

**TEAM ID : PNT2022TMID01617**

**PROJECT NAME : A Novel Method for Handwritten Digit Recognition System**

# **Project Report**

## **1. INTRODUCTION**

### **1.1 PROJECT OVERVIEW**

The ability of a computer to interpret manually written digits from various sources, such as messages, bank checks, papers, pictures, and so forth and in various situations, such as web-based handwriting recognition on PC tablets, identifying vehicle number plates, handling bank checks, and digits entered in any forms, is known as handwritten digit recognition. Machine learning offers several ways to recognise the manually written digits with less effort from humans

Deep Learning is a machine learning technique that teaches computers to do what comes naturally to people: learn by doing. Human efforts in seeing, learning, recognising, and many other areas can be reduced with the use of deep learning techniques. The computer learns to perform categorization tasks from images or content from any document using deep learning. Deep Learning models can achieve cutting-edge precision that surpasses human ability. The digit recognition model makes use of substantial datasets to identify digits from various sources.

## **2. 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. People can struggle to read others' handwriting. The handwritten digits are not always of the same size, width, or orientation as they differ from writing from person to person, so the general problem would be while classifying the digits.

### **2.1 REFERENCES**

**Handwritten Digit Recognition using CNN (2019)**

*Vijayalaxmi R Rudraswamimath 1, Bhavanishankar K2*

Digit Recognition is a noteworthy and important issue. As the manually written digits are not of a similar size, thickness, position and direction, in this manner, various difficulties must be considered to determine the issue of handwritten digit recognition. The uniqueness and assortment in the composition styles of various individuals additionally influence the example and presence of the digits.

### **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.

## **2.2 PROBLEM STATEMENT DEFINITION**

The problem statement is to classify handwritten digits. The goal is to take an image of a handwritten digit and determine what that digit and character is. It is easy for humans to perform a task accurately by practicing it repeatedly and memorizing it for the next time. Human brain can process and analyze images easily. Also, recognize the different elements present in the images.

Handwritten digit recognition is the capability of computer applications 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 shapes and sizes. The handwritten digit recognition system is a way to tackle this problem which uses the image of a digit and recognizes the digit present in the image. Convolutional Neural Network model created using Python library over the MNIST dataset to recognize handwritten digits. Handwriting number recognition is a challenging problem researchers have been researching in this area for so long, especially in recent years.

### 3.1 EMPATHY MAP CANVAS

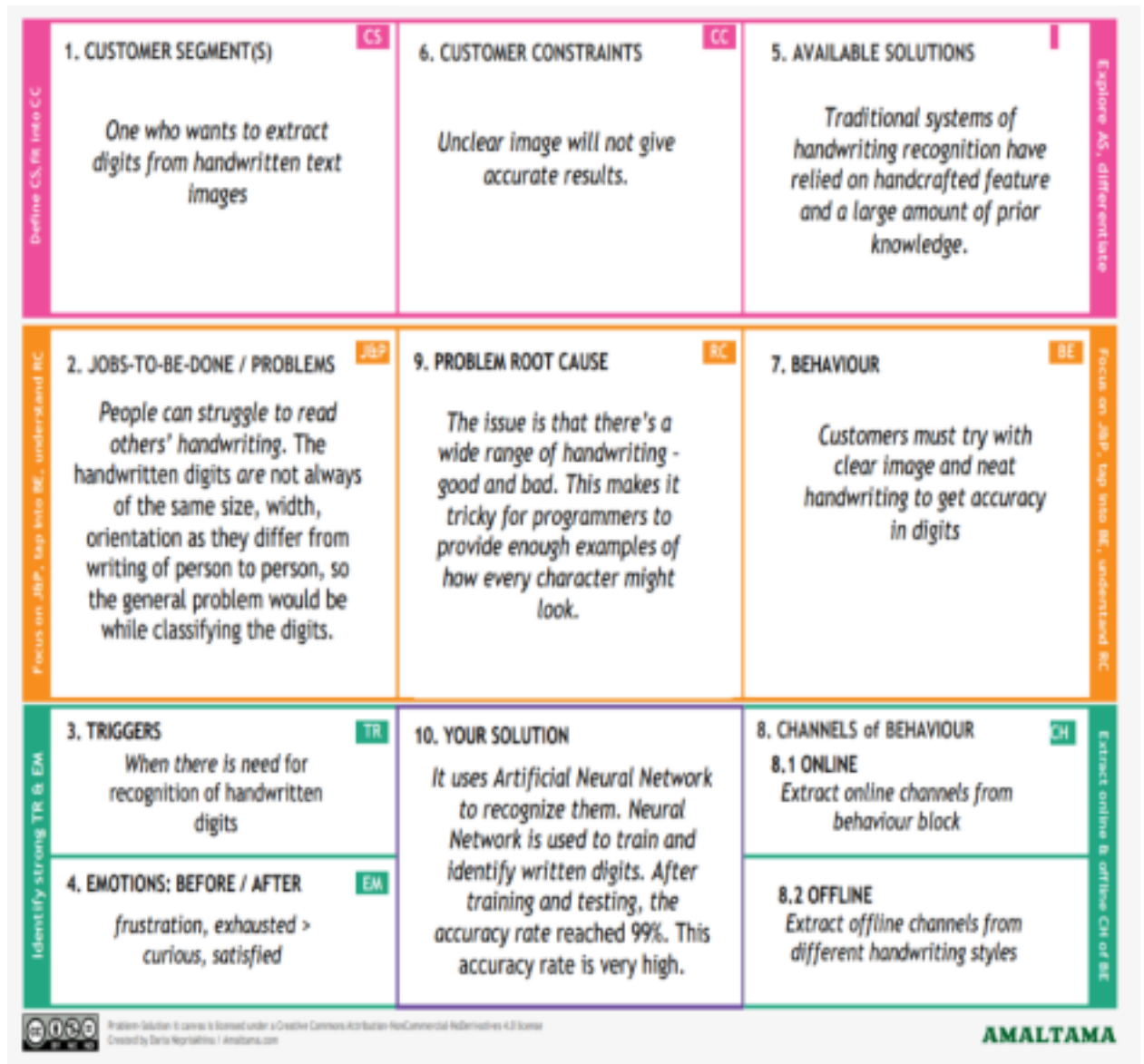
### 3.2 IDEATION AND BRAINSTORMING



### 3.2 PROPOSED SOLUTION

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	The problem statement aims at developing a novel handwritten recognition system using ML. The handwritten digit recognition system is a way to tackle the problem which uses the image of a digit and recognizes the digit present in the image .
2.	Idea / Solution description	Developing an AI predictive model to predict the handwritten digits and to construct a neural network with hidden layers and train to detect the digits.
3.	Novelty / Uniqueness	The system not only produces a classification of the digit but also a rich description of the <u>instantiation</u> parameters which can yield information such as the writing style.
4.	Social Impact / Customer Satisfaction	Handwritten digits can be recognized easily without any strenuous efforts. This reduces time and improves productivity for people.
5.	Business Model (Revenue Model)	It is used in the detection of vehicle numbers, banks for reading <u>cheques</u> , post offices for arranging letters, and many other tasks.
6.	Scalability of the Solution	To attain higher performances in the domain of character recognition and pattern recognition, due to its excellent feature extraction and working as best classifier characteristics. There is no limit in the number of digits that can be recognized.

### 3.2 PROBLEM SOLUTION FIT



## REQUIREMENT ANALYSIS

### 4.1 FUNCTIONAL REQUIREMENTS

FR No.	Functional Requirement
	Sub Requirement (Story / Sub-Task) (Epic)
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	Upload image Image upload via files Image upload via folders Image upload via drive Image upload via web Image upload via scan/camera
FR-4	Spelling support Identifies handwriting of different styles and fonts Spelling check
FR-5	Translation Handwritten digits from the image are extracted. Conversion of handwritten digits into machine readable form
FR-6	Log out Log out / sign out.

#### 4.2 NON-FUNCTIONAL REQUIREMENTS

FR No.	Non-Functional Requirement Description
NFR-1	<b>Usability</b> The proposed system gives good results for images that contain handwritten text written in different styles, different size and alignment with varying background
NFR-2	<b>Security</b> Only authorized people can access the system data and modify the database.
NFR-3	<b>Reliability</b> The Database is frequently updated with handwriting of different styles and size and will rollback when any update fails.

NFR-4	<b>Performance</b> The proposed system is advantageous as it uses fewer features to train the neural network, which results in faster convergence.
NFR-5	<b>Availability</b> The system functionality and services are available for use with all operations.

NFR-6	<b>Scalability</b> The website traffic limit must be scalable enough to support 2 lakhs users at a time
-------	---

## PROJECT DESIGN

### 5.1 DATA FLOW DIAGRAM

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

#### DFD Level-0

The DFD Level-0 consists of two external entities, the UI and the Output, along with a process, representing the CNN for Digit Recognition .Output is obtained after processing.



## DFD Level-1

The DFD Level-1 consists of 2 external entities, the GUI and the Output, along with five process blocks and 2 data stores MNIST data and the Input image store, representing the internal workings of the CNN for Digit Recognition System. Process block imports MNIST data from the library. Process block imports the image and processes it and sends it to the block where the regression model is built. It sends objects with probabilities to CNN where weights are updated and multiple layers are built. Block trains and evaluates the model to generate output.



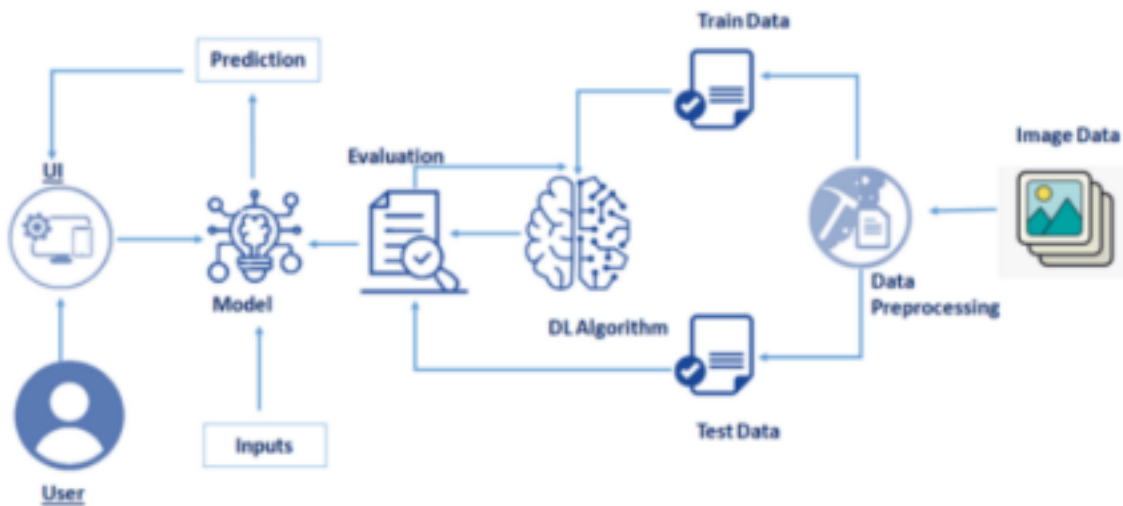
## DFD Level-2

The DFD Level-2 for import data (figure 4) consists of two external data and one entity UI along with three process blocks, representing the three functionalities of the CNN for Digit Recognition System. It imports data from MNIST data store and stores on the system.





## 5.2 SOLUTION & TECHNICAL ARCHITECTURE



## 5.3 COMPONENTS & TECHNOLOGIES:

S.N o	Component	Description Technology

1.	User Interface	How user interacts with application e.g., Mobile Application HTML, CSS, JavaScript
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	<div>File storage requirements</div> <div>IBM Block Storage or Other Storage Service or local file system</div>

#### 8. External API-1 IBM Weather API, etc.

		Purpose of External API used in the application
--	--	---

9.	Internet of Things Model	Purpose of AI Model is for integrating the sensors with a user interface IBM AI Platform
10.	Machine Learning Model	Purpose of Machine Learning Model Digit Recognition Model

## 5.4 USER STORIES

<b>User Type</b>	the application	USN-3
<b>Functional Requirement</b>	from anywhere	As a user, I should
<b>User Story Number</b>	and use on any devices	Viewing the User can
<b>User Story / Task</b>		access the application using the
<b>Acceptance Criteria</b>		browser on any device
<b>Priority Release</b>		

Customer  
Building the application

USN-1 As a user, I should be able to access

USN-2  
Uploading Image  
As a user, I should be able to upload images to predict the digits

User can

upload  
images

The result of  
High Sprint-4

High Sprint-2 High

Results		displayed
	USN-4 As a	The
Viewing Other	user, I should be	accuracy of otherMedium
Prediction	able to see	values must be Sprint-3 Sprint-1
be able to view	other close	displayed
the	predictions	
results	the prediction is	

Usage Instruction	USN-5 As a	<b>SPRINT PLANNING AND</b>
user, I		<b>ESTIMATION</b>

should have a  
usage instruction  
to know how to  
use the  
application

**6.PROJECT PLANNING**  
**& SCHEDULING 6.1**

The usage instruction is displayed  
on the home  
page  
Medium

### Sprint-3

#### Project Planning Phase

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

Date	18 October 2022
Team ID	PNT2022TMID01617
Project Name	A Novel Method for Handwritten Digit Recognition System
Maximum Marks	8 Marks

##### Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data Collection	USN-1	As a user, I can collect the dataset from various resources with different handwritings.	10	Low	Shobigaa Sri R K Sobika S Gopika Sri R Akhilan S
Sprint-1	Data Preprocessing	USN-2	As a user, I can load the dataset, handling the missing data, scaling and split data into train and test.	10	Medium	Shobigaa Sri R K Sobika S
Sprint-2	Model Building	USN-3	As a user, I will get an application with ML model which provides high accuracy of recognized handwritten digit.	5	High	Gopika Sri R Akhilan S
Sprint-2	Add CNN layers	USN-4	Creating the model and adding the input, hidden, and output layers to it.	5	High	Shobigaa Sri R K Sobika S Gopika Sri R Akhilan S

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-2	Compiling the model	USN-5	With both the training data defined and model defined, it's time to configure the learning process.	2	Medium	Shobigaa Sri R K Sobika S Gopika Sei R Akhilan S
Sprint-2	Train & test the model	USN-6	As a user, let us train our model with our image dataset.	6	Medium	Shobigaa Sri R K Sobika S Gopika Sei R Akhilan S
Sprint-2	Save the model	USN-7	As a user, the model is saved & integrated with an android application or web application in order to predict something.	2	Low	Shobigaa Sri R K Sobika S Gopika Sei R Akhilan S
Sprint-3	Building UI Application	USN-8	As a user, I will upload the handwritten digit image to the application by clicking a upload button.	5	High	Shobigaa Sri R K Sobika S Gopika Sei R Akhilan S
Sprint-3		USN-9	As a user, I can know the details of the fundamental usage of the application.	5	Low	Shobigaa Sri R K Sobika S Gopika Sei R Akhilan S
Sprint-3		USN-10	As a user, I can see the predicted / recognized digits in the application.	5	Medium	Shobigaa Sri R K Sobika S Gopika Sei R Akhilan S
Sprint-4	Train the model on IBM	USN-11	As a user, I train the model on IBM and integrate flask/Django with scoring end point.	10	High	Shobigaa Sri R K Sobika S Gopika Sei R Akhilan S
Sprint-4	Cloud Deployment	USN-12	As a user, I can access the web application and make the use of the product from anywhere.	10	High	Shobigaa Sri R K Sobika S Gopika Sei R Akhilan S

## 6.2 SPRINT DELIVERY SCHEDULE

**Total Points    Sprint End Date    Completed (as    Sprint Release**

**Iteration    Sprint    Planned)    Scheduled End    Date (Actual)**

**Start Date    Story Points    Date)**

**Sprint**

Sprint-1    20 Days    Oct 2022    9 Oct 2022    20 29 Oct 2022    Sprint-2    20 Days    Oct 2022    5 Nov 2022    20 05 Nov 2022    Sprint-3    20 Days    Nov 2022    2 Nov 2022    20 12 Nov 2022    Sprint-4    20 Days    Nov 2022    9 Nov 2022    20 19 Nov 2022

## 6.3 REPORT FROM JIRA

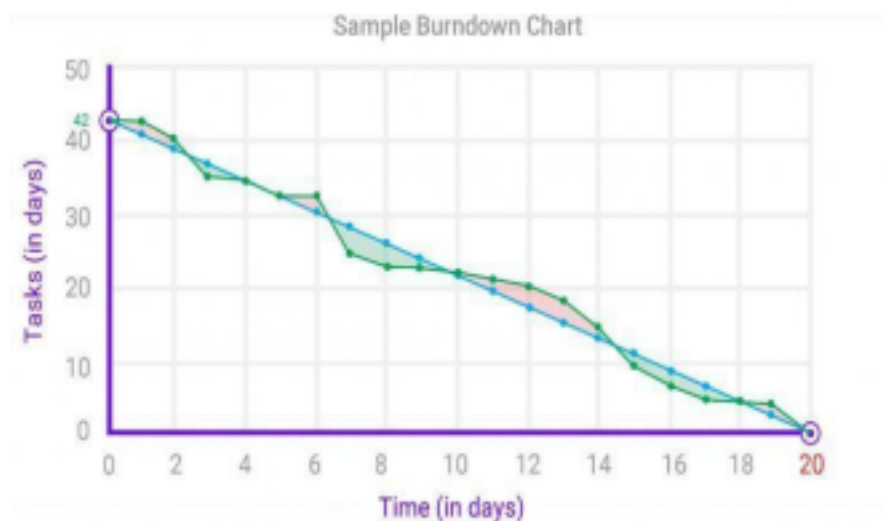
### Velocity:

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

$$\text{Average Velocity} = 20 / 6 = 3.33$$

## Burndown Chart:

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



## CODING & SOLUTION

### 7.1 FEATURE 1 – FLASK FILE UPLOADING

Handling file upload in Flask is very easy. It needs an HTML form with its enctype attribute set to 'multipart/form-data', posting the file to a URL. The URL handler fetches a file from request.files[] object and saves it to the upload folder.

```
numpy as np

import os

from PIL import Image

from flask import Flask, request, render_template

from werkzeug.utils import secure_filename

from keras.models import load_model
```



```

UPLOAD_FOLDER =
    'C:/Users/Dell/PycharmProjects/A-novel-method-for-digit-recognition-system/fl
a sk_app/uploads'
app = Flask(__name__)

app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER


model = load_model("mnistCNN.h5")


@app.route('/')

def index():

    return render_template('index.html')


@app.route('/predict', methods=['GET', 'POST'])

def upload():

    if request.method == "POST":

        f = request.files["image"]

        filepath = secure_filename(f.filename)

        f.save(os.path.join(app.config['UPLOAD_FOLDER'], filepath))

        upload_img = os.path.join(UPLOAD_FOLDER, filepath)

        img = Image.open(upload_img).convert("L") # convert image to
monochrome

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

        im2arr = np.array(img) # converting to image

```

```
im2arr = im2arr.reshape(1, 28, 28, 1) # reshaping according to our
requirement
```

```
pred = model.predict(im2arr)
```

```
num = np.argmax(pred, axis=1) # printing our Labels
```

```
return render_template('predict.html', num=str(num[0]))
```

```
if __name__ == '__main__':
```

```
app.run(debug=True, threaded=False)
```

## 7.2 FEATURE 2 – UPLOAD IMAGE WITH PREVIEW

A preview refers to a feature that lets you glimpse or view something in part or whole without it being opened. A picture preview would show a small version of the picture and give you a good idea what each picture is without opening each picture it is a useful feature created using JavaScript.

```
id="content">

<div class="leftside">

<form action="/predict" method="POST" enctype="multipart/form-data">

<label>Select a image:</label>

<input id="image" type="file" name="image" accept="image/png,
image/jpeg" onchange="preview()"><br><br>
<img id="frame" src="" width="100px" height="100px"/>

<div class="buttons_div">
```

```

        <button type="submit" class="btn btn-dark"
id="predict_button">Predict</button>

        <button type="button" class="btn btn-dark" id="clear_button"> Clear
</button>

    </div>

</form>

</div>

</section>

    frame.src=URL.createObjectURL(event.target.files[0]);

}

```

### 7.3 FEATURE 3 – CLEAR IMAGE

This feature can be used to clear the image if we uploaded a wrong image or if we need to change the image. The clear button clears both the image value and the preview of the image in the script tag.

# TESTING

## 8.1 TEST CASES

Feature Type	Component			
	Test Scenario	Expected Result	Actual Result	
UI Home Page	Verify UI elements in the Home			
	The Home page must be displayed			
	Working as			
HP_T C_001	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	Page properly expected x 1801 and 768 x 630
HP_T C_002	UI Home Page			
HP_T C_003				

		their file	uploaded to the application successfully	working properly	All the routes should properly work
HP_TC_004	Functional	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	Working as expected
HP_TC_005	Functional	Home Page		Working as expected	
		Check if the page redirects to the result page once the input is given	The page should redirect to the results page	BE_TC_001	Functional
	Functional	Home Page		Backend	
	Functional	Check if user can upload	The input image should be	Working as expected	Check if all the routes are
			Functional		
M_TC_001	M_TC_002	Functional	Functional	Model Check if the model predicts the digit	
			Model Check if the model can handle various image sizes		
	M_TC_003				

				predictions are displayed properly
Model Check if the model can handle complex input image	Working as expected	RP_TC_003	Result Page Verify UI elements in the Result Page	The Result page must be displayed properly
The model should rescale the image and predict the results	The model fails to identify the digit since the model is	RP_TC_004	Check if the input image is displayed properly	The input image should be displayed properly
The model should predict the number		Result Page		
	RP_TC_001			
		Result Page		
The model should predict the number in the complex image	RP_TC_002		Check if the result is displayed properly	The result should be displayed properly
Working as expected		Result Page		
			Check if the other	The other

predictions should be displayed	Working as expected		exceeds the display container	
properly not built to handle such data		The size of the input image	Working as expected	Working as expected

## 8.2 USER ACCEPTANCE TESTING

### 8.2.1 DEFECT ANALYSIS

Resolution Severity 1 Severity 2 Severity 3 Severity 4

By Design 1 0 1 0 Duplicate 0 0 0 0 External 0 0 2 0 Fixed 4 1 0 1

Not Reproduced 0 0 0 1 Skipped 0 0 0 1

Won't Fix 1 0 1 0 Total 6 1 4 3

### 8.2.2 TEST CASE ANALYSIS

Section Total Cases Not Tested  
Client Application 1 0 0 3 7

Security 2 0 1 1 Performance 3 0 1 2

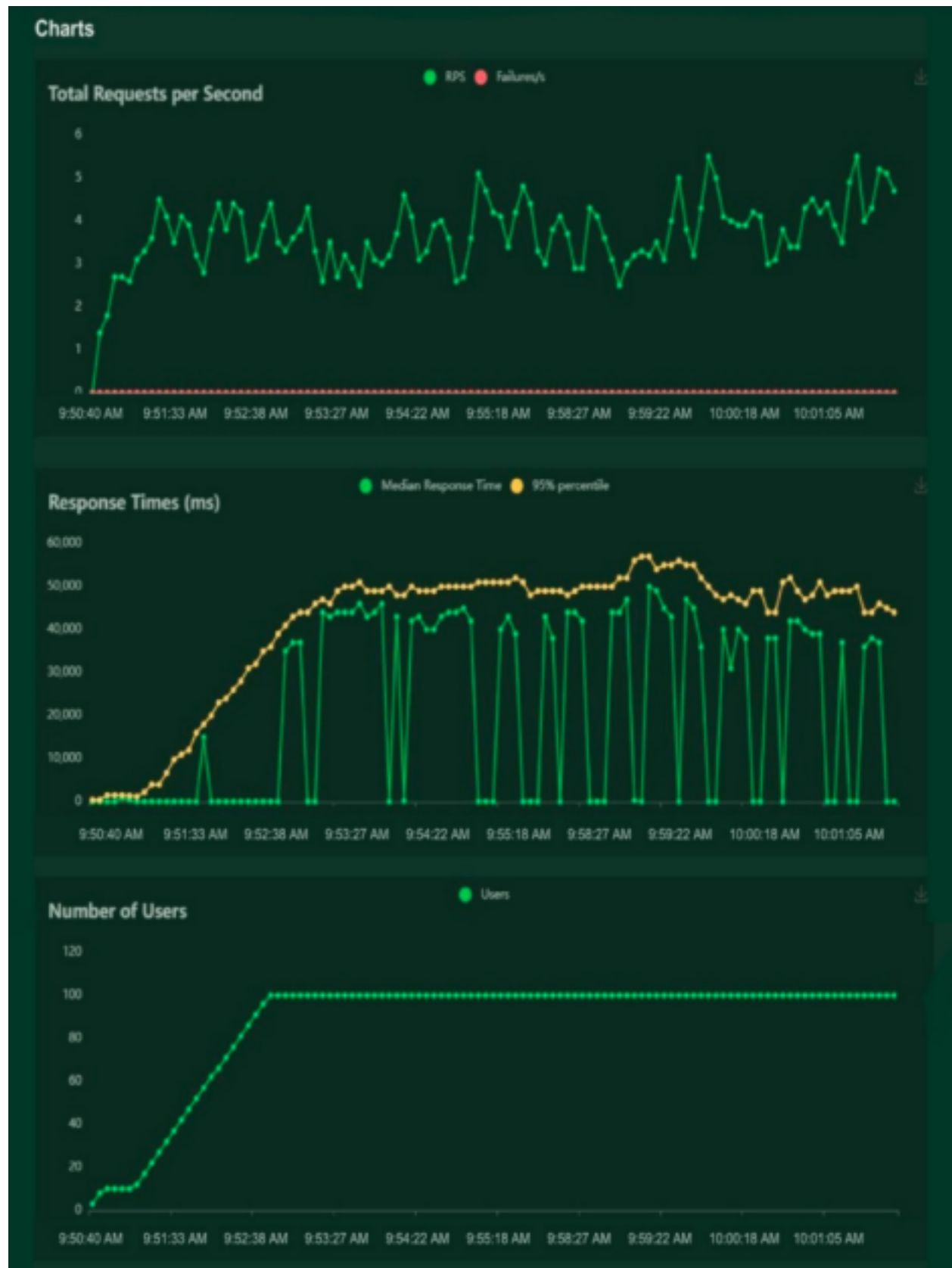
Exception Reporting 2 0 0 2

## 9.RESULTS

### 9.1 PERFORMANCE METRICS

Locust Test Report									
During: 11/15/2022, 9:50:40 AM - 11/15/2022, 10:01:59 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





## 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
- Neural Network is used to train and identify written digits for greater efficiency.
- The accuracy rate is very high.
- Speed of data entry
- It is much easier to dictate the machine than to write
- Easier data retrieval

## **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
- There is a wide range of handwriting – good and bad.
- It is tricky for programmers to provide enough examples of how every character might look.
- Customers must try with clear image and neat handwriting to get accuracy in digits.
- Unclear image will not give accurate results.

## **11.CONCLUSION**

Convolutional Neural Network (CNN) adds its significant improvement to the Manuscript Document Recognition System. This paper tells us the effectiveness of CNN-based classification of data and pre-processing methods. Our model clearly sees handwriting and achieves outgoing predictions of up to 82.16% and accurate predictions of up to 69.16%. However the model can be continuously developed using multiple training samples. This will help the model to learn as well as generalize better. There are many images in the training set that are completely invisible to the human eye.

This project demonstrated a web application that uses machine learning to recognize 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.

Through extensive evaluation using a MNIST dataset, the present work suggests the role of various hyper-parameters. Fine tuning of hyper-parameters is essential in improving the performance of CNN architecture. We achieved a recognition rate of 99.89% with the Adam optimizer for the MNIST database, which is better than all previously reported results. The effect of increasing the number of convolutional layers in CNN architecture on the performance of handwritten digit recognition is clearly presented through the experiments .

## **12.FUTURE SCOPE**

This project can be enhanced with a great field of machine learning and artificial intelligence. The world can think of a software which can recognize the text from a picture and can show it to the others, for example a shop name detector. Or this project can be extended to a greater concept of all the character sets in the world. This project has not gone for the total English alphabet because there will be more and many more training sets and testing values that the neural network model will not be enough to detect. Think of a AI modeled car sensor going with a direction modeling on the roadside, the user shall give only the destination.

All of these enhancements is an application of the texture analysis where advanced image

processing, Neural network model for training and advanced AI concepts will come. These applications can be modeled further. As this project is fully done by free and available resources and packages this can be also a limitation of the project. The fund is very important because all machine learning libraries and advanced packages are not available for free. Unlike those, most of the visualizing platforms on which developers are doing some work like Watson Studio or Aws. These all are mainly paid platforms where a lot of ML projects are going on.

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.