

## **Project Report Format**

### **1. INTRODUCTION**

- 1.1 Project Overview
- 1.2 Purpose

### **2. LITERATURE SURVEY**

- 2.1 Existing problem
- 2.2 References
- 2.3 Problem Statement Definition

### **3. IDEATION & PROPOSED SOLUTION**

- 3.1 Empathy Map Canvas
- 3.2 Ideation & Brainstorming
- 3.3 Proposed Solution
- 3.4 Problem Solution fit

### **4. REQUIREMENT ANALYSIS**

- 4.1 Functional requirement
- 4.2 Non-Functional requirements

### **5. PROJECT DESIGN**

- 5.1 Data Flow Diagrams
- 5.2 Solution & Technical Architecture
- 5.3 User Stories

### **6. PROJECT PLANNING & SCHEDULING**

- 6.1 Sprint Planning & Estimation
- 6.2 Sprint Delivery Schedule
- 6.3 Reports from JIRA

### **7. CODING & SOLUTIONING (Explain the features added in the project along with code)**

- 7.1 Feature 1
- 7.2 Feature 2
- 7.3 Database Schema (if Applicable)

### **8. TESTING**

- 8.1 Test Cases
- 8.2 User Acceptance Testing

## **9. RESULTS**

### 9.1 Performance Metrics

## **10.ADVANTAGES & DISADVANTAGES**

## **11.CONCLUSION**

## **12.FUTURE SCOPE**

## **13.APPENDIX**

Source Code

GitHub & Project Demo Link

# **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 recognize 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.

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.

Handwritten digit recognition using MNIST dataset is a major project made with the help of Neural Network. It basically detects the scanned images of handwritten 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 margin 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 REFERENCE**

- Handwritten digit recognition is a translational problem of human writings into machine editable text format. Author – Ganesh Khekare Year-2021.
- A human may not appreciate how difficult it is to solve handwriting. Author - B.Ali Alameer. Year – 2021.
- Handwritten Digit Recognition using Machine Learning . Author – Nazmule Siddique Year – 2021.
- A progressive learning approach for low resource handwritten text recognition. Author- Mohamed Ali Souibgui Year-2022.
- 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

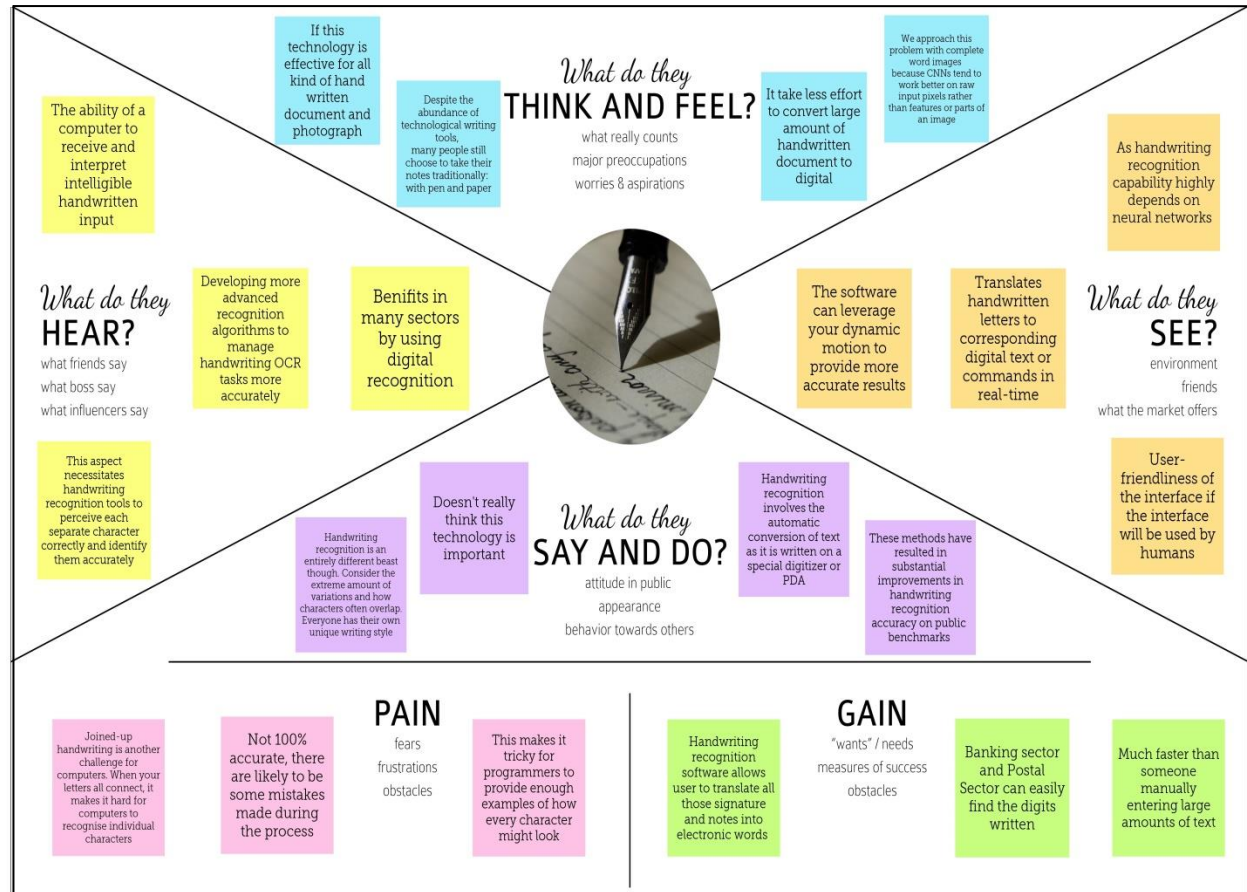
## 2.3 PROBLEM STATEMENT DEFINITION

- In this paper, Convolutional Neural Networks (CNN) is presented for handwritten digit recognition. Edges of the extracted graph were categorized into shape types and vertices were extracted from each of the edges and their layer wise evaluation using deep learning.
- The challenge of visual pattern recognition is only apparent to develop a computer system to read handwriting. The artificial neural networks approach is considered as the best way to develop systems for recognizing handwriting.
- Handwritten character recognition is one of the practically important issues in pattern recognition applications.
- Handwritten text recognition in low resource scenarios, such as manuscripts with rare alphabets, is a challenging problem. we propose a few-shot learning-based handwriting recognition approach that significantly reduces the human annotation process, by requiring only a few images of each alphabet symbols.

# CHAPTER 3

## IDEATION & PROPOSED SOLUTION

### 3.1 Empathy Map Canvas



## 3.2 IDEATION & BRAINSTORMING

### 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  
2-8 people recommended

[Share template feedback](#)

**Before you collaborate**

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

1. **Team gathering**  
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.
2. **Set the goal**  
How about the problem you'll be focusing on solving in the brainstorming session?
3. **Learn how to use the facilitation tools**  
Use the Facilitation Superpowers to let it happen and facilitate sessions.

[Open article](#)

**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

**PROBLEMS**

- \* Papers are replaced by digital documents for various reasons. However, we still see a lot of paper documentation in our daily life.
- \* Machines do not have the ability to understand what has been written on those physical papers. Converting handwritten characters to digital characters has been a tough problem in the past and continues to be.
- \* We cannot efficiently process those physical documents with computers unless we can convert them to digital documents.

**2 Brainstorm**

Write down any ideas that come to mind that address your problem statement.

10 minutes

**Tip**  
When you collect a sticky note, add the word "problem" to identify what is your starting point.

**Sarun**

The aim of a handwriting digit recognition system is to convert handwritten digits into machine-readable format.

Features are based on shape analysis of the digit image and extract shape or slope information.

Much faster than someone manually entering large amount of data.

Obtained by a simple image matching method.

**Kaviarasan**

The system not only recognizes handwritten digits but also a digit recognition system that recognizes the shape of the digit in the writing style.

Deep learning is also used to recognize handwritten digits. It is a neural network that automatically figures out the patterns that they have learned.

Convolutional neural networks (CNNs) are very effective in performing the pattern of handwritten characters/words.

**Santhosh kumar**

The proposed system gives the highest recognition accuracy of 99.9% using Support Vector Machine (SVM) classifier.

The system uses a novel pre-processing Multi-Correlation (MC) features for the recognition purpose.

A network's recognition accuracy increases by incorporating irreducible architecture.

Handwritten digits are images in the form of 28x28 pixel-scale coordinates of images.

**Lokesh**

CNN architecture is preferred in order to achieve the accuracy of the system.

These methods provide a substantial improvement in handwriting recognition.

Handwritten digits are images in the form of 28x28 pixel-scale coordinates of images.

**Next time**

Need more ideas? Use a custom report and a custom report to get more ideas and a custom report to get more ideas.

[Share template](#)

**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 can break it up into smaller sub-groups.

10 minutes

**Tip**  
Add color-coded tags to sticky notes to make it easier to find, organize, and categorize related ideas to monitor when your work.

**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**  
Each of these ideas is important, but without the ability to make them happen, they are not the most useful.

**Feasibility**  
Regardless of their importance, which ideas are more feasible than others? Give ideas which are feasible.

**Tip**  
Participants can use their own ideas to place sticky notes on the grid. This tool can be used to help you see the most important ideas in the room.

**5 After you collaborate**

You can export the mural as an image or pdf to share with members of your company who might find it helpful.

**Quick add-ons**

1. **Share the mural**  
Share a new link to the mural with stakeholders to keep them in the loop about the outcomes of the session.
2. **Export the mural**  
Export a copy of the mural as a PNG or PDF to attach to emails, include in slides, or save in your drive.

**Keep moving forward**

1. **Strategy blueprint**  
Define the components of a new idea or strategy.  
[Open the template](#)
2. **Customer experience journey map**  
Understand customer needs, motivations, and emotions for an experience.  
[Open the template](#)
3. **Strengths, weaknesses, opportunities & threats**  
Identify strengths, weaknesses, opportunities, and threats that can develop a plan.  
[Open the template](#)

[Share template feedback](#)

### 3.3 Proposed Solution

S.no	Parameter	Description
1.	Problem Statement (Problem to be solved)	<p>In this digital world, everything including documents, notes is kept in digital form the requirement of converting these digital documents into processed information is in demand. Handwriting recognition is one of the compelling research works going on because every individual in this world has their own style of writing. It is the capability of the computer to identify and understand handwritten digits or characters automatically. Because of the progress in the field of science and technology, everything is being digitalized to reduce human effort. Hence, there comes a need for handwritten digit recognition in many real-time applications.</p>
2.	Idea / Solution description	<p>Convolutional Neural Networks (CNN) becomes one of the most appealing approaches and has been an ultimate factor in a variety of recent success and challenging machine learning applications such as challenge ImageNet object Detection image segmentation and face recognition. Therefore, we choose CNN for our challenging tasks of image classification. We can use it for hand writing digits recognition which is one of high academic and business transactions. There are many applications of hand writing digit recognition in our real life purposes. Precisely, we can use it in banks for reading checks, post offices for sorting letter, and many other related works.</p>



3.	Novelty / Uniqueness	<p>Handwritten digit recognition is the capability of computer applications to recognize the human the 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. Handwritten Digit Recognition is the capability of a computer to fete the mortal handwritten integers from different sources like images, papers, touch defenses, etc. And classify them into 10 predefined classes(0-9). This is the existing method along with this we add some features to make our project unique among them. We create a model that recognize multiple-digits present in the image in offline mode. The future scope of this method is, we can recognize number in License plate, bank cheques and postal mail sorting. We get a predicted result in two manner one is, the recognized digits is showned in the interface and the another manner is, we can get a predicted result through voice mode. This means the model tell the multiple-digit in voice mode. This feature helps the old age people they are difficult in understanding handwritten digits, blind people and who contain eye sight issues.</p>
4.	Social Impact / Customer Satisfaction	<p>Digit recognition plays an role in the modern world. 'Digits' are a part of our everyday life, be it License plate on our cars or bike, the price of a product, speed limit plate on our cars or bike, the price of a product, speed limit on a road, or details associated with a bank account. In the case of a text which is unclear, it is easier to guess the digits in comparison to the alphabets. Machine</p>

		<p>Learning and Deep Learning are reducing human efforts in almost every field.</p> <p>Moreover, a solution achieved using ML and DL can power various applications at the same time, thereby reducing human effort and increasing the flexibility to use the solution. One such solution is a handwritten digit recognition system that can be used in postal mail sorting, bank check processing, form data entry, etc.</p>
5.	Business Model (Revenue Model)	<p>Digit recognition plays an important roles in many places. It is independent of environment, while using the recognizer we don't need the network. The benefits of hand written digit recognizer is high. In banking sector, it is very useful. It is used to recognize the account number, figure of cash and checks. It is also used to recognize the written digits on cash, deposit /withdrawal. So, the requirement of manpower is less, because the machine done the work of bank employees. So, we can earn the profit by using the hand written digit recognizer.</p>
6.	Scalability of the Solution	<p>To make the path toward digitalization clearly by providing high accuracy and faster computational for recognizing the handwritten digits. The present Neural Network as classifier, MNIST as dataset with suitable parameter for training and testing and frame work for hand written recognition. The aforementioned system successfully impart accuracy up to 99.20% which is higher than formally proposed scheme.</p>

## 3.4 Problem Solution fit

Define CS, fit into CC	<b>1. CUSTOMER SEGMENT(S)</b> <b>CS</b> Who is your customer? i.e. working parents of 0-5 y.o. kids  It is useful for <ul style="list-style-type: none"> <li>• Children to understand the digits</li> <li>• Person who are at industry side for recognizing various handwriting digits.</li> <li>• People working in bank, post offices</li> </ul>	<b>6. CUSTOMER CONSTRAINTS</b> <b>CC</b> What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices.  <ul style="list-style-type: none"> <li>• Time</li> <li>• Accuracy</li> <li>• Ease to access</li> <li>• Imperfect findings</li> </ul>	<b>5. AVAILABLE SOLUTIONS</b> <b>AS</b> Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper is an alternative to digital notetaking <ul style="list-style-type: none"> <li>• In past they get trouble in finding handwritten digits</li> <li>• Using this system, they can resolve this type of problems</li> <li>• Pros of this system is quick recognition and</li> <li>• Accurate prediction</li> <li>• Cons are network connection is mandatory for using this system</li> <li>• For using this system Knowledge about the system is required</li> </ul>	Explore AS, differentiate
	<b>2. JOBS-TO-BE-DONE / PROBLEMS</b> <b>J&amp;P</b> Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one, explore different sides.  There are different types of handwriting are in world. Each and every handwriting has its own characteristics and uniqueness.  Its difficult to understand the different people's handwriting digit.	<b>9. PROBLEM ROOT CAUSE</b> <b>RC</b> What is the real reason that this problem exists? What is the back story behind the need to do this job? i.e. customers have to do it because of the change in regulations.  <ul style="list-style-type: none"> <li>• Not everyone can understand everyone's handwriting</li> <li>• The handwriting is differed from person to person</li> <li>• So, it is difficult to recognize the digits</li> <li>• To solve this problem this system has developed</li> </ul>	<b>7. BEHAVIOUR</b> <b>BE</b> What does your customer do to address the problem and get the job done? i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)  To address the problem, they can take a snap of the handwritten digit and upload it in the software	
Identify strong TR & EM	<b>3. TRIGGERS</b> <b>TR</b> What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news.  <ul style="list-style-type: none"> <li>• By word of mouth</li> <li>• Good user experience</li> </ul>	<b>10. YOUR SOLUTION</b> <b>SL</b> If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality. If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.  <ul style="list-style-type: none"> <li>• A novel method for handwritten digit recognition system helps in recognizing the handwritten digits that uses MNIST dataset for training the model.</li> <li>• The model gets the image of the handwritten digits and recognizes the handwritten digits.</li> <li>• CNN algorithm is used over the MNIST dataset to recognize the handwritten digits.</li> </ul>	<b>8. CHANNELS of BEHAVIOUR</b> <b>CH</b> <b>8.1 ONLINE</b> What kind of actions do customers take online? Extract online channels from #7  <b>ONLINE</b> In online they can upload the handwritten picture and yield output	Extract online & offline CH of BE
	<b>4. EMOTIONS: BEFORE / AFTER</b> <b>EM</b> How do customers feel when they face a problem or a job and afterwards? i.e. lost, insecure > confident, in control - use it in your communication strategy & design. <ul style="list-style-type: none"> <li>• It is a quite irritating and frustrating while manually convert the handwritten digits</li> <li>• By using our system, user can save the time and reduce the error occur on recognition</li> </ul>		<b>8.2 OFFLINE</b> What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.  In offline they can ask their neighbors to scribble the digits to find them	

## CHAPTER 4

### REQUIREMENT ANALYSIS

#### 4.1 Functional Requirements

FR NO.	Functional Requirement (Epic)	Description
FR-1	User Registration	Registration through Gmail
FR-2	User Confirmation	Confirmation via Email
FR-3	User Login	Login via registered Username and Password
FR-4	Uploading images	Able to input the handwritten images into the application
FR-5	Input correlation	Image Correlation is a technique used to recognize characters from images. Collecting data and prepare it for training
FR-6	Feature extraction	Feature extraction is analysing the images and deriving some characteristics from these images that identify each specific element Feature extraction is analysing the images and deriving some characteristics from these images that identify each specific element
FR-7	Recognizing digits	Display the recognized digits from the input images to the user.

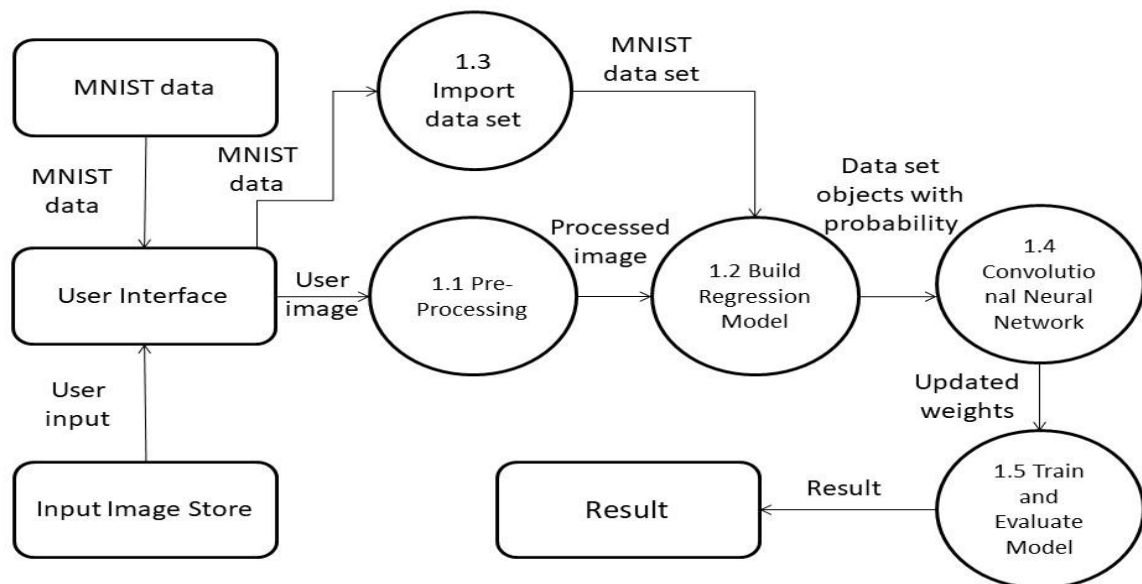
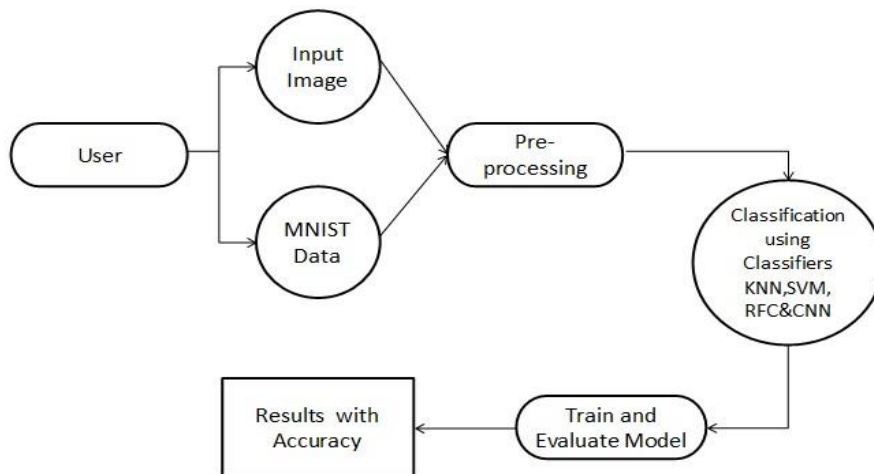
## 4.2 Non-Functional Requirements

FR No.	Non-Functional Requirement (Epic)	Description
FR-1	Usability	The application needs to respond smoothly so that the user can use the application effectively and need to be an user friendly application. To recognize the digits from bank cheque, papers, numeric entry in forms etc.
FR-2	Security	Ensure the security by authenticating the users using their username and password.
FR-3	Reliability	This software will work reliably for low resolution images and not for graphical images.
FR-4	Performance	Needs to respond fast and provide the output even for complex handwritings. The input image will be recognized with an accuracy of about 90% and more
FR-5	Availability	Need to available for all users at any time and can able to process the handwritten image as input to the application easily
FR-6	Scalability	It consist thousands of handwritten digits that have been used in the development of programs
		It is able to handle N numbers of users at the same time with faster response and recognize the digits effectively.

# CHAPTER 5

## PROJECT DESIGN

### 5.1 DATA FLOW DIAGRAM



Data Flow Diagram(Level-0)

## 5.2 SOLUTION & TECHNICAL ARCHITECTURE

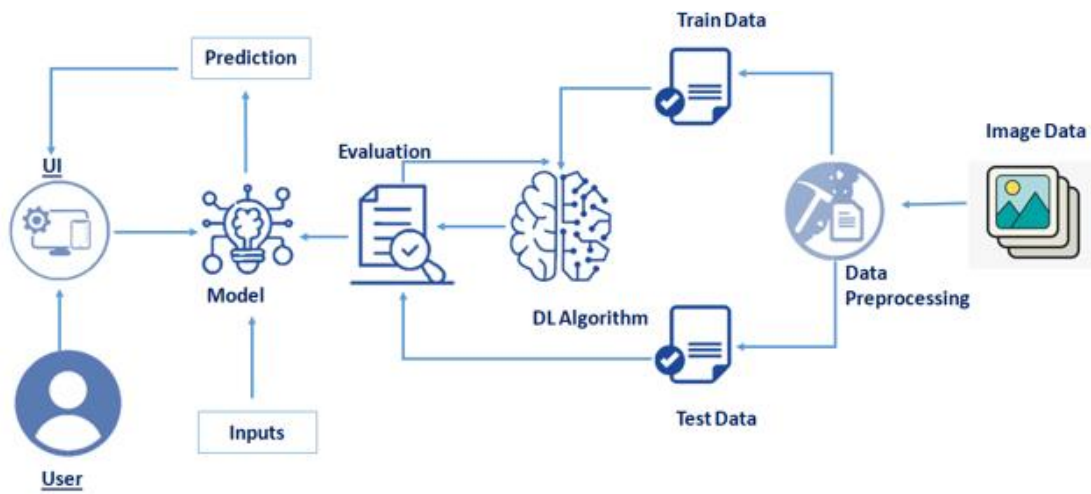


Fig. Solution Architecture

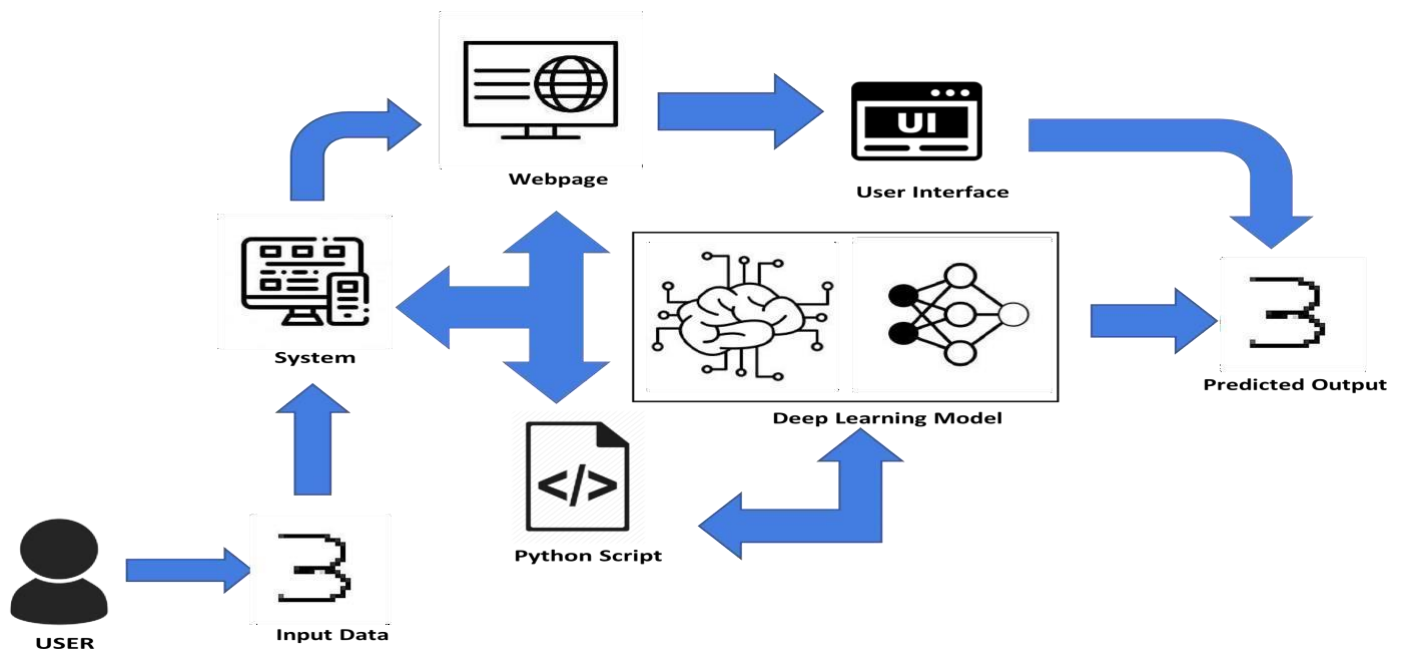


Fig. Technical Architecture

### 5.3 USER STORIES

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 view the guide and awareness to use this application.	I can view the awareness to use this application by a practical method.	Low	Sprint-2
		USN-4	As a user, I can read the instructions to use this application	I can read instructions to use it in user-friendly method.	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	I can access the application	High	Sprint-1
	About	USN-6	I can click on the "About" to get the idea of a handwritten digit recognition tool for recognition of digits.	I can get an idea about the project	Low	Sprint-1



Customer (Web user)	Predict	USN-7	As a user I can upload my handwritten digits images to be recognised from the computer.	I can choose any image from my device	High	Sprint-2
Customer Care Executive		USN-8	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
Administrator	Launch	USN-9	As a user, I can upload my handwritten digit images to be recognised from the computer.	I can choose and upload the image from the system storage and also in any virtual storage	Medium	Sprint-3
		USN-10	I can scan one page at once.	I can get the recognised digits from the input given	High	Sprint-4
	Recognize	USN-11	As a user I can turn on the camera using the input button.	I can get the input to be digitized.	High	Sprint-3
		USN-12	As a user, I can use the web application virtually anywhere.	I can use the application portably anywhere.	High	Sprint-1
		USN-13	As it is open source, I can use it cost freely.	I can use it without any payment to be paid for it to access	Medium	Sprint-2

## CHAPTER 6

### PROJECT PLANNING AND SCHEDULING

#### 6.1 SPRINT PLANNING AND ESTIMATION

Sp ri nt	Functional Re quirement (Ep ic)	User Stor yNu mbe r	User Story / Task	Sto ry Poi nts	Pri ori ty	Team Member s
Sp rin t-1	Data Collection	USN-1	As a user, I can collect the dataset from various resources with different handwritings.	10	Lo w	Santhosh Kumar D Lokesh B
Sp rin t-1	Data Pre processing	USN-2	As a user, I can load the dataset, handling the missing data, scaling and split data into tr ain and test.	10	Me diu m	Santhosh Kumar D Kaviarasan A
Sp rin t-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	Hig h	Sarun A Lokesh B Kaviara san A
Sp rin t-2	Add CNN layers	USN-4	Creating the model and addin g the input, hidden, and output layers to it.	5	Hig h	Sarun A Santhosh Kumar DKaviarasan A
Sp rin t-2	Compiling the model	USN-5	With both the training data defined and model defined, it's t ime to configure the learning process.	2	Me diu m	Sarun A Santhosh Kuma r D
Sp rin t-2	Train & test the model	USN-6	As a user, let us train our model with our image dataset.	6	Me diu m	Sar un A Lo kes hB

Sp rin t-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	Santhosh Kumar D
Sp rin t-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	Sarun A Kavi arasan A
Sp rin t-3		USN-9	As a user, I can know the details of the fundamental usage of the application.	5	Low	Lokesh B
Sp rin t-3		USN-10	As a user, I can see the predicted / recognized digits in the application.	5	Medium	Lokesh B S anthosh Kumar D
Sp rin t-4	Train the model on IBM	USN-11	As a user, I train the model on IBM and integrate flask/Django with scoring endpoint.	10	High	Sarun A Kavi arasan A
Sp rin t-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	Sarun A Kav arasan A

## 6.2 SPRINT DELIVERY SCHEDULE

<b>Sprint</b>	<b>Total Story Points</b>	<b>Duration</b>	<b>Sprint Start Date</b>	<b>Sprint End Date(Planned)</b>	<b>Story Points Completed (as on Planned EndDate)</b>	<b>Sprint Release Date(Actual)</b>
Sprint-1	20	6 Days	24 Oct2022	29 Oct2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	04 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	18 Nov 2022

# CHAPTER 7

## CODING AND SOLUTIONING

```
app.py > upload
1  import numpy as np
2  import os
3  from PIL import Image
4  from flask import Flask, request, render_template, url_for
5  from werkzeug.utils import secure_filename, redirect
6  from event.pywsgi import WSGIServer
7  from keras.models import load_model
8  from keras.preprocessing import image
9  from flask import send_from_directory
10
```

```
10
11  UPLOAD_FOLDER = 'C:/Users/Guru/flask/data'
12
13
14  app = Flask(__name__)
15  app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER
16
17  model = load_model("models/mnistCNN.h5")
18
19  @app.route('/')
20  @app.route('/index.html')
21  def index():
22      return render_template('index.html')
23
24
```

```
24
25  @app.route('/predict', methods=['GET', 'POST'])
26  def upload():
27      if request.method == "POST":
28          f = request.files["image"]
29          filepath = secure_filename(f.filename)
30          f.save(os.path.join(app.config['UPLOAD_FOLDER'], filepath))
31
32          upload_img = os.path.join(UPLOAD_FOLDER, filepath)
33          img = Image.open(upload_img).convert("L") # convert image to monochrome
34          img = img.resize((28, 28)) # resizing of input image
35
36          im2arr = np.array(img) # converting to image
37          im2arr = im2arr.reshape(1, 28, 28, 1) # reshaping according to our requirement
38
39          pred = model.predict(im2arr)
40
41          num = np.argmax(pred, axis=1) # printing our Labels
42
43          return render_template('web.html', num=str(num[0]))
44
45
46  if __name__ == '__main__':
47      app.run(debug=True, threaded=False)
48
```

## CHAPTER 8

### TESTING

#### 8.1 TEST CASE

Test caseID	FeatureType	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	FAIL
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 an 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

### 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

### TEST CASE ANALYSIS

Section	TotalCases	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

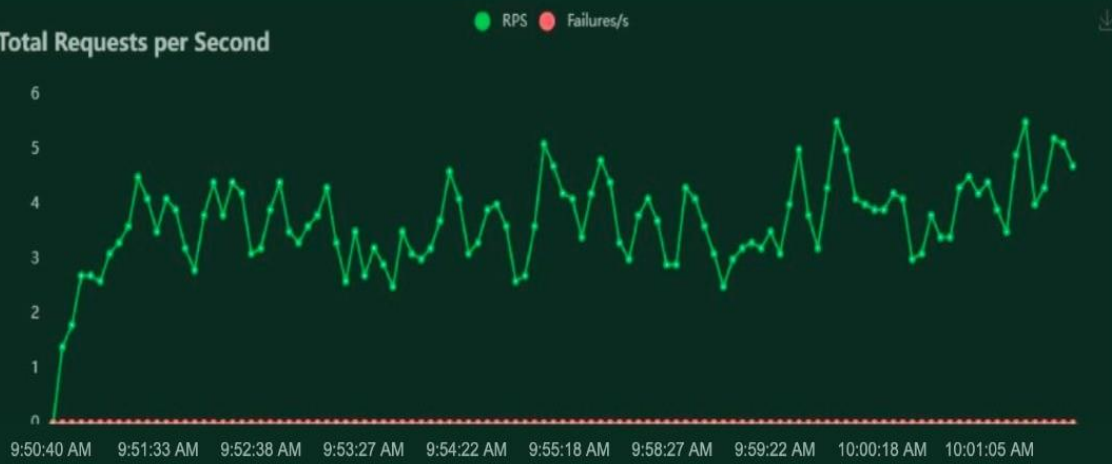
## RESULT

### 9.1 PERFORMANCE METRICS

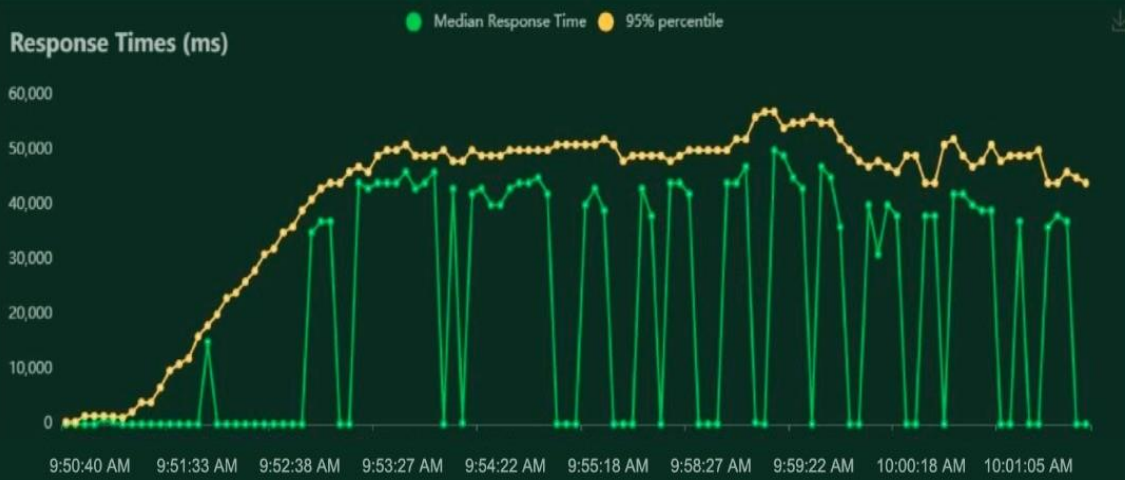
Locust Test Report									
During: 11/15/2022, 9:50:40 AM - 11/15/2022, 10:01:59 AM									
Target Host: http://127.0.0.1:5000/									
Script: locust.py									
Request Statistics									
Method	Name	# Requests	# Fails	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	RPS	Failures/s
GET	/	1043	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

## Charts

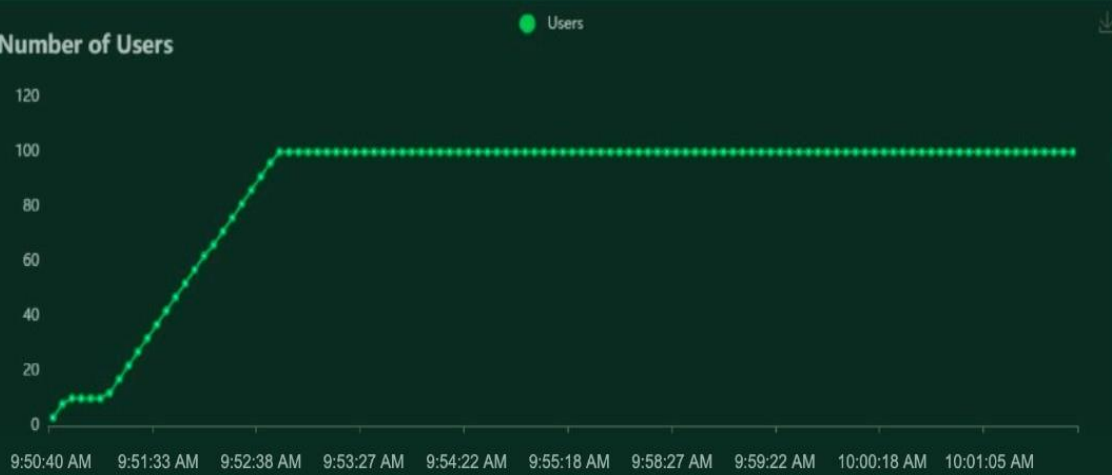
### Total Requests per Second



### Response Times (ms)



### Number of Users



## **CHAPTER 10**

### **ADVANTAGE AND DISADVANTAGE**

#### **ADVANTAGES**

1. Reduces manual work
2. More accurate than average human
3. Capable of handling a lot of data
4. Can be used anywhere from any device

#### **DISADVANTAGES**

1. Cannot handle complex data
2. All the data must be in digital format
3. Requires a high performance server for faster predictions
4. 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 numberplates 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:

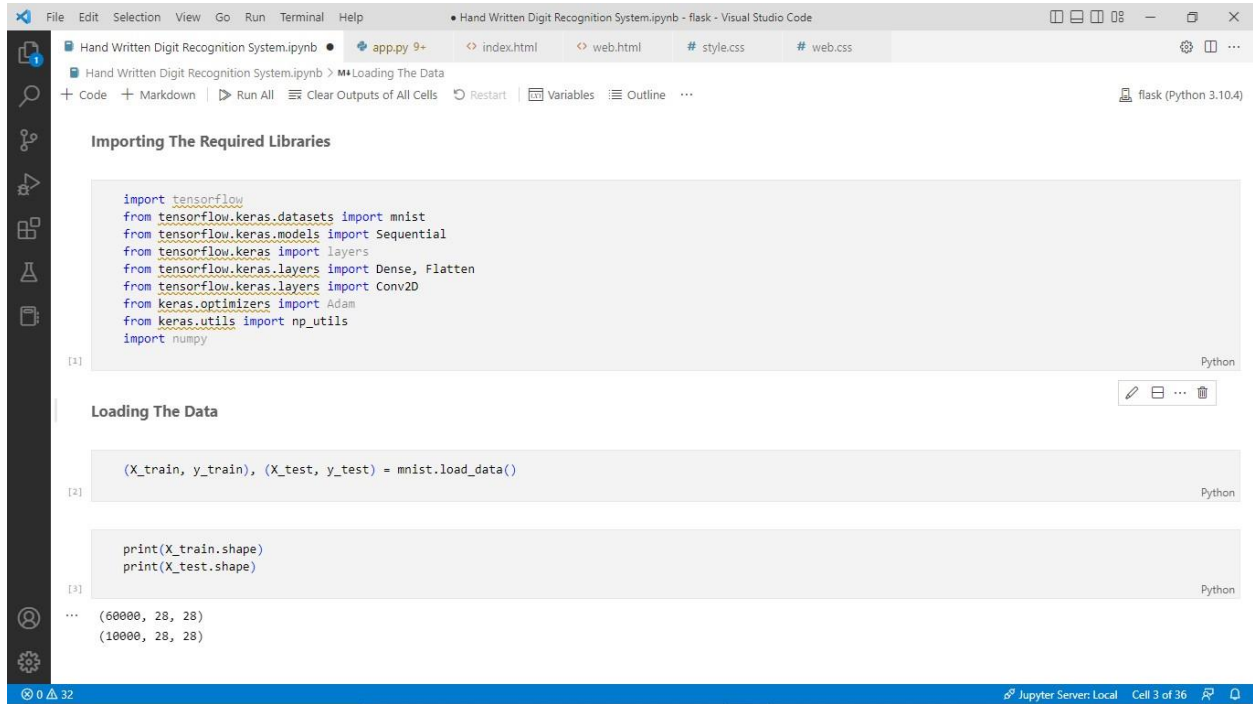
1. Add support to detect from digits multiple images and save the results
2. Add support to detect multiple digits
3. Improve model to detect digits from complex images
4. Add support to different languages to help users from all over the world

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.

# CHAPTER 13

## APPENDIX

### SOURCE CODE MODEL BUILDING



Hand Written Digit Recognition System.ipynb - flask - Visual Studio Code

File Edit Selection View Go Run Terminal Help

Hand Written Digit Recognition System.ipynb • app.py 9+ index.html web.html style.css web.css

Hand Written Digit Recognition System.ipynb > Loading The Data

+ Code + Markdown Run All Clear Outputs of All Cells Restart Variables Outline ...

flask (Python 3.10.4)

#### Importing The Required Libraries

```
import tensorflow
from tensorflow.keras.datasets import mnist
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 keras.optimizers import Adam
from keras.utils import np_utils
import numpy
```

[1] Python

#### Loading The Data

```
(X_train, y_train), (X_test, y_test) = mnist.load_data()
```

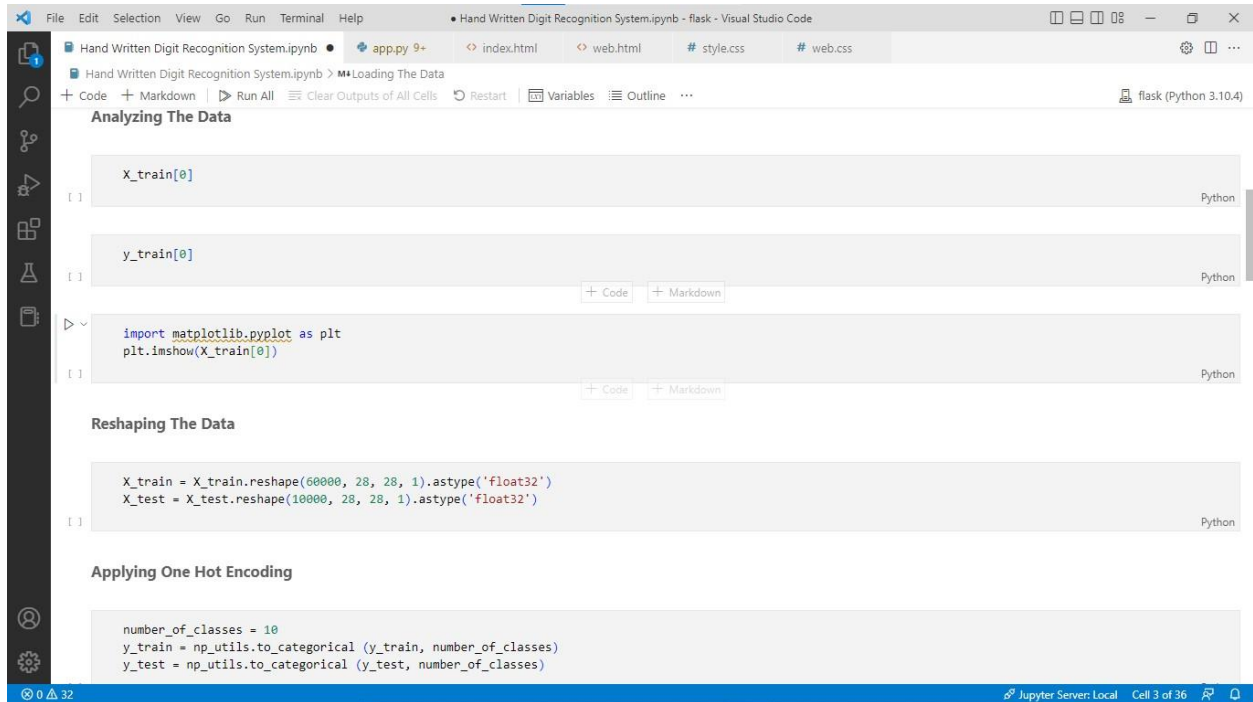
[2] Python

```
print(X_train.shape)
print(X_test.shape)
```

[3] Python

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

0 32 Jupyter Server: Local Cell 3 of 36



Hand Written Digit Recognition System.ipynb - flask - Visual Studio Code

File Edit Selection View Go Run Terminal Help

Hand Written Digit Recognition System.ipynb • app.py 9+ index.html web.html style.css web.css

Hand Written Digit Recognition System.ipynb > Loading The Data

+ Code + Markdown Run All Clear Outputs of All Cells Restart Variables Outline ...

flask (Python 3.10.4)

#### Analyzing The Data

```
X_train[0]
```

[ ] Python

```
y_train[0]
```

[ ] Python

+ Code + Markdown

```
import matplotlib.pyplot as plt
plt.imshow(X_train[0])
```

[ ] Python

+ Code + Markdown

#### 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')
```

[ ] Python

#### Applying One Hot Encoding

```
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)
```

0 32 Jupyter Server: Local Cell 3 of 36

Hand Written Digit Recognition System.ipynb - flask - Visual Studio Code

Hand Written Digit Recognition System.ipynb • Loading The Data

# Model Building

## 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(number_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=10, batch_size=32)
```

## Observing The Metrics

Jupyter Server: Local Cell 3 of 36

Hand Written Digit Recognition System.ipynb - flask - Visual Studio Code

Hand Written Digit Recognition System.ipynb • Loading The Data

## Observing The Metrics

```
metrics = model.evaluate(X_test, y_test, verbose=0)
print("Metrics (Test loss & Test Accuracy): ")
print(metrics)
```

## Test The Model

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

```
import numpy as np
print(np.argmax(prediction, axis=1))
print(y_test[:4])
```

## Observing The Metrics

```
metrics = model.evaluate(X_test, y_test, verbose=0)
print("Metrics (Test loss & Test Accuracy): ")
print(metrics)
```

Jupyter Server: Local Cell 3 of 36

Hand Written Digit Recognition System.ipynb - flask - Visual Studio Code

Hand Written Digit Recognition System.ipynb > Loading The Data

Code + Markdown Run All Clear Outputs of All Cells Restart Variables Outline ... flask (Python 3.10.4)

### Observing The Metrics

```
metrics = model.evaluate(X_test, y_test, verbose=0)
print("Metrics (Test loss & Test Accuracy): ")
print(metrics)
```

### Test The Model

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

```
import numpy as np
print (np.argmax(prediction, axis=1))
print (y_test[:4])
```

### Save The Model

```
model.save('models/mnistCNN.h5')
```

32 Jupyter Server: Local Cell 3 of 36

Hand Written Digit Recognition System.ipynb - flask - Visual Studio Code

Hand Written Digit Recognition System.ipynb > Loading The Data

Code + Markdown Run All Clear Outputs of All Cells Restart Variables Outline ... flask (Python 3.10.4)

### Test With Saved Model

```
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
import numpy as np

model = load_model('C:/Users/Guru/Handwritten Digit Recognition System/models/mnistCNN.h5')
img = image.load_img('C:/Users/Guru/Handwritten Digit Recognition System/data/image-2.jpg', target_size=(28,28), grayscale=True)
img = image.img_to_array(img)
print(img.shape)
x = np.expand_dims(img,axis=0)
print(x.shape)
print('*'*50)
print(model.predict(x))
print('*'*50)
print(np.round_(model.predict(x)))
```

32 Jupyter Server: Local Cell 3 of 36



## APPLICATION

### APP.PY

```
import numpy as np
import os
from PIL import Image
from flask import Flask, request, render_template, url_for
from werkzeug.utils import secure_filename, redirect
from gevent.pywsgi import WSGIServer
from keras.models import load_model
from keras.preprocessing import image
from flask import send_from_directory
UPLOAD_FOLDER = 'C:/Users/Guru/flask/data'
app = Flask(__name__)
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER
model = load_model("models/mnistCNN.h5")
@app.route('/')
@app.route('/index.html')
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('web.html', num=str(num[0]))
if __name__ == '__main__':
    app.run(debug=True, threaded=False)
```

## HOME PAGE INDEX.HTML

```
<html>
<head>
  <title>Digit Recognition</title>
  <meta name="viewport" content="width=device-width">
  <link rel="stylesheet" type="text/css" href= "{{ url_for('static',
filename='style.css') }}">

  <link href="https://fonts.googleapis.com/css2?family=Prompt:wght@600&displa
y=swap" rel="stylesheet">
  <link href="https://fonts.googleapis.com/css2?family=Varela+Round&display=s
wap" rel="stylesheet">
  <link href="https://fonts.googleapis.com/css2?family=Source+Code+Pro:wght@
500&display=swap" rel="stylesheet">
  <link href="https://fonts.googleapis.com/css?family=Calistoga|Josefin+Sans:400,
700|Pacifico&display=swap" rel="stylesheet">
  <!-- bootstrap -->
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/cs
s/bootstrap.min.css" integrity="sha384-
ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2M
Zw1T" crossorigin="anonymous">
  <!--fontawesome -->
  <script src="https://kit.fontawesome.com/b3aed9cb07.js" crossorigin="anonymou
s"></script>
  <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-
q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo
" crossorigin="anonymous"></script>
  <script src="https://cdn.jsdelivr.net/npm/popper.js/1.14.7/umd/popper.m
in.js" integrity="sha384-
UO2eT0CpHqdsJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHND
z0W1" crossorigin="anonymous"></script>
  <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.j
s" integrity="sha384-
JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM
" crossorigin="anonymous"></script>
  <script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@latest"></script>
</head>
<script>
  function preview() {
```

```

    frame.src=URL.createObjectURL(event.target.files[0]);
}
$(document).ready(function() {
    $('#clear_button').on('click', function() {
        $('#image').val("");
        $('#frame').attr('src','');
    });
});
</script>
<body>
<h1 class="welcome">Handwritten Digit Recognition</h1>
<section id="title">
<div class = "splitleft">
<h4 class="heading">A Novel Method For Handwritten Digit Recognition</h4>
<br><br>
<p>
    This website is designed to predict the handwritten digit.
<br><br>
    Handwriting recognition is one of the compelling research works going on
    because every individual in this world
    has their own style of writing. It is the capability of the computer to identify
    and understand
    handwritten digits or characters automatically. Because of the progress in the
    field of science and technology,
    everything is being digitalized to reduce human effort.
<br><br>
    Hence, there comes a need for handwritten digit recognition in many real-time
    applications.
    MNIST data set is widely used for this recognition process and it has 70000
    handwritten digits.
    We use Artificial neural networks to train these images and build a deep
    learning model.
    Web application is created where the user can upload an image of a
    handwritten digit.
    This image is analyzed by the model and the detected result is returned on to
    UI
</p>
</div>
<div id="content">
<div class = "splitright">

```



```

ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvORxT2M
Zw1T" crossorigin="anonymous">
    <!--fontawesome -->
    <script src="https://kit.fontawesome.com/b3aed9cb07.js" crossorigin="anonymo
us"></script>
    <script src="https://code.jquery.com/jquery-
3.3.1.slim.min.js" integrity="sha384-
q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo
" crossorigin="anonymous"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.
min.js" integrity="sha384-
UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHND
z0W1" crossorigin="anonymous"></script>
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.
js" integrity="sha384-
JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM
" crossorigin="anonymous"></script>
    <script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@latest"></script>
</head>
<script>
    function preview() {
        frame.src=URL.createObjectURL(event.target.files[0]);
    }
    $(document).ready(function() {
        $('#clear_button').on('click', function() {
            $('#image').val("");
            $('#frame').attr('src', "");
        });
    });
</script>
<body>
    <div id="rectangle">
        <h1 id="ans">Predicted Number : {{num}}</h1>
        <button type="submit" class="btn btn-dark" id="goback" ><a href="/">
&nbsp&nbsp&nbsp Go Back &nbsp&nbsp&nbsp</a></button>
    </div>
    <div id="content">
        <div class = "splitright">
        <div class="rightside">
        <form action="/predict" method="POST" enctype="multipart/form-data">

```

```

<label style=" padding-left: 12px;">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="150px" height="150px"/>
<div class="buttons_div">
  <button type="submit" class="btn btn-
dark" id="predict_button">&nbsp;&nbsp;&nbsp;Predict &nbsp;&nbsp;&nbsp;</button>
  <button type="button" class="btn btn-
dark" id="clear_button">&nbsp;&nbsp;&nbsp;Clear &nbsp;&nbsp;&nbsp;</button>
</div>
</form>
</div>
</div>
</body>
</html>

```

## OUTPUT HOME PAGE





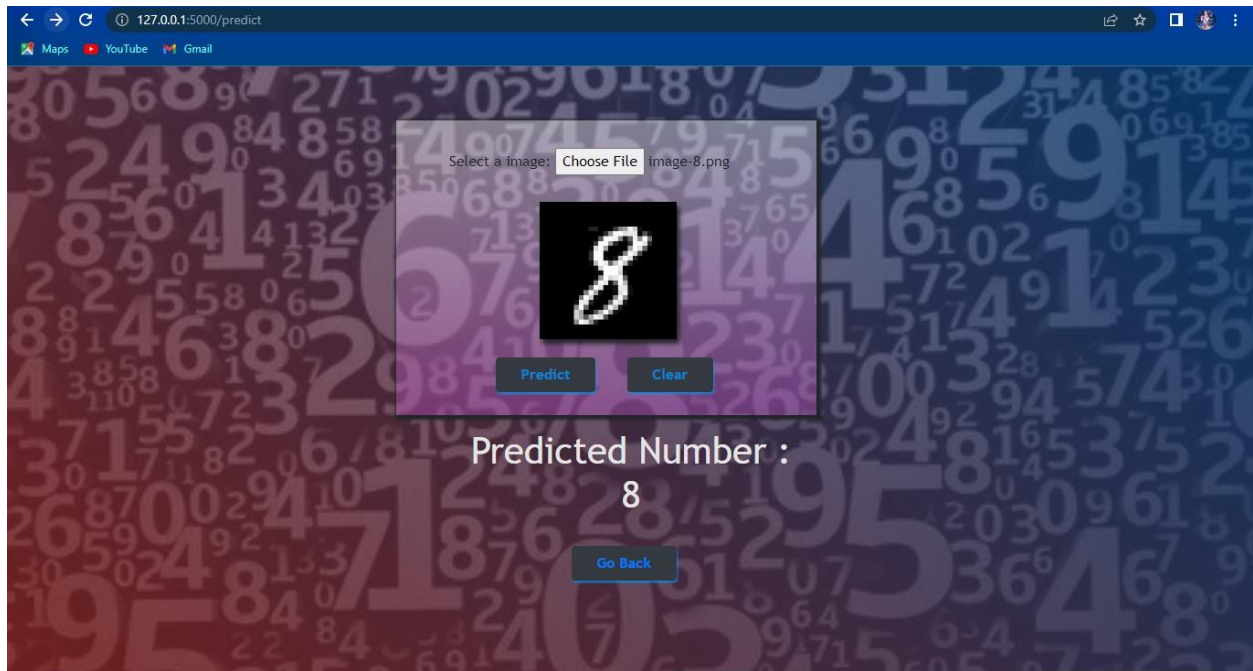
## UPLOADING INPUT IMAGE



## PROCESS THE IMAGE FOR PREDITION



## PREDICTED OUTPUT



## GITHUB LINK

<https://github.com/IBM-EPBL/IBM-Project-45034-1660727964>

## PROJECT DEMO LINK

<https://www.youtube.com/embed/4z1HzYKOwjE>