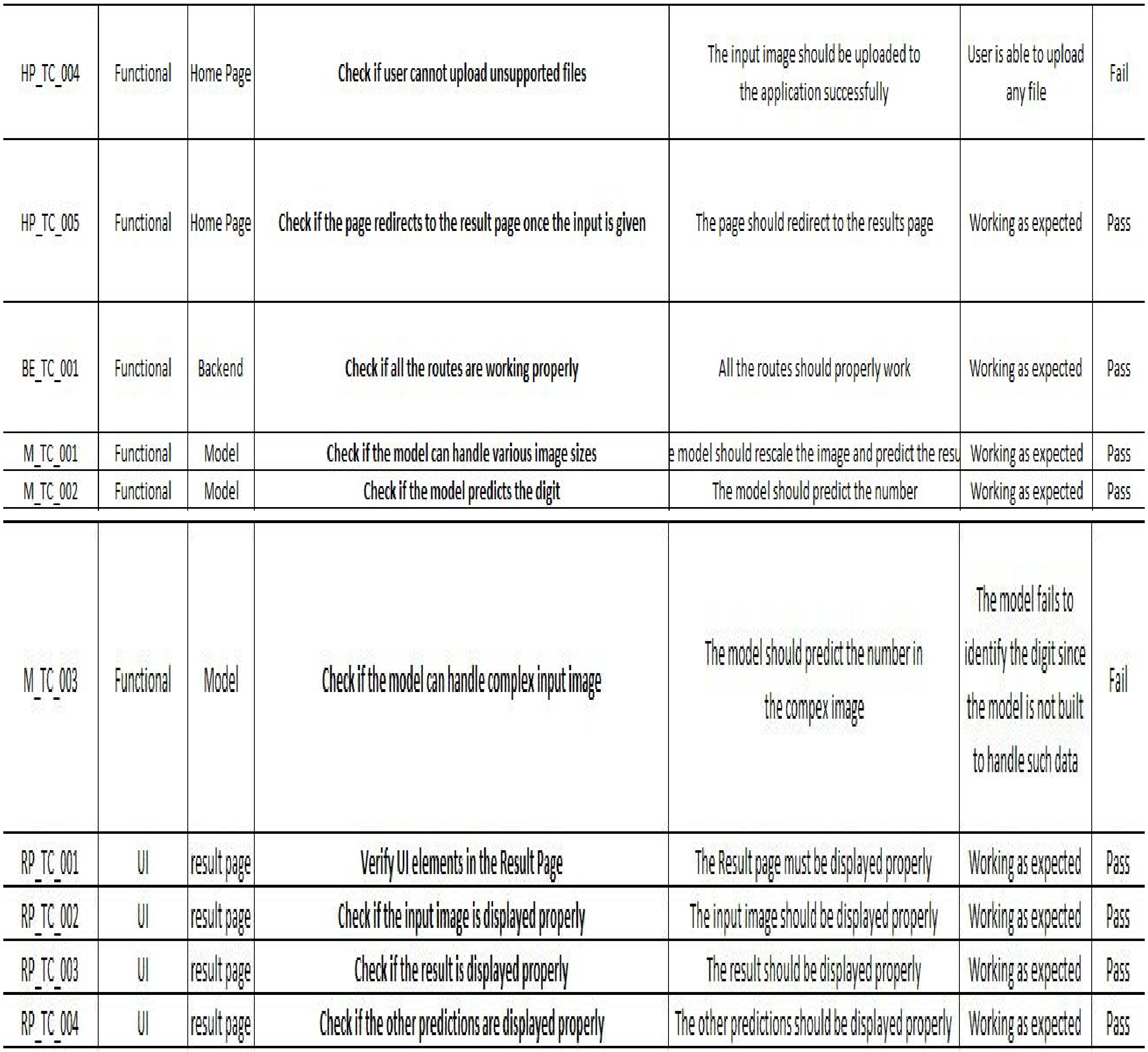
Predictions:

1) User can see the prediction percentage of the digit
2) User can see the other predictions also for better understanding of system.

CHAPTER 8
TESTING

8.1 TEST CASES

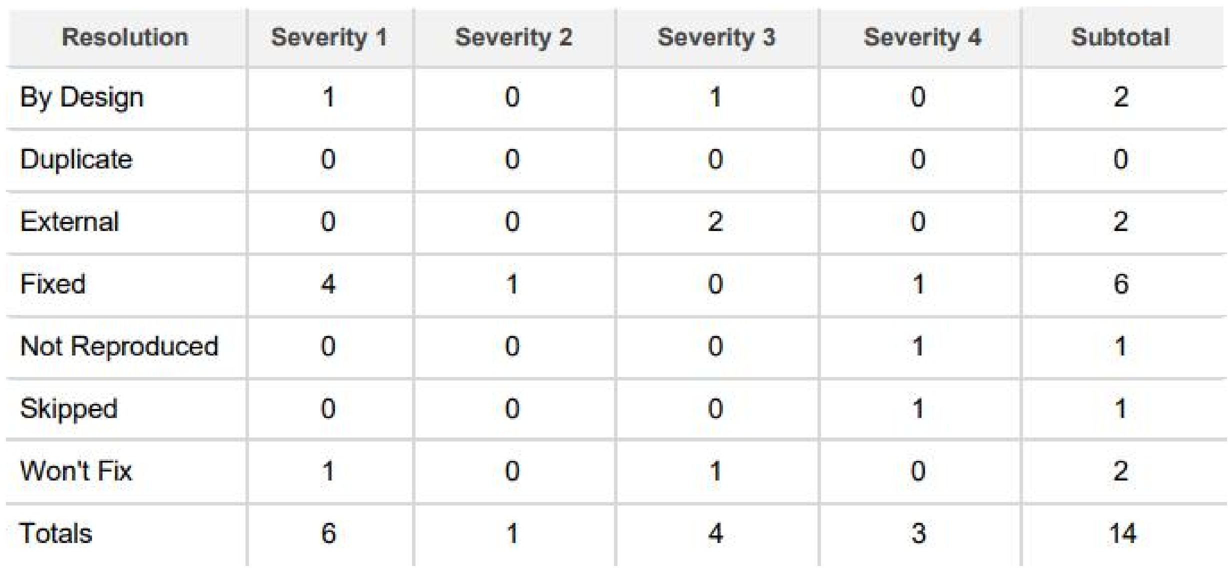
Test case ID	Feature Type	Component	Test Scenario	Expected Result	Actual Result	Status
HP_TC_001	UI	Home Page	**Verify UI elements in the Home Page**	The Home page must be displayed properly	Working as expected	Pass
HP_TC_002	UI	Home Page	**Check if the UI elements are displayed properly in different screen siz**	The Home page must be displayed properly in all size	Working as expected	Pass
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 input image should be uploaded to the application successfully	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	e model should rescale the image and predict the resu	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 compex 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	Working as expected	Pass
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

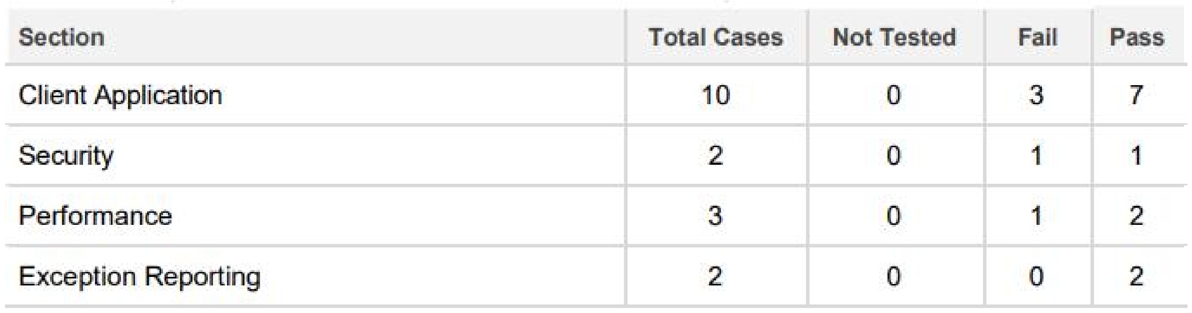
8.2 USER ACCEPTANCE TESTING

8.2.1 DEFECT ANALYSIS



Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
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
Totals	6	1	4	3	14

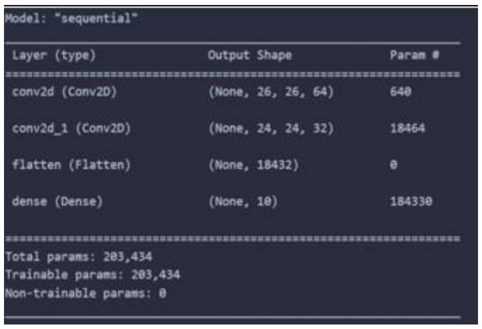
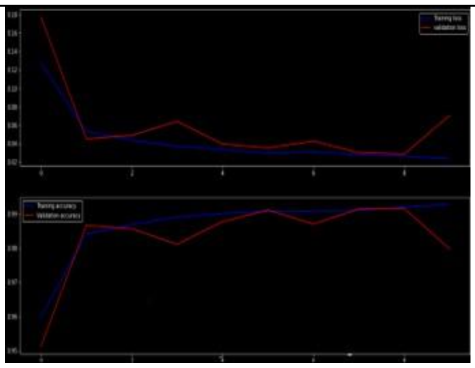
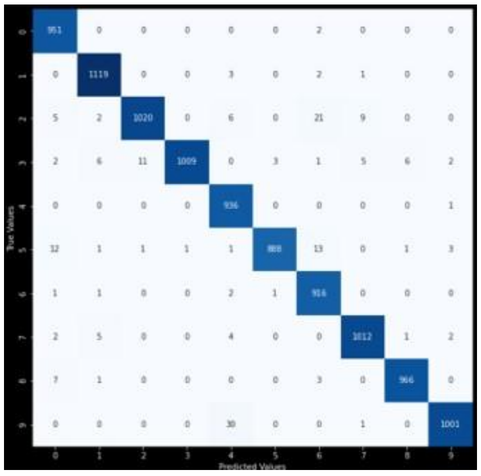
8.2.2 TEST CASE ANALYSIS



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

S.No.	**Parameter**	**Values**	**Screenshot**
1.	Model Summary		
2.	Accuracy	Training Accuracy - 99.14% Validation Accuracy - 97.76% Training Loss - 2.70% Validation Loss – 10.36%	
3.	Confusion Matrix		

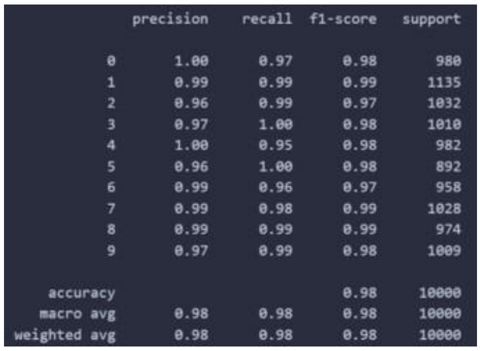
```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 26, 26, 64)        640

 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
_________________________________________________________________
```

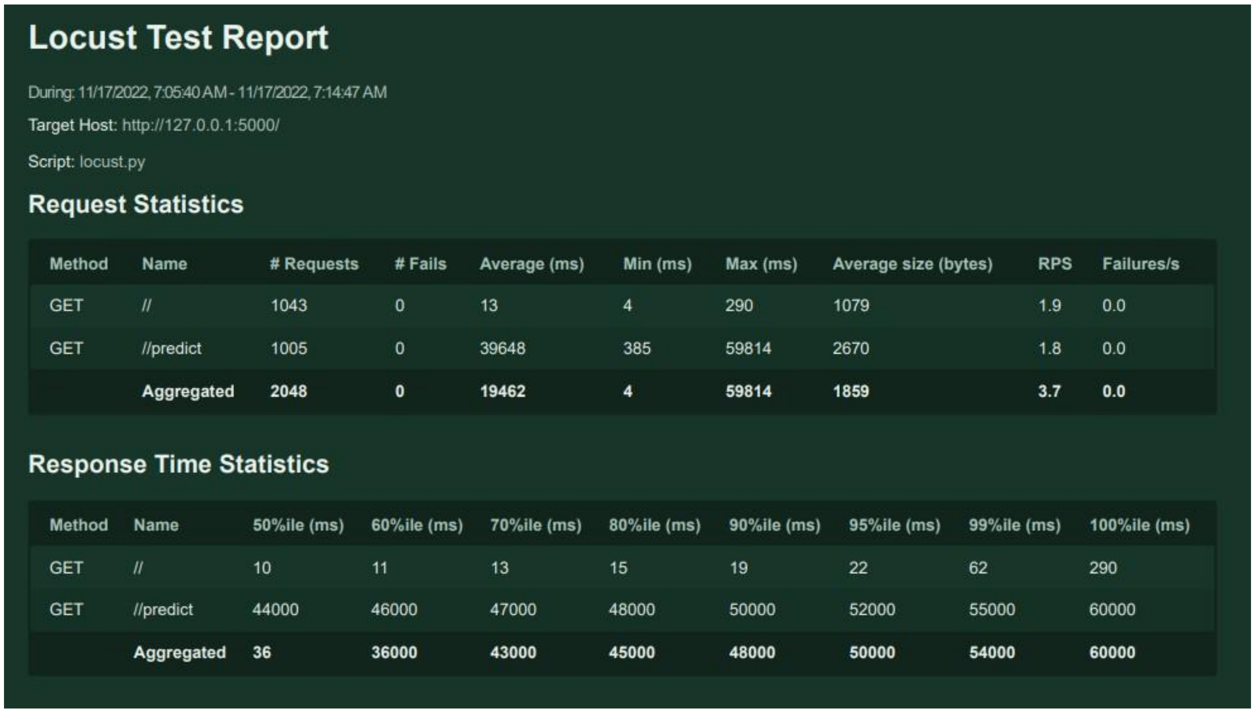
4.	Classification Report		

```
              precision    recall  f1-score   support

           0       1.00      0.97      0.98       980
           1       0.99      0.99      0.99      1135
           2       0.96      0.99      0.97      1032
           3       0.97      1.00      0.98      1010
           4       1.00      0.95      0.98       982
           5       0.96      1.00      0.98       892
           6       0.99      0.96      0.97       958
           7       0.99      0.98      0.99      1028
           8       0.99      0.99      0.99       974
           9       0.97      0.99      0.98      1009

    accuracy                           0.98     10000
   macro avg       0.98      0.98      0.98     10000
weighted avg       0.98      0.98      0.98     10000
```

9.2APPLICATION TEST REPORT



Locust Test Report

During: 11/17/2022, 7:05:40 AM - 11/17/2022, 7:14:47 AM

Target Host: http://127.0.0.1:5000/

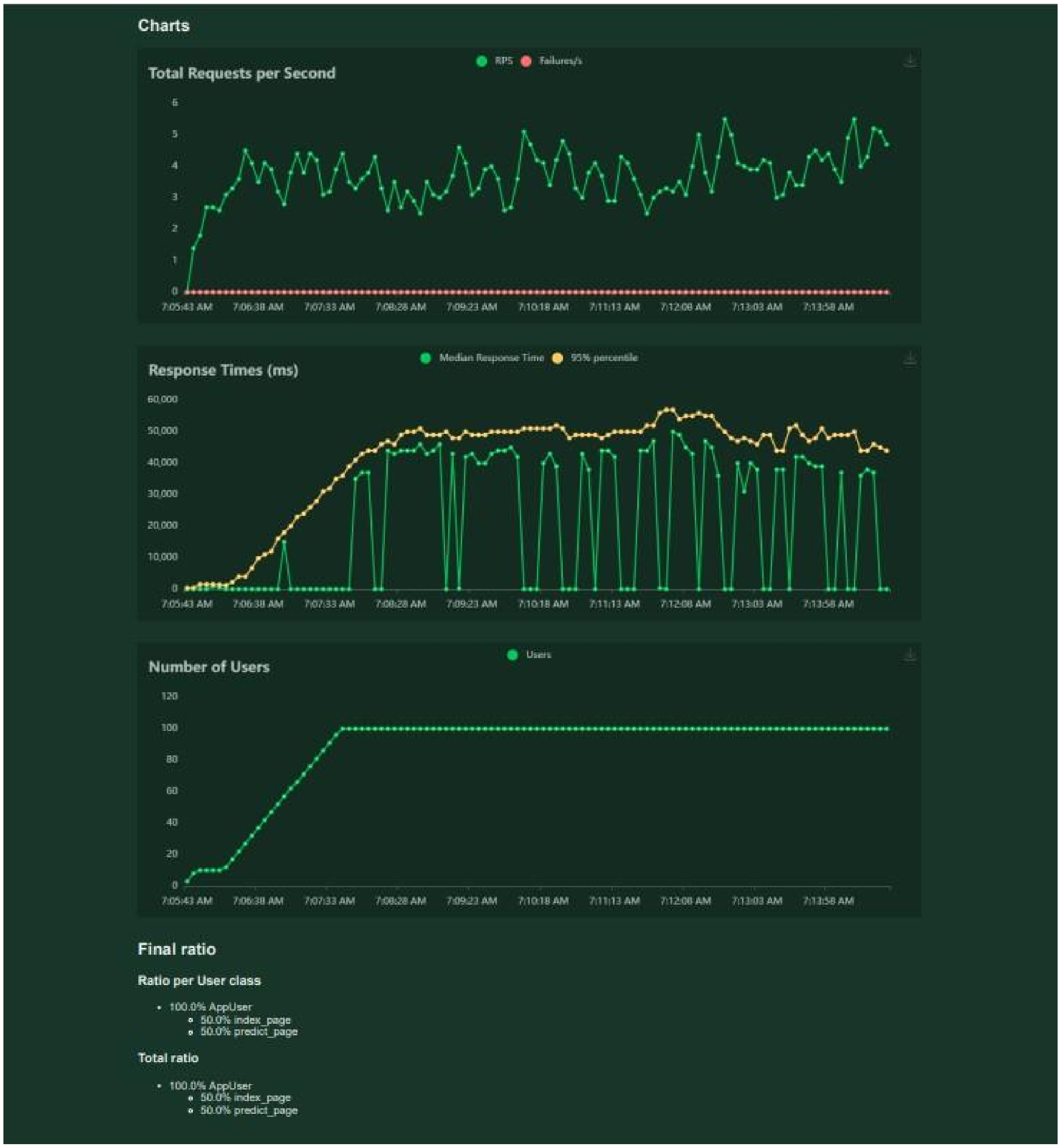
Script: locust.py

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**



Final ratio

Ratio per User class

- 100.0% AppUser
  - 50.0% index_page
  - 50.0% predict_page

Total ratio

- 100.0% AppUser
  - 50.0% index_page
  - 50.0% predict_page

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
- They can scan the handwritten text
- it extracts text & can embed the text
- It reduces the hardness in recognizing variety of handwritten digits
- Can be used by clerks, financiers, accountants

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

Using Neural Network system, back-propagation learning, to recognize handwritten digits was very successful. An image, which contained 100 samples of each number, was trained and tested.

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 95.21% 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.

It will need to take a close look at the system and should look for improvements for the future. From the net-file, the system was able to produce an image-file. The image-file produced showed the recognized number. This part will also need more improvements. Apart from the above problems and parts that need improvements, the overall recognition system was successful.

CHAPTER 13
APPENDIX

SOURCE CODE

MODEL CREATION

```
#IMPORTING THE REQUIRED LIBRARIES

import numpy #for numerical analysis
import tensorflow #open source ml tool by google
from tensorflow.keras.datasets import mnist #mnist dataset
from tensorflow.keras.models import Sequential
from tensorflow.keras import layers
from tensorflow.keras.layers import Dense,Flatten
from tensorflow.keras.layers import Conv2D
from tensorflow import keras
from tensorflow.keras.optimizers import Adam
from keras.utils import np_utils

#LOADING THE DATASET

(x_train,y_train),(x_test,y_test)=mnist.load_data()

print(x_train.shape)
print(y_train.shape)

#ANALYZING THE DATA

x_train[3]
y_train[3]

import matplotlib.pyplot as plt
plt.imshow(x_train[3])

#RESHAPING THE DATA.
```

```
x_train=x_train.reshape(60000,28,28,1).astype('float32')
x_test=x_test.reshape(10000,28,28,1).astype('float32')
APPLY ONE HOT ENCODING

no_of_classes=10
y_train=np_utils.to_categorical(y_train,no_of_classes)
y_test=np_utils.to_categorical(y_test,no_of_classes)
y_test[3]

#MODEL BUILDING

#CREATE THE MODEL / ADD CNN LAYERS

model=Sequential()

model.add(Conv2D(64,(3,3),input_shape=(28,28,1),activation='relu'))
model.add(Conv2D(32,(3,3),activation='relu'))

model.add(Flatten())
model.add(Dense(no_of_classes,activation='softmax'))
COMPILING THE MODEL

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

#TRAIN THE MODEL

model.fit(x_train,y_train,validation_data=(x_test,y_test),epochs=5,batch_size=32)

#OBSERVING THE MERTICS

metrics=model.evaluate(x_test,y_test,verbose=0)
print("metrics-score=>test loss & accuracy")
print(metrics)

#TEST THE MODEL
```

```
prediction=model.predict(x_test[:5])
print(prediction)

import numpy as np
print(np.argmax(prediction,axis=1))
print(y_test[:5])

#SAVE THE MODEL

model.save('mnistcnn.h5')
```

FLASK APP (app.py)

```
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()
```

RECOGNIZER (recognizer.py)

```
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: int) -> str:
                                        """
                                        Generates a random file name.
                                        Args:
                                        n (int): Length the of the file name.
                                        Returns:
                                        str: The file name.
                                        """
                                        return ''.join(random.choices(string.ascii_uppercase +
string.digits, k=n))

def recognize(image: bytes) -> tuple:
                                        """
                                        Predicts the digit in the image.
                                        Args:
                                        image (bytes): The image data.
                                        Returns:
                                        tuple: The best prediction, other predictions and file name
                                        """

                                        model=load_model(Path("./model/mnistcnn.h5"))

                                        img = Image.open(image).convert("L")

                                        # Generate a random name to save the image file.
                                        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}"))

                                        # Convert the Image to Grayscale, Invert it and Resize to
get better prediction.
                                        img = ImageOps.grayscale(img)
                                        img = ImageOps.invert(img)
                                        img = img.resize((28, 28))

                                        # Convert the image to an array and reshape the data to
make prediction.
                                        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]

                                        # Get all the predictions and it's respective accuracy.
                                        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))

                                        # Get the value with the highest accuracy
                                        best = others.pop(best)

                                        return best, others, img_name
```

HOME PAGE HTML (home.html)

```
<html>
   <head>
      <meta name="viewport" content="width=device-width, initial-scale=1.0" />
      <title>A Novel Method for Handwritten Digit Recognition System</title>
      <link rel="icon" type="image/svg" sizes="32x32"
href="{{url_for('static',filename='images/icon.svg')}}" />
```

```
        <link rel="stylesheet" href="{{url_for('static',filename='css/main.css')}}" />
        <script src="https://unpkg.com/feather-icons"></script>
        <script defer src="{{url_for('static',filename='js/script.js')}}"></script>
    </head>
    <body>
        <div class="container">
            <div class="heading">
                <h1 class="heading_main">Handwritten Digit Recognition Application -
PNT2022TMID14907</h1>
                <h2 class="heading_sub">This Handwritten Digit Recognition Application System
Is Used To Easily Aanalyze And Detect Handwritten Digits</h2>
            </div>
            <div class="upload-container">
                <div class="form-wrapper">
                    <form class="upload" action="/predict" method="post" enctype="multipart/form-
data">
                        <label id="label" for="upload-image"><i data-feather="file-plus"></i>Select
File</label>
                        <input type="file" name="photo" id="upload-image" hidden />
                        <button type="submit" id="up_btn"></button>
                    </form>
                    <img id="loading" src="{{url_for('static',filename='images/loading.gif')}}">
                </div>
            </div>
        </div>
    </body>
</html>
```

HOME PAGE CSS (main.css)

```
@import
url("https://fonts.googleapis.com/css2?family=Overpass:wght@200;300;400;500;600;700;900
&display=swap");

* {
                                        padding: 0;
                                        margin: 0;
}

body {
                                        color: black;
                                        font-family: "Overpass", sans-serif;
}

.container {
                                        width: 100%;
                                        height: 100%;
                                        display: flex;
                                        flex-direction: column;
                                        justify-content: center;
                                        align-items: center;
    background-color: lightblue;
    background-image: url("/static/images/header.png");
    /* Center and scale the image nicely */
    background-position: center;
    background-repeat: no-repeat;
    background-size: cover;


}

.heading {
                                        margin-top: -2rem;
                                        padding-bottom: 2rem;
                                        width: fit-content;
                                        text-align: center;
```

```
}

.heading .heading__main {
                                        font-size: 3rem;
                                        font-weight: 550;
}

.heading .heading__sub {
                                        font-size: 1rem;
                                        color: rgb(90, 88, 88);
}

.upload-container {
                                        box-shadow: 0 0 20px rgb(172, 170, 170);
                                        width: 40rem;
                                        height: 25rem;
                                        padding: 1.5rem;
}

.form-wrapper {
                                        background-color: rgb(150, 198, 250, 0.5);
                                        width: 100%;
                                        height: 100%;
                                        display: flex;
                                        border: 1px dashed black;
                                        justify-content: center;
                                        align-items: center;
}

.form-wrapper #loading {
                                        display: none;
                                        position: absolute;
}

.form-wrapper .upload {
                                        display: flex;
                                        justify-content: center;
```

```
                                            align-items: center;
                                            width: 8rem;
                                            height: -webkit-fit-content;
                                            height: -moz-fit-content;
                                            height: fit-content;
                                            border-radius: 6px;
                                            color: white;
                                            background-color: rgb(191, 40, 88);
                                            box-shadow: 0 5px 10px rgb(146, 135, 247);
}

.form-wrapper .upload #up_btn {
                                            display: none;
}

.form-wrapper .upload label {
                                            font-size: 1rem;
                                            font-weight: 600;
                                            color: white;
                                            height: 100%;
                                            width: 100%;
                                            padding: 10px;
                                            display: block;
}

.form-wrapper .upload svg {
                                            height: 15px; width:
                                            auto; padding-right:
                                            8px;
                                            margin-bottom: -2px;
}

@media screen and (max-width: 700px) {
                                            .upload-container
                                            { height: 20rem;
                                            width: 18rem;
                                            margin-top: 3.5rem;
```

```
                                        margin-bottom: -8rem;
                                        }

                                        .heading .heading_main
                                        {margin-top: -6rem;
                                        font-size: 2rem;
                                        padding-bottom: 1rem;
                                        }
}
```

HOME PAGE JS (script.js)

```
feather.replace(); // Load feather icons

form = document.querySelector('.upload')
loading = document.querySelector("#loading")
select = document.querySelector("#upload-image");

select.addEventListener("change", (e) => {
                                        e.preventDefault();

                                        form.submit()
                                        form.style.visibility = "hidden";
                                        loading.style.display = 'flex';
});
```

PREDICT PAGE HTML (predict.html)

```
<html>
                                        <head>
                                        <title>Prediction | A Novel Method for Handwritten Digit
Recognition System </title>
                                        <link rel="stylesheet"
href="{{url_for('static',filename='css/predict.css')}}" />
                                        <link rel="icon" type="image/svg" sizes="32x32"
href="{{url_for('static',filename='images/icon.svg')}}" />
```

```
                                                <meta name="viewport" content="width=device-width,
initial-scale=1.0" />
                                                </head>
                                                <body>
                                                <div class="container">
                                                          <h1>The Predicted Result Is Displayed</h1>
                                                          <div class="result-wrapper">
                                                                    <div class="input-image-container">
                                                                              <img
src="{{url_for('static',filename='data/')}}{{img_name}}" />
                                                                    </div>
                                                                    <div class="result-container">
                                                                              <div
class="value">{{best.0}}</div>
                                                                              <div
class="accuracy">{{best.1}}%</div>
                                                                    </div>
                                                          </div>
                                                          <h1>Other Predictions Are Given Below For
Reference</h1>
                                                          <div class="other_predictions">
                                                                    {% for x in others %}
                                                                    <div class="value">
                                                                              <h2>{{x.0}}</h2>
                                                                              <div
class="accuracy">{{x.1}}%</div>
                                                                    </div>
                                                                    {% endfor %}
                                                          </div>
                                                </div>
                                                </body>
</html>
```

PREDICT PAGE CSS (predict.css)

```
@import
url("https://fonts.googleapis.com/css2?family=Overpass:wght@200;300;400;500;600;700;900
&display=swap");

body {
                                        color: black;
                                        font-family: "Overpass", sans-serif;


}

h1 {
                                        padding-top: 2rem;
}

.container {
    width: 100%;
                                        height: 100%;
                                        display: flex;
                                        justify-content: center;
                                        align-items: center;
    flex-direction: column;
    background-color: lightblue;
    background-image: url("/static/images/header.png");
    /* Center and scale the image nicely */
    background-position: center;
    background-repeat: no-repeat;
    background-size: cover;




}

.result-wrapper {
                                        width: -webkit-fit-content;
                                        width: -moz-fit-content;
                                        width: fit-content;
                                        height: -webkit-fit-content;
                                        height: -moz-fit-content;
                                        height: fit-content;
                                        box-shadow: 0 0 10px rgb(126, 125, 125);
                                        padding: 1.5rem;
                                        display: flex;
                                        justify-content: center;
                                        align-items: center;
                                        -moz-column-gap: 1rem;
                                        column-gap: 1rem;
}

.result-wrapper .input-image-container,
.result-wrapper .result-container {
                                        width: 15rem;
                                        height: 15rem;
                                        border: 1px dashed black;
                                        justify-content: center;
                                        display: flex;
                                        align-items: center;
                                        flex-direction: column;
    background-color: rgba(150, 198, 250, 0.5);


}

.result-wrapper .input-image-container img {
                                        width: 60%;
                                        height: 60%;
                                        background-color: rgba(150, 198, 250, 0.5);
                                        background-size: contain;
}

.result-wrapper .result-container .value {
                                        font-size: 6rem;
}
```

```
.result-wrapper .result-container .accuracy {
                                        margin-top: -1rem;
}

.other_predictions {
                                        display: flex;
                                        justify-content: center;
                                        align-items: center;
                                        flex-wrap: wrap;
                                        column-gap: 1rem;
                                        row-gap: 1rem;
                                        font-weight: 700;


}

.other_predictions .value {
                                        display: flex;
                                        justify-content: center;
                                        align-items: center;
                                        flex-direction: column;
                                        width: 5rem;
                                        height: 5rem;
                                        box-shadow: 0 0 7px rgb(158, 157, 157);
    background-color: rgba(128, 0, 128, 0.8);
}

.other_predictions .value div {
                                        margin-top: -1.2rem;
}

@media screen and (max-width: 700px) {
                                        h1 {
                                        font-size: 2.3rem;
                                        }

                                        .result-wrapper .input-image-container,
                                        .result-wrapper .result-container
                                        {width: 7rem;
                                        height: 7rem;
                                        }

                                        .result-wrapper .result-container .value
                                        {font-size: 4rem;
                                        }
}
```

GITHUB

https://github.com/IBM-EPBL/IBM-Project-25876-1659975630