

# **A Novel Method for Handwritten Digit Recognition System**



## **IBM PROJECT** *in* **Artificial Intelligence Domain**

**PSNA COLLEGE OF ENGINEERING AND TECHNOLOGY**

**(AUTONOMOUS)**

**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

**A PROJECT REPORT**

*Submitted by*

Sivaprakash s	-	921319106209
Sharvesh D S	-	921319106203
Saravana Karthikeyan A	-	921319106195
Saravana Prabhu K	-	921319106196

TEAM ID : ***PNT2022TMID05335***

# 1. INTRODUCTION

## 1.1 Project Overview

Machine learning and deep learning plays an important role in Information technology and artificial intelligence. With the use of deep learning and machine learning, human effort can be reduced in recognizing, learning, predictions and many more areas. This article presents recognizing the handwritten digits (0 to 9) from the famous MNIST dataset, comparing classifiers like KNN, PSVM, NN and convolution neural network on basis of performance, accuracy, time, sensitivity, positive productivity, and specificity with using different parameters with the classifiers. To make machines more intelligent, the developers are diving into machine learning and deep learning techniques. A human learns to perform a task by practicing and repeating it again and again so that it memorizes how to perform the tasks. Then the neurons in his brain automatically trigger and they can quickly perform the task they have learned. Deep learning is also very similar to this. It uses different types of neural network architectures for different types of problems.

**For example** – object recognition, image and sound classification, object detection, image segmentation, etc. 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.

## 1.2 Purpose of Digit Recognition System:

Digit recognition system is the working of a machine to train itself or 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 , numeric entries in forms filled up by hand and so on.

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

## 2. LITERATURE SURVEY

TITLE : Handwritten Character Recognition

AUTHOR : Ayush Purohit DESCRIPTION : Handwriting recognition has gained a lot of attention in the field of pattern recognition and machine learning due to its application in various fields. Optical Character Recognition (OCR) and Handwritten Character Recognition (HCR) has specific domain to apply. Various techniques have been proposed to for character recognition in handwriting recognition system. Even though, sufficient studies and papers describes the techniques for converting textual content from a paper document into machine readable form. In coming days, character recognition system might serve as a key factor to create a paperless environment by digitizing and processing existing paper documents.

PUBLISHED IN : 2016

TITLE : Automatic Handwritten Digit Recognition

AUTHOR : Akkireddy Challa

DESCRIPTION : The main purpose of this thesis is to build an automatic handwritten digit recognition method for the recognition of connected handwritten digit strings. To accomplish the recognition task, first, the digits were segmented into individual digits. Then, a digit recognition module is employed to classify each segmented digit completing the handwritten digit string recognition task. In this study, different machine learning methods, which are SVM, ANN and CNN architectures are used to achieve high performance on the digit string recognition problem. In these methods, images of digit strings are trained with the SVM, ANN and CNN model with HOG feature vectors and Deep learning methods structure by sliding a fixed size window through the images labeling each sub-image as a part of a digit or not. After the completion of the segmentation, to achieve the complete recognition of handwritten digits.

PUBLISHED IN : 2019

TITLE : Handwritten Numeral Recognition

AUTHOR : Stuti Asthana

DESCRIPTION : An extensive literature review on Neural Network based numeric recognition by describing the survey of some research articles have been involved for better analysis in order to enhance the system efficiency. Handwritten Numeric

Recognition is very interesting area of Pattern Recognition and it deals with Offline Handwriting Recognition. Handwriting Recognition has kept on continuing as a method for correspondence, gathering, recording and transmitting data in everyday life since the hundreds of years even with the appearance of the new advancements. Machine recognition has numerous functional applications, perusing manually written postal envelopes, sum written in bank checks, bill handling, government records, business frames, signature confirmation, disconnected from the net archive acknowledgment and so on. This Paper portrays the best in class study of the work accomplished for the Numeric recognition.

PUBLISHED IN : 2017

TITLE : Neural Network Based Handwritten Digit Recognition

AUTHOR : Ankit Sharma

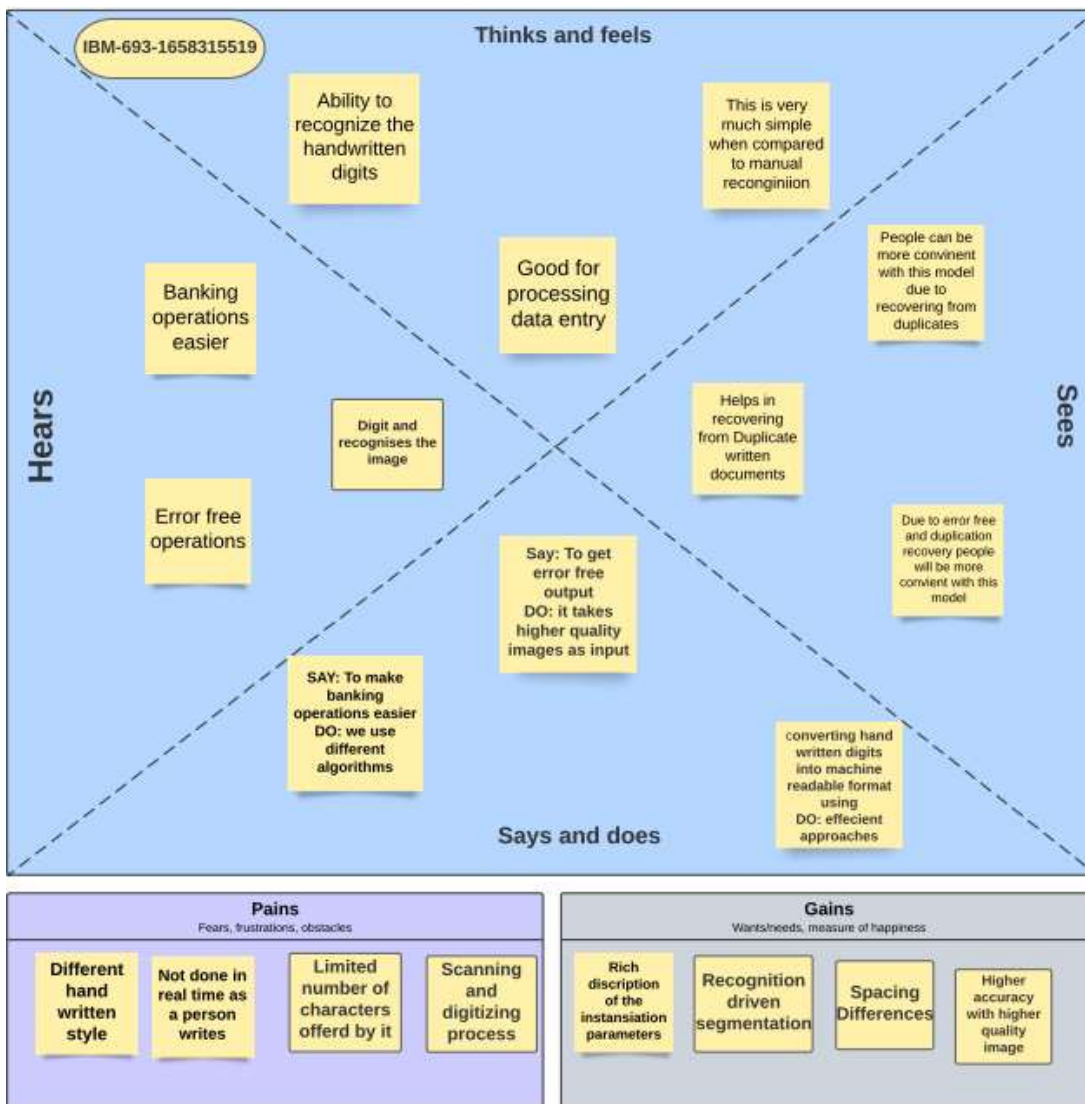
DESCRIPTION : Recognition of handwritten character is a difficult task in the field of image processing, artificial intelligence since the handwriting varies from person to person. In proposed paper, we are training the neural network to recognize the off-line strategies for the isolated handwritten character (0 to 9). This work improves the character recognition and pre- processing of the Character is done by image rendering, character extraction and training and testing steps. The proposed method is based on the use of linear regression algorithm to classify the characters and is used to train the given dataset. After training a network performance curve is generated along with the individual required characters. In given system, numerical character is represented by binary numbers that are used as input then they are fed to an ANN. Neural network followed by the linear regression Algorithm which compromises Training.

PUBLISHED IN : 2016

### 3. IDEATION & PROPOSED SOLUTION

#### 3.1 Empathy Map Canvas

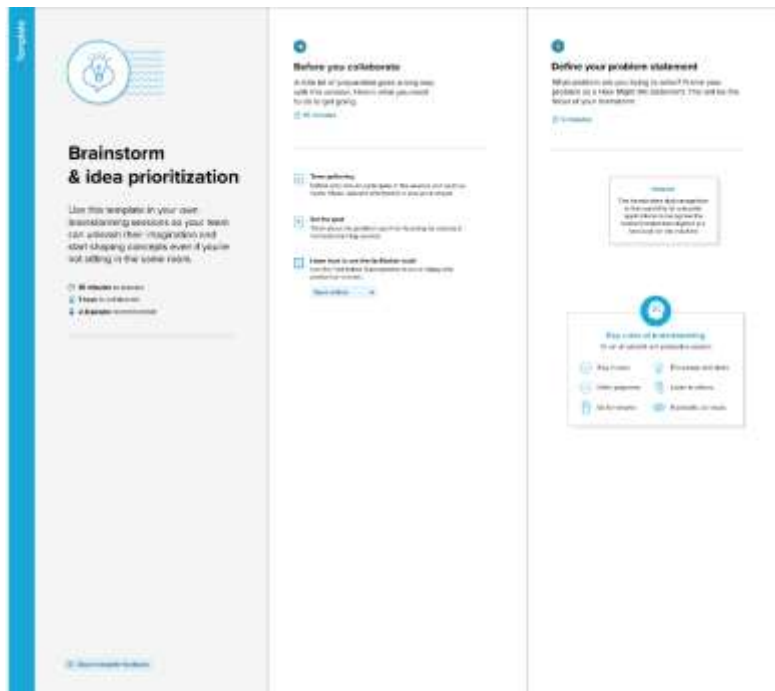
An empathy map is a **collaborative tool teams can use to gain a deeper insight into their customers**. Much like a user persona, an empathy map can represent a group of users, such as a customer segment. The empathy map was originally created by Dave Gray and has gained much popularity within the agile community.



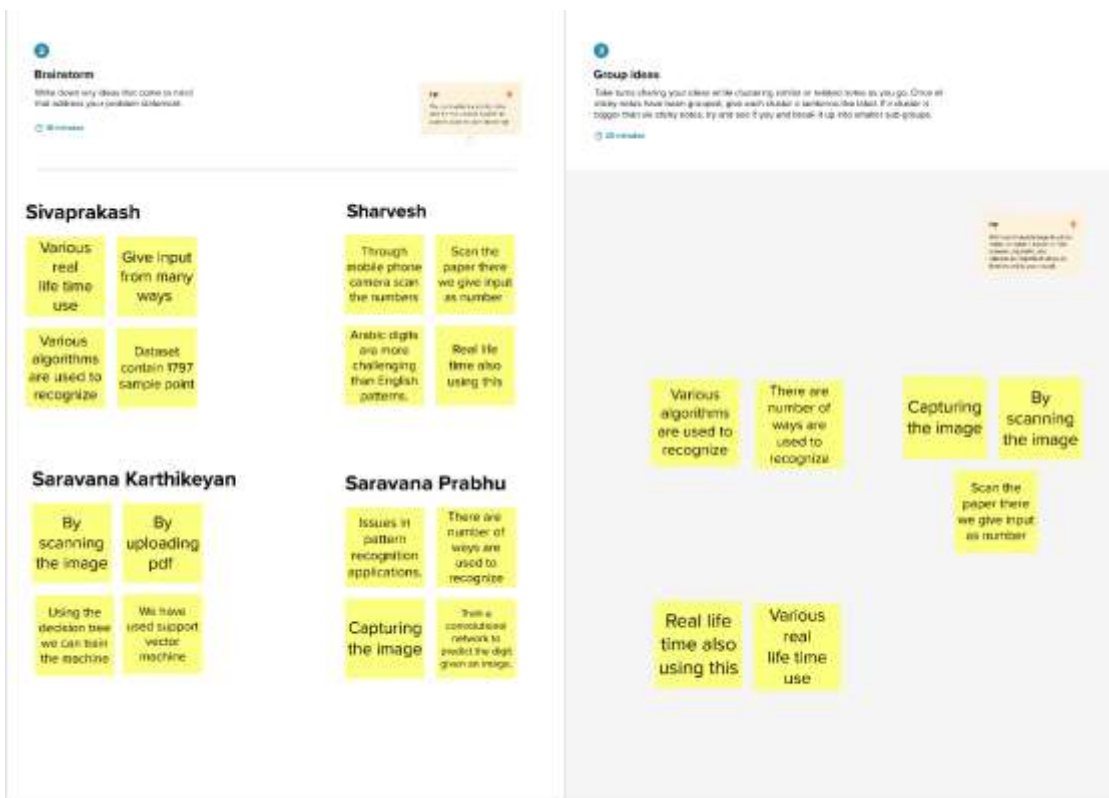
#### 3.2 Ideation & Brainstorming

**Ideation** refers to the whole creative process of coming up with and communicating new ideas. **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.

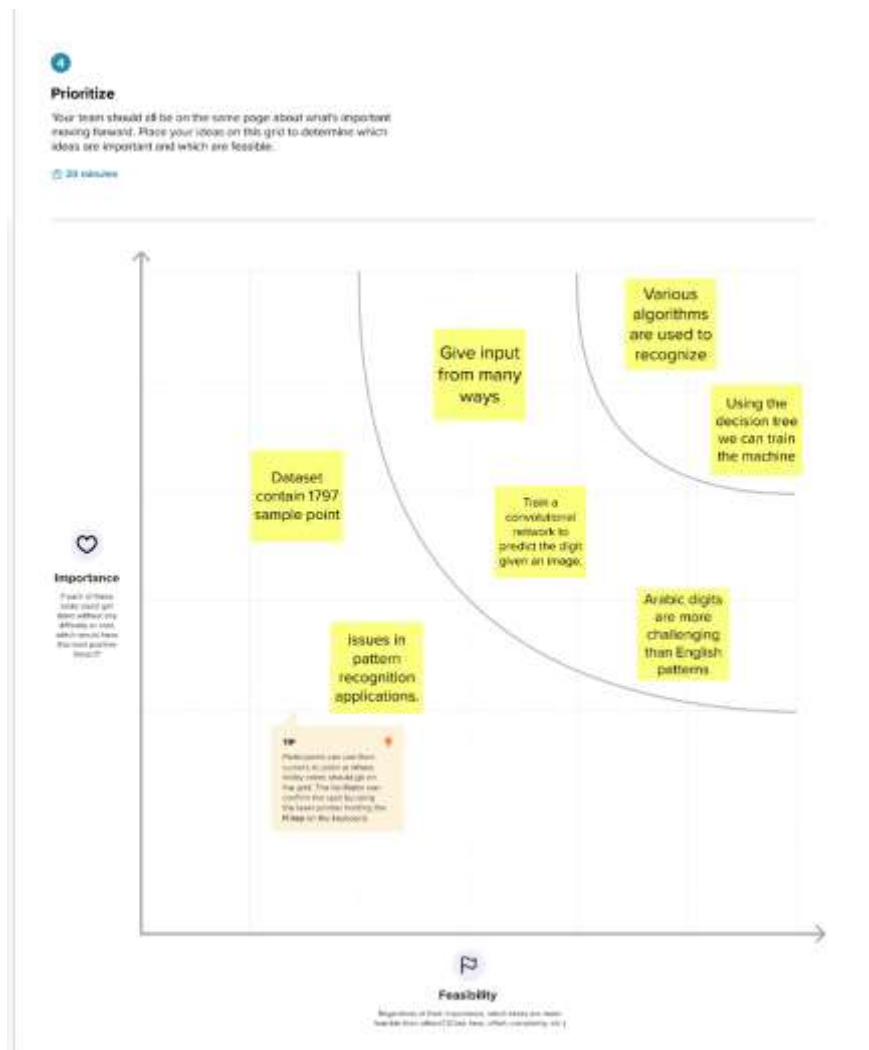
**Step-1: Team Gathering, Collaboration and Select the Problem Statement**



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



## Step-3: Idea Prioritization



### 3.3 Proposed Solution

Project team shall fill the following information in proposed solution template.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	<b>Statement</b> -The handwritten digit recognition is the capability of computer applications to recognize the human handwritten digits. <b>Description:</b> It is a hard task for the machine because handwritten digits are not perfect and can be made with many different shapes and size.
2.	Idea / Solution description	<b>1.</b> It is the capability of a computer to fete the mortal handwritten integers from different sources

		<p>like images, papers, touch defences.</p> <p><b>2.</b> Hand written digit recognition system allows user to translate all those signature and notes into electronic words in a text document format and this data only requires farless physical space than the storage of the physical copies.</p>
3.	Novelty / Uniqueness	<p>Accurately recognize the digits rather than recognizing all the characters like (Object Character Recognition)OCR.</p>
4.	Social Impact / Customer Satisfaction	<p><b>1.</b>Artificial Intelligence developed the app called Handwritten digit Recognizer.</p> <p><b>2.</b> It converts the written word into digital approximations and utilizes complex algorithms to identify characters before churning out a digital approximation.</p>
5.	Business Model (Revenue Model)	<ul style="list-style-type: none"> <li>• This system can be integrated with traffic surveillance cameras to recognize the vehicle's number plates for effective traffic management.</li> <li>• Can be integrated with Postal system to identify and recognize the pin-code details easily.</li> </ul>
6.	Scalability of the Solution	<ul style="list-style-type: none"> <li>• Ability to recognise digits in more noisy environments.</li> <li>• There is no limit in the number of digits it can be recognized</li> </ul>

### 3.4 Problem Solution Fit



Define CS, fit into CC	<b>1. CUSTOMER SEGMENT(S) <span>CS</span></b> <ul style="list-style-type: none"> <li>➤ Customers are those who work with handwritten numbers in places like banks, schools, colleges, railroads, etc.</li> </ul>	<b>6. CUSTOMER CONSTRAINTS <span>CC</span></b> <ul style="list-style-type: none"> <li>➤ Lack of reliable internet connections, unavailability of gadgets like mobile phones and computers, inaccessibility of appropriate cameras.</li> <li>➤ Because handwritten numbers are not always accurate and might have a wide variety of tastes, it is a difficult work for the computer.</li> <li>➤ This issue can be solved by using an image of a digit to identify the digit that is present in the image, which is done through handwritten digit recognition.</li> </ul>	<b>5. AVAILABLE SOLUTIONS <span>AS</span></b> <ul style="list-style-type: none"> <li>➤ Although there are current alternatives to this approach, they are not very precise, robust, or rotation- and variation-invariant.</li> <li>➤ The ability of a computer to honor the mortal handwritten characters from many sources, including as photographs, papers, and touch input.</li> </ul>
	<b>2. JOBS-TO-BE-DONE / PROBLEMS <span>J&amp;P</span></b> <ul style="list-style-type: none"> <li>➤ It is really challenging to comprehend and analyze the handwritten numbers.</li> <li>➤ More training data required.</li> <li>➤ Hard to recognize digits, dim lighting, weak eyesight.</li> </ul>	<b>9. PROBLEM ROOT CAUSE <span>I</span></b> <ul style="list-style-type: none"> <li>➤ Hand-written digits are in varying fonts and sizes; thus, they are becoming increasingly difficult to ascertain due to various factors such as weakening eyesight, time constraints, etc.</li> </ul>	<b>7. BEHAVIOUR <span>I</span></b> <ul style="list-style-type: none"> <li>➤ Finding the best software that more quickly and accurately identifies digits.</li> <li>➤ Customer wants reliable internet connections and high-quality cameras.</li> </ul>
Identify strong TR & EM	<b>3. TRIGGERS</b> <ul style="list-style-type: none"> <li>➤ Obtain the data quickly and accurately.</li> <li>➤ The exchange of information is made simple and is one of the simplest ways to speak with a computer and grasp the language.</li> </ul>	<b>10. YOUR SOLUTION</b> <ul style="list-style-type: none"> <li>➤ The solution aims to reliably recognize hand-written digits using Convolutional Neural Network (CNN) algorithm. Therefore, reducing costs for the company and increasing worker productivity.</li> </ul>	<b>8. CHANNELS OF BEHAVIOUR</b>
	<b>4. EMOTIONS: BEFORE / AFTER</b> <p><b>BEFORE:</b> Uncertain, Reserved, and Perplexed.</p> <p><b>AFTER:</b> Assured, Upright, and Rational.</p>		<b>8.1 ONLINE</b> <ul style="list-style-type: none"> <li>➤ The processing and uploading of the photographs both require a steady internet connection.</li> </ul> <p><b>8.2 OFFLINE</b></p> <ul style="list-style-type: none"> <li>➤ Purchase contemporary electronics and confirm their functionality.</li> </ul>

4.

## Requirements Analysis

### 4.1 Functional Requirements:

Following are the functional requirements of the proposed solution.

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	Confirmation via Email Confirmation via OTP
FR-3	Input data	The system process the input given by the user only if it is an image file (JPG, PNG...). System should detect and retrieve characters present in the image and display them in the user
FR-4	Reporting errors	System shall show the error message to the user when the input given is not in the required format.
FR-5	Data Pre-processing	Performing some normalization and pre-processing in the given input.
FR-6	Classification	The feature extraction method trained on training images dataset of MNIST and then tested on test dataset of MNIST dataset.
FR-7	Accuracy	In his experiments with the MNIST dataset, the deep neural network model provided 99.53% accuracy rate, the convolutional neural network model provided 99.88% accuracy rate, and the iterative neural network model provided 99.05% accuracy rate.

## 5. PROJECT DESIGN

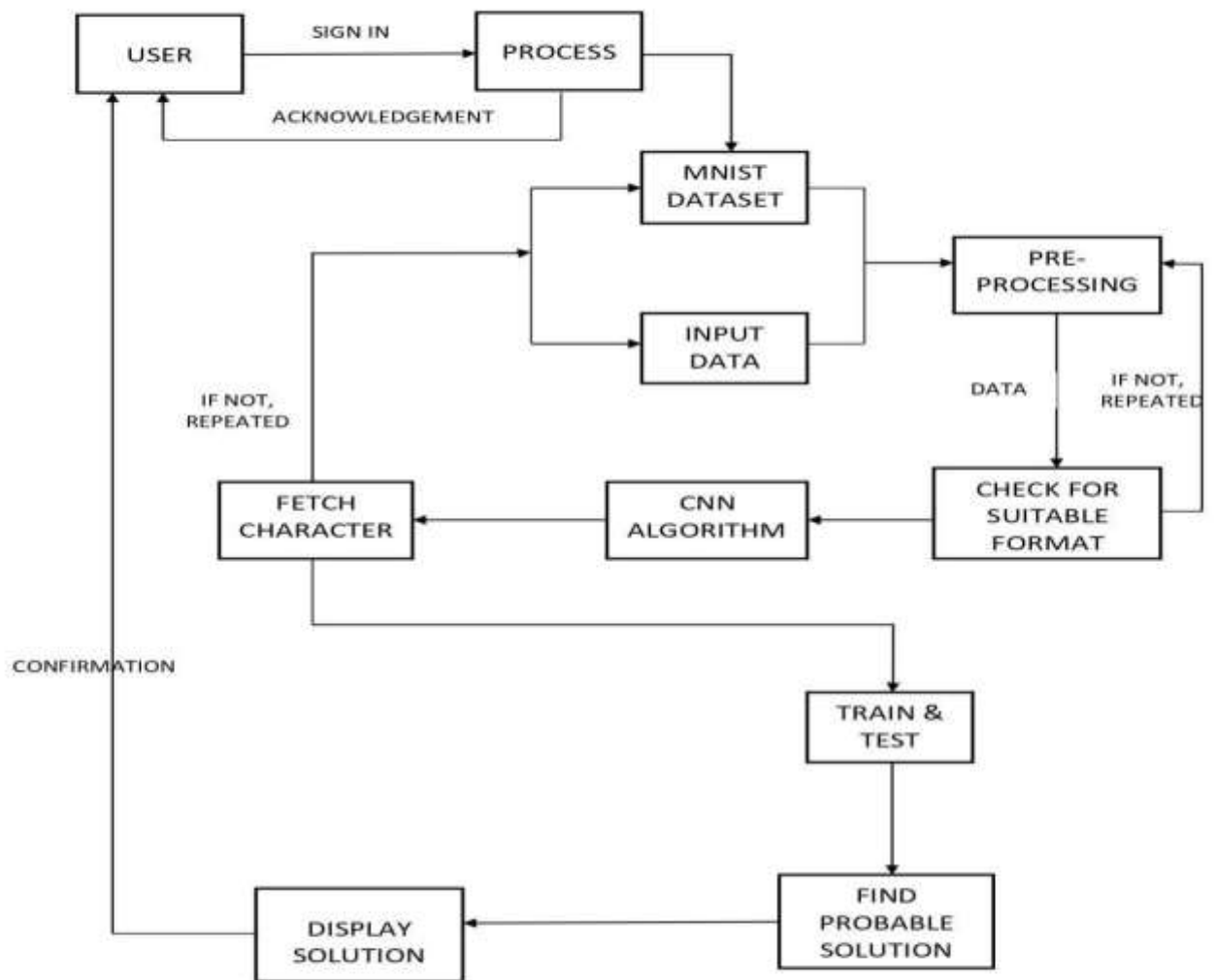
### 5.1 DataFlow Diagram

#### User Stories

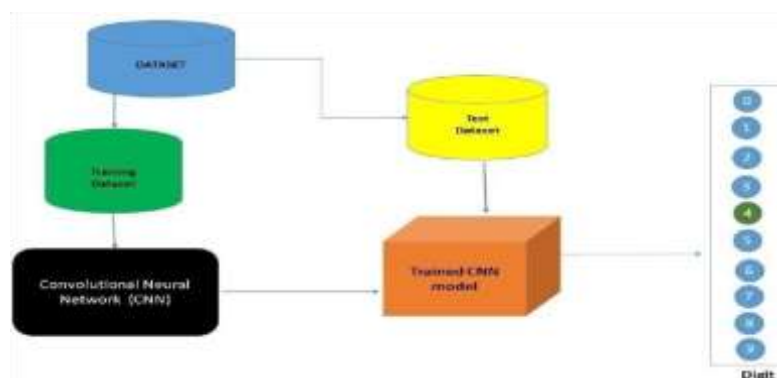
Use the below template to list all the user stories for the product.

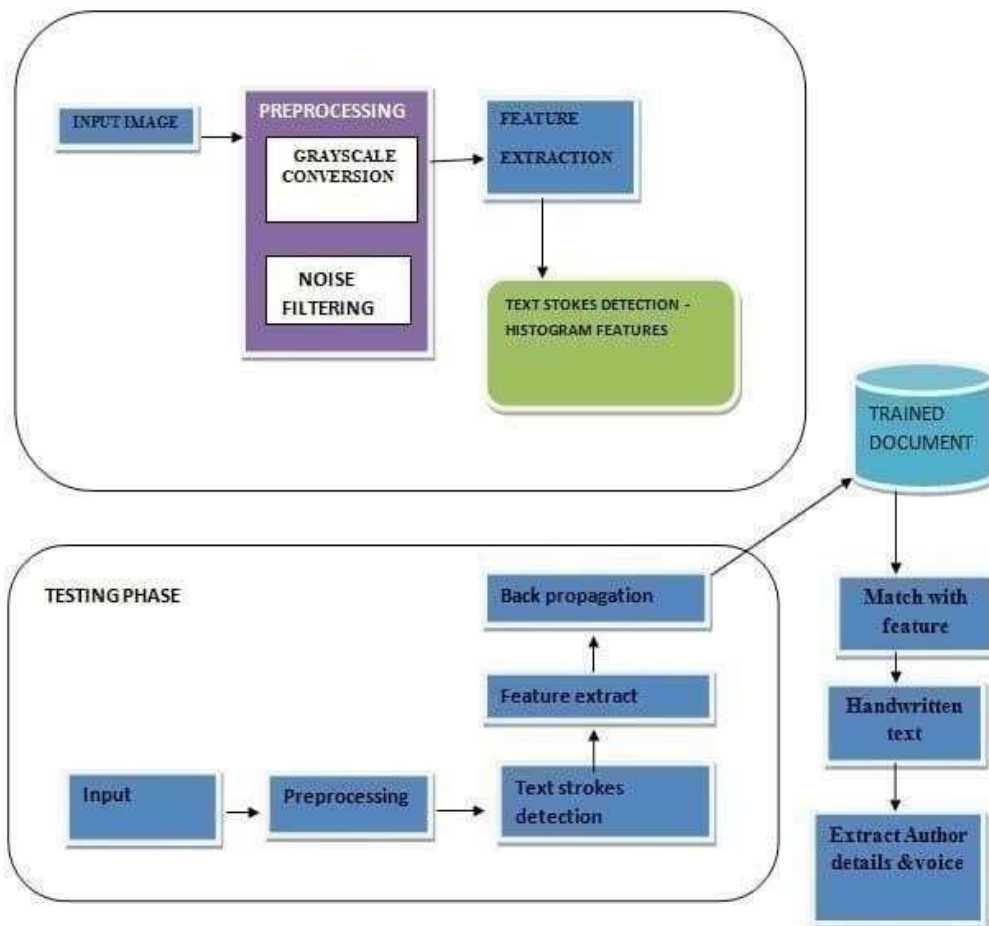
User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1

		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail	I can register and access the dashboard with Google Mail	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	I can login to the application to give my input	High	Sprint-1
	Dashboard	USN-6	As a user. I can move to dashboard section, involves to view various types of visual data	I can see many visual data in dashboard	Low	Sprint-2
Customer (Web user)	Functionality	USN-7	As a web user, I want to give input data to get recognition	I can get the probable match for my input data	High	Sprint-1
Customer Care Executive	Communication	USN-8	As a customer care executive, I can assist the customer to fulfil their expectations	By their suggestions, I can fulfil their needs	High	Sprint-1
Administrator	Classification	USN-9	As an administrator, I can preprocess the data for further process	I can used the preprocessed data	Medium	Sprint-2
	Formatting	USN-10	As an administrator, I can check for the suitable format	I will get errors if it is not in proper format	Low	Sprint-3
	Testing and Training	USN-11	As an administrator, I can feed the data to the CNN	I can train and test the data	High	Sprint-1
	Accuracy	USN-12	As an administrator, I can compare the text and give better accuracy	I can get better accurate output	Medium	Sprint-2
	User Experience	USN-13	As an administrator, I can get the most probable	I can get the most probable digit/character	High	Sprint 1



## 5.2 Solution & Technical Architecture





S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g., Mobile Application	HTML, CSS, JavaScript / Angular JS / Node Red
2.	Application Logic-1	Logic for a process in the application	Java / Python
3.	Application Logic-2	Logic for a process in the application	IBM Watson STT service
4.	Application Logic-3	Logic for a process in the application	IBM Watson Assistant
5.	Database	Data Type, Configurations etc.	MySQL, NoSQL, etc.
6.	Cloud Database	Database Service on AI in cloud	IBM DB2
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.	Internet of Things Model	Purpose of AI Model is for integrating the	IBM AI Platform

		sensors with a user interface.	
10.	Machine Learning Model	Purpose of Machine Learning Model	Object Recognition Model
11.	Infrastructure (Server / AI)	Application Deployment on Local System / AI Local Server Configuration AI Server Configuration	Local, Kubernetes, etc.

**Table-2: Application Characteristics:**

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Deep learning frameworks can help you upload data and train a deep learning model that would lead to accurate and intuitive predictive analysis.	Tensorflow, PyTorch
2.	Security Implementations	The system should automatically be able to authenticate all users with their unique username and password	NA
3.	Scalable Architecture	The system should be able to handle 10000 users accessing the site at the same time	NA
4.	Availability	Information is restricted to each users limited access	NA
5.	Performance	Should reduce the delay in information when hundreds of requests are given	Google Co-Lab Pro/ Require high end system.

## 6. Project Planning & Scheduling

### 6.1 Sprint Planning & Estimation

## 6.2 Sprint Delivery Schedule

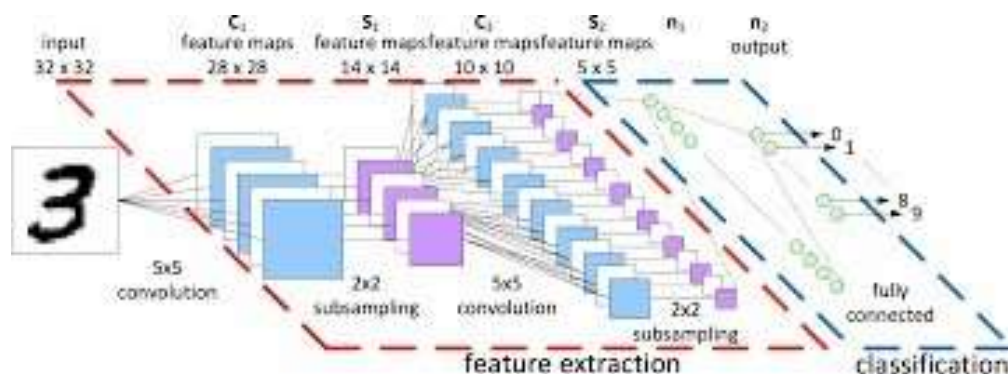
Sprint	Functional Requirement	Task
Sprint-1	Image Data	As a User need to collect the Image Data of Handly Written Images to train the model.
Sprint-2	Dash Board or Website	We using Python Flask Framework to create a dynamic Webpage to host our model (UI).
Sprint-3	Classifier Model	Using CNN Model for Image Classification.
Sprint-4	Cloud	Hosting the Organized appication in Cloud platform.

### 7.1 Feature [1]

i). **Using CNN Model in our Project** : CNN is basically a model known to be **Convolutional Neural Network** and in recent times it has gained a lot of popularity because of its usefulness.

CNN uses multilayer perceptrons to do computational works.

ii). CNN uses relatively little pre-processing compared to other image classification algorithms. This means the network learns through filters that in traditional algorithms were hand-engineered. So, for the image processing tasks CNNs are the best-suited option.



### 7.2 Feature [2]

ii). **Using Flask application in our Project** : Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools.



## 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 sizes	The Home page must be displayed properly in all sizes	The UI is not displayed properly in screen size 2560 x 1801 and 768 x 630	FAIL
HP_TC_003	Functional	Home Page	Check if user can upload their file	The input image should be uploaded to the application successfully	Working as expected	PASS
HP_TC_004	Functional	Home Page	Check if user cannot upload unsupported files	The application should not allow user to select a non image file	User is able to upload any file	FAIL
HP_TC_005	Functional	Home Page	Check if the page redirects to the result page once the input is given	The page should redirect to the results page	Working as expected	PASS
BE_TC_001	Functional	Backend	Check if all the routes are working properly	All the routes should properly work	Working as expected	PASS

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

## 8.2 USER ACCEPTANCE TESTING

### 8.2.1 DEFECT ANALYSIS

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Total
------------	------------	------------	------------	------------	-------

By Design	1	0	1	0	2
Duplicate	0	0	0	0	0
External	0	0	2	0	2
Fixed	4	1	0	1	6
Not Reproduced	0	0	0	1	1
Skipped	0	0	0	1	1
Won't Fix	1	0	1	0	2
Total	6	1	4	3	14

## 8.2.2 TEST CASE ANALYSIS

Section	Total Cases	Not Tested	Fail	Pass
Client Application	10	0	3	7
Security	2	0	1	1
Performance	3	0	1	2
Exception Reporting	2	0	0	2

## 9. RESULTS

### 9.1 Performance Metrics

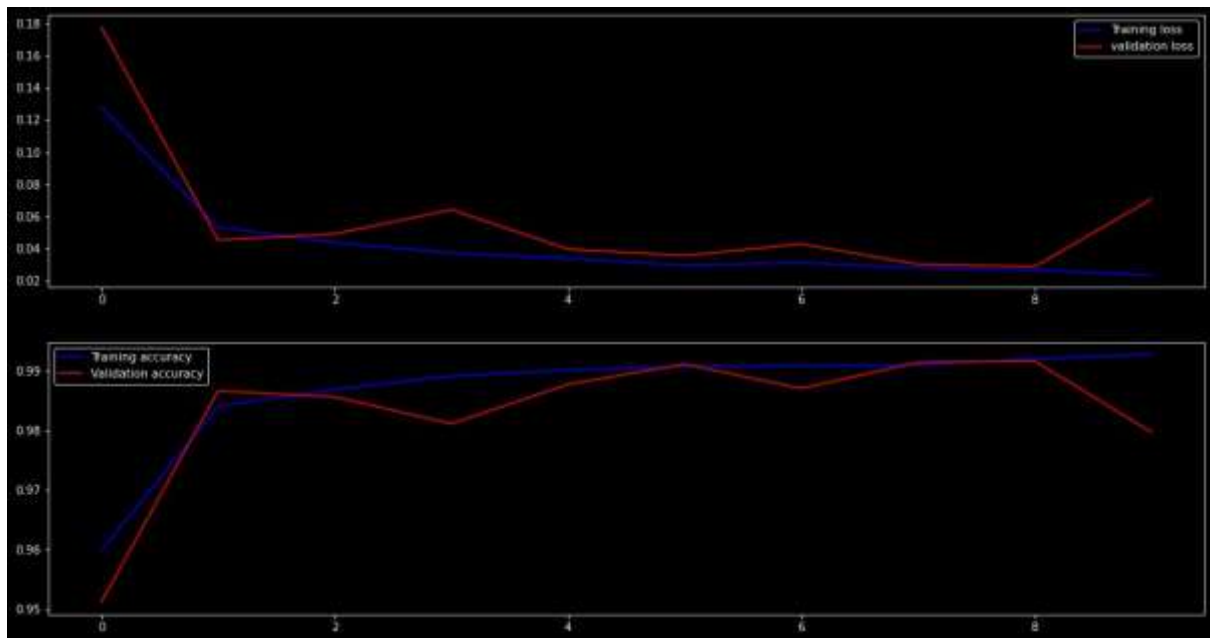
#### 9.1 PERFORMANCE METRICS

##### 9.1.1 MODEL SUMMARY

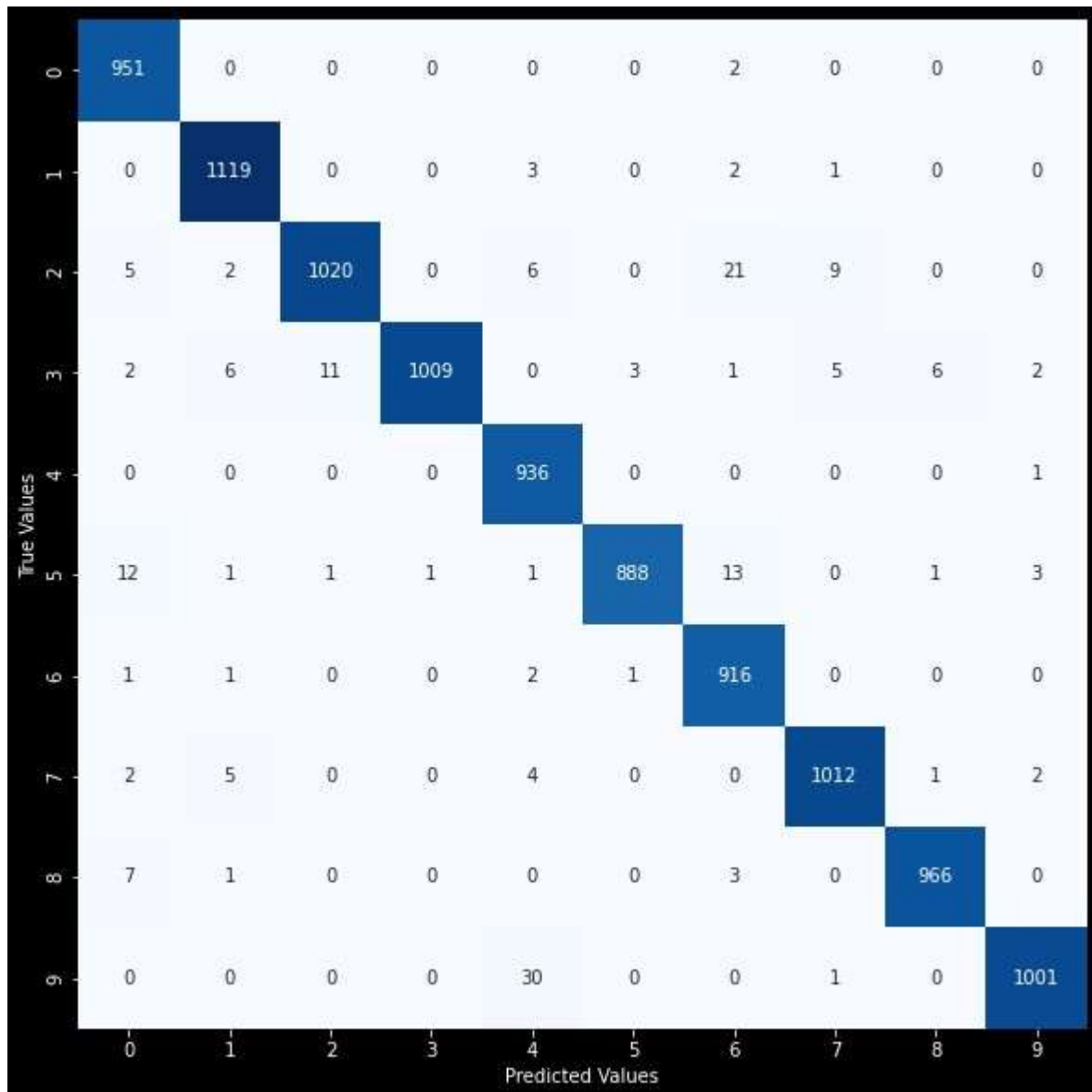
Model: "sequential"		
Layer (type)	Output Shape	Param #
=====		
conv2d (Conv2D)	(None, 26, 26, 64)	640
conv2d_1 (Conv2D)	(None, 24, 24, 32)	18464
flatten (Flatten)	(None, 18432)	0
dense (Dense)	(None, 10)	184330
=====		
Total params: 203,434		
Trainable params: 203,434		
Non-trainable params: 0		
=====		

##### 9.1.2 ACCURACY

CONTENT	VALUE
Training Accuracy	99.14%
Training Loss	2.70%
Validation Accuracy	97.76%
Validation Loss	10.36%



### 9.1.3 CONFUSION MATRIX



### 9.1.4 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

### 9.1.5 APPLICATION TEST REPORT

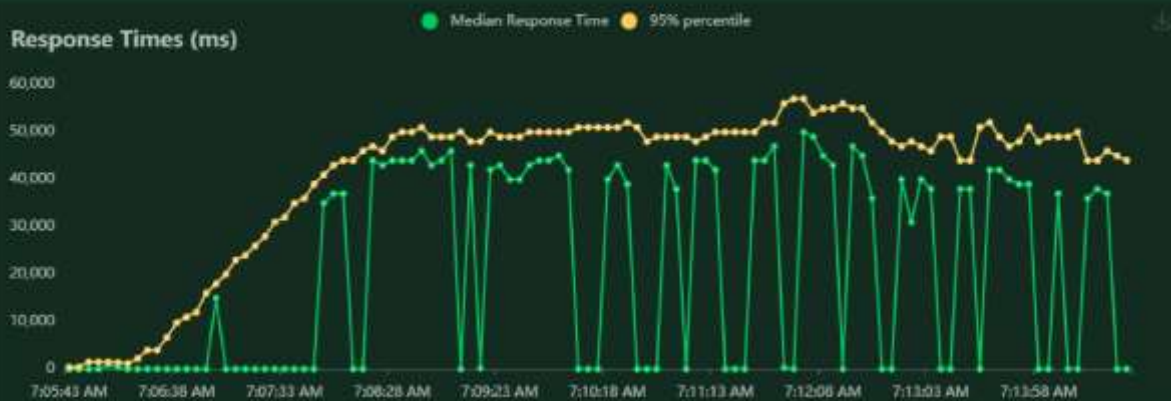
Locust Test Report									
During: 11/12/2022, 7:05:40 AM - 11/12/2022, 7:14:47 AM									
Target Host: http://127.0.0.1:5000/									
Script: locust.py									
Request Statistics									
Method	Name	# Requests	# Fails	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	RPS	Failures/s
GET	/	1043	0	13	4	290	1079	1.9	0.0
GET	/predict	1005	0	39648	385	59814	2670	1.8	0.0
Aggregated		2048	0	19462	4	59814	1859	3.7	0.0
Response Time Statistics									
Method	Name	50%ile (ms)	60%ile (ms)	70%ile (ms)	80%ile (ms)	90%ile (ms)	95%ile (ms)	99%ile (ms)	100%ile (ms)
GET	/	10	11	13	15	19	22	62	290
GET	/predict	44000	46000	47000	48000	50000	52000	55000	60000
Aggregated		36	36000	43000	45000	48000	50000	54000	60000

## Charts

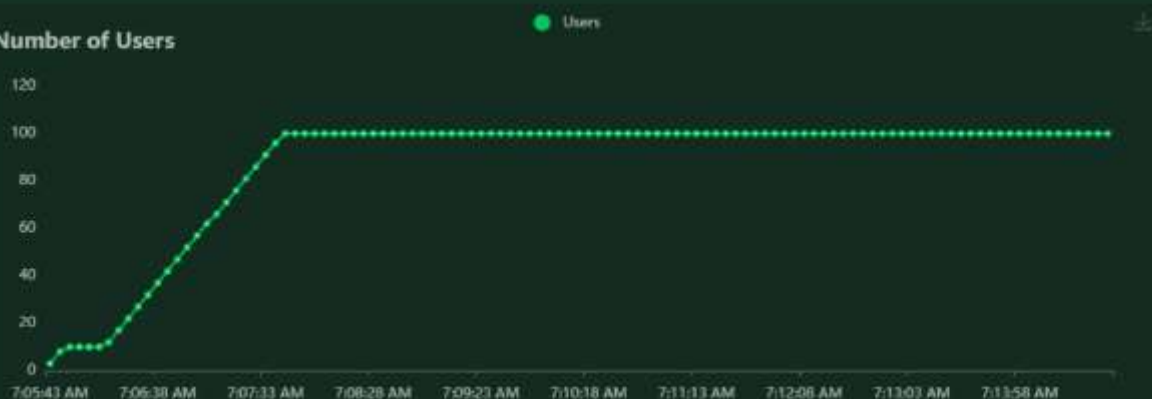
Total Requests per Second



Response Times (ms)



Number of Users



## CHAPTER 10

### ADVANTAGES & DISADVANTAGES

#### ADVANTAGES

- Reduces manual work
- More accurate than average human
- Capable of handling a lot of data
- Can be used anywhere from any device

## **DISADVANTAGES**

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

## **CHAPTER 11**

### **CONCLUSION**

This project demonstrated a web application that uses machine learning to recognise handwritten numbers. Flask, HTML, CSS, JavaScript, and a few other technologies were used to create this project. The model predicts the handwritten digit using a CNN network. During testing, the model achieved a 99.61% recognition rate. The proposed project is scalable and can easily handle a huge number of users.

Since it is a web application, it is compatible with any device that can run a browser. This project is extremely useful in real-world scenarios such as recognizing number plates of vehicles, processing bank cheque amounts, numeric entries in forms filled up by hand (tax forms) and so on. There is so much room for improvement, which can be implemented in subsequent versions.

## **CHAPTER 12**

### **FUTURE SCOPE**

This project is far from complete and there is a lot of room for improvement. Some of the improvements that can be made to this project are as follows:

- Add support to detect from digits multiple images and save the results
- Add support to detect multiple digits
- Improve model to detect digits from complex images
- Add support to different languages to help users from all over the world

This project has endless potential and can always be enhanced to become better. Implementing this concept in the real world will benefit several industries and reduce the workload on many workers, enhancing overall work efficiency.



## MODEL CREATION

```
# Load the necessary packages
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 the data
(X_train, y_train), (X_test, y_test) = mnist.load_data()

# Data pre-processing
X_train = X_train.reshape(60000, 28, 28, 1).astype('float32')
X_test = X_test.reshape(10000, 28, 28, 1).astype('float32')

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)
```

```
# Create the model
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"))

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

# Train the model
model.fit(X_train, Y_train, batch_size=32, epochs=5, validation_data=(X_test, Y_test))

# Evaluate the model
metrics = model.evaluate(X_test, Y_test, verbose=0)
print("Metrics (Test Loss & Test Accuracy): ")
print(metrics)

# Save the model
model.save("model.h5")
```

```
# Test the saved model
model=load_model("model.h5")

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)
```

## FLASK APP

```

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

```

# Import necessary packages
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/model.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)

```

<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Handwritten Digit Recognition</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 Recognizer</h1>
        <h2 class="heading__sub">Easily analyze 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>
          
        </div>
      </div>
    </div>
  </body>
</html>

```

## HOME PAGE (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;
}

```

## HOME PAGE (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)

```

<html>
  <head>
    <title>Prediction | Handwritten Digit Recognition</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>Prediction</h1>
      <div class="result-wrapper">
        <div class="input-image-container">
          
        </div>
        <div class="result-container">
          <div class="value">{{best.0}}</div>
          <div class="accuracy">{{best.1}}%</div>
        </div>
      </div>
      <h1>Other Predictions</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>

```



<https://github.com/IBM-EPBL/IBM-Project-8789-1658929747>



**PROJECT DEMO**

[https://drive.google.com/drive/folders/1WbIBYv10LWImZB7xxbv6KairR9s9AJFq?usp=share\\_link](https://drive.google.com/drive/folders/1WbIBYv10LWImZB7xxbv6KairR9s9AJFq?usp=share_link)