A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

IBM – DOCUMENTATION

UNDER THE GUIDANCE OF:

INDUSTRY MENTOR: Pradeepthi

FACULTY MENTOR: Sivagami V M

SUBMITTED BY:

Akshaya K B	190801004
Baraniya S	190801008
Mrithula Angeline M J	190801046
Poonam B N	190801053

OF:

Department of Information Technology

Sri Venkateswara College of Engineering

Pennalur, Sriperumbudur – 602117

Kancheepuram, Tamil Nadu

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW:

Handwritten Digit Recognition is the capacity of a computer to interpret the manually written digits, from various sources like messages, bank cheques, papers, pictures, and so forth and in various situations for web-based handwriting recognition on PC tablets, identifying number plates of vehicles, handling bank cheques, digits entered in any forms etc. Machine Learning provides various methods through which human efforts can be reduced in recognizing the manually written digits.

Deep Learning is a machine learning method that trains computers to do what easily falls into place for people: learning through examples. With the utilization of deep learning methods, humanattempts can be diminished in perceiving, learning, recognizing and in a lot more regions. Using deep learning, the computer learns to carry out classification works from pictures or contents fromany document. Deep Learning models can accomplish state- of-art accuracy, beyond the human level performance. The digit recognition model uses large datasets in order to recognize digits from distinctive sources.

1.2 PURPOSE:

The main objective was to actualize a pattern characterization method to perceive the handwritten digits provided in the MINIST data set of images of handwritten digits (0-9). The goal of our work is to create a model that will be able to recognize and classify the handwrittendigits from images by using concepts of Convolution Neural Network. Though the goal of our research is to create a model for digit recognition and classification, it can also be extended to letters and an individual's handwriting. With high accuracy rates, the model can solve a lot of real life problems.

The main applications are vehicle license-plate recognition, postal letter-sorting services, Cheque truncation system (CTS) scanning and historical document preservation in archaeology departments, old documents automation in libraries and banks, etc. All these areas deal with large databases and hence demand high recognition accuracy, lesser computational complexityand consistent performance of the recognition system.

CHAPTER 2

LITERATURE SURVEY

2.1 EXISTING PROBLEM:

The fundamental problem with handwritten digit recognition is that handwritten digits do not always have the same size, width, orientation, and margins since they vary from person to person. People can struggle to read others' handwriting. The handwritten digits are not always of the samesize, width, orientation as they differ from writing of person to person, sothe general problem would be whilexclassifying the digits.

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

2.2 REFERENCES:

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. It is the strategy for perceiving and arranging transcribeddigits. It has a wide range of applications, for example, programmed bank checks, postal locations and tax documents and soon. The aim of this project is to implement a classification algorithm to recognize the handwrittendigits. The after effects of probably the most broadly utilized Machine Learning Algorithms like SVM, KNN and RFC and with Deep Learning calculation like multilayer CNN utilizing Keras with Theano and Tensorflow. Utilizing these, the accuracy of 98.70% utilizing CNN (Keras + Theano) when contrasted with 97.91% utilizing SVM, 96.67% utilizing KNN, 96.89% utilizing RFC was obtained.

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

Pashine, Samay and Dixit, Ritik and Kushwah, Rishik

In this study, they developed three deep and machine learning-based models for handwritten digit recognition using MNIST datasets. In their research, they discovered that CNN produced the most precise outcomes for handwritten digit recognition. This led them tothe conclusion that CNNis the most effective 5 solution for all types of prediction issues, including those using picture data.Next, by comparing the execution times of the algorithms, they determined that increasing the number of epochs without changing the configuration of the algorithm is pointless due to the limitation of a certain model, and they discovered that beyond a certain number of epochs, the model begins over-fitting the dataset and provides biased predictions.

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

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

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

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

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

This paper's primary goal was to enhance handwritten digit recognition ability. To avoid difficult pre-processing, expensive feature extraction, and a complex ensemble (classifier combination) method of a standard recognition system, they examined different convolutional neural network variations. Their current work makes suggestions on the function of several hyper - parameters through thorough evaluation utilizing an MNIST dataset. They also confirmed that optimizing hyper-parameters is crucial for enhancing CNN architecture performance. With the Adam optimizer for the MNIST database, they were able to surpass many previously published results with a recognition rate of 99.89%. Through the trials, it is made 3 abundantly evident how the performance of handwritten digit recognition is affected by the number of convolutional layers in CNN architecture.

According to the paper, evolutionary algorithms can be explored for optimizing convolutional filter kernel sizes, CNN learning parameters, and the quantity of layers and learning rates.

A novel method for Handwritten Digit Recognition with Neural Networks(2020)

*Malothu Nagu*1, N Vijay Shankar#2, K.Annapurna*

Character recognition plays an important role in the modern world. It can solve more complex problems and makes humans' job easier. An example is handwritten character recognition. This is asystem widely used in the world to recognize zip code or postal code for mail sorting. There are different techniques that can be used to recognize handwritten characters. Two techniques researched in this paper are Pattern Recognition and Artificial Neural Network (ANN). Both techniques are defined and different methods for each technique is also discussed. Bayesian Decision theory, Nearest Neighbor rule, and Linear Classification or Discrimination is types of methods for Pattern Recognition. Shape recognition, Chinese Character and Handwritten Digit recognition uses Neural Network to recognize them. Neural Network is used to train and identify written digits. After trainingand testing, the accuracy rate reached 99%.This accuracy rate is very high.

2.3 PROBLEM STATEMENT DEFINITION

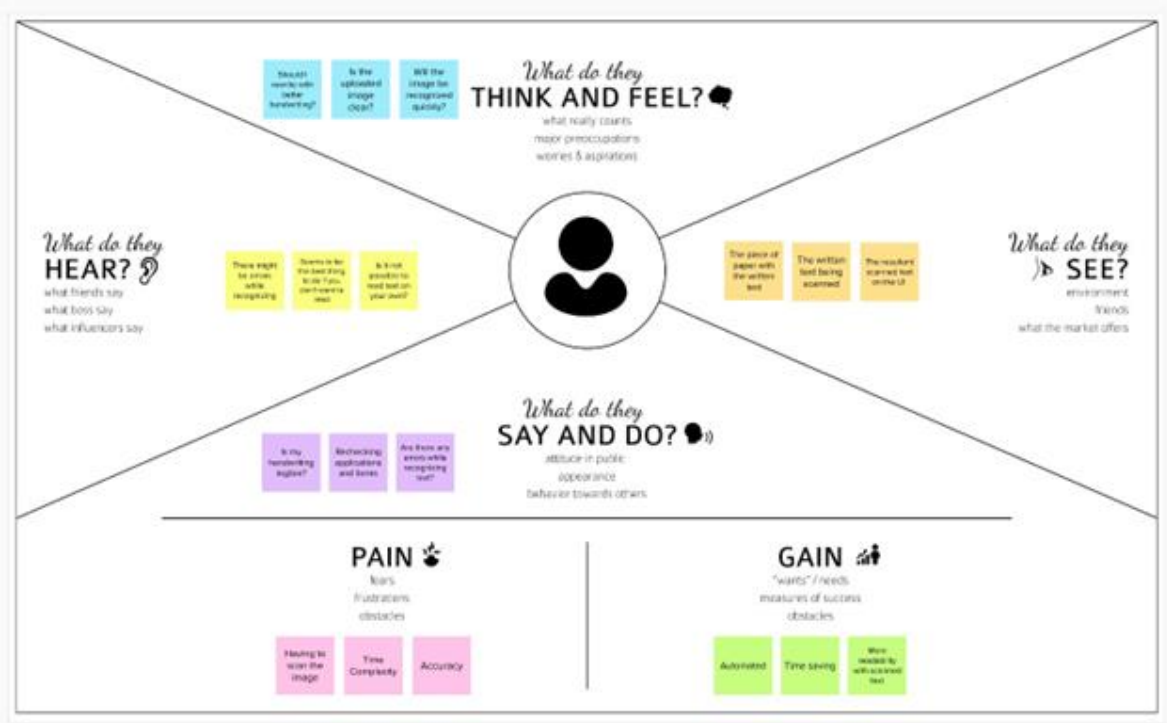
The problem statement is to classify handwritten digits. The goal is to take an image of a handwritten digit and determine what that digit and character is. It is easy for the human to perform a task accurately by practicing it repeatedly and memorizing it for the next time. Human brain can process and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

The handwritten digit recognition is the capability of computer applications to recognize the 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 had been research into this area for so long especially in the recent years.

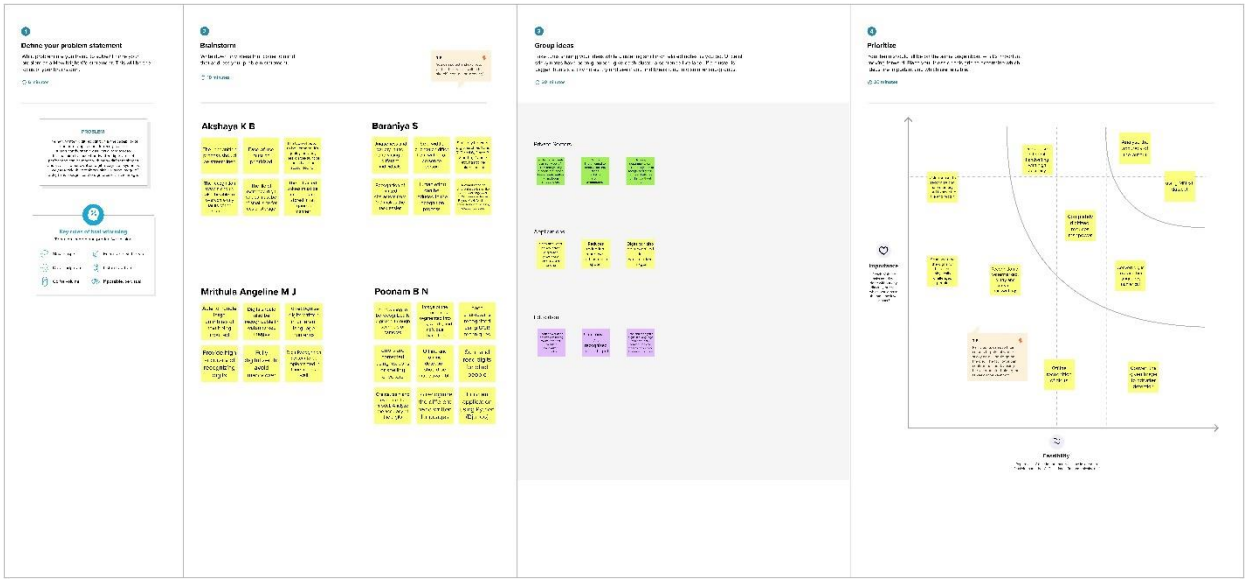
CHAPTER 3

IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS:



3.2 IDEATION & BRAINSTORMING:



3.3 PROPOSED SOLUTION:

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Digits are used extensively in everyday life, be it for distinguishing the bank accounts, or to document the marks scored by a set of students in a specific subject. These values usually have to be input into a file for easy computation, but the task gets tedious when alarge amount of values are involved, and it maylead to more errors during manual input.
2.	Idea / Solution description	This project aims to make the task of digit input easier, by reducing it to a simple procedure of scanning the paper the digits are written in, while the system handles the extraction and storage of the values. This avoids any chance of manual input errors during the process.
3.	Novelty / Uniqueness	If the image has various columns of digit data that needs to be scanned, it would be properly accounted for. The interface where the image of handwritten text will be uploaded, would also be updated to present various options to the user, relating to computation of the data, the presentation of the values (in an easily readable format), and the storage (the file format desired by the client).
4.	Social Impact / Customer Satisfaction	Removing the burden caused by the manual input of data would make the customer happier. This proposed solution will be useful for the faster processing of postal addresses, bank cheques, forms, handwritten ancient texts etc. The time required for uploading data becomes much shorter, and the less time consumed, the quicker the rest of the process proceeds.
5.	Business Model (Revenue Model)	This solution can be used by companies to process bulk datas such as phone numbers instantly. It simply automates the whole process of manual checking. And it is also a

3.4 PROBLEM - SOLUTION FIT:

CHAPTER 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	Input correlation	Image Correlation is a technique used to recognize characters from images.
FR-2	Data Preparation	Collecting data and prepare it for training
FR-3	Feature extraction	Feature extraction is analysing the images and derivesome characteristics from these images that identifyeach specific element
FR-4	Character classification	During the classification phase, the attributes of the data in the picture are compared to the classes in the database to determine which class the picture belongs to.

4.2 NON – FUNCTIONAL REQUIREMENTS:

Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The software is very easy to use and reduces the learning work.To recognize the digits from bank cheque,papers,numeric entry in forms etc.
NFR-2	**Security**	The handwritten digit recognition can be used by banking sector where it can be used to maintain thesecurity pin numbers, it can be also used for blind peoples by using sound output.
NFR-3	**Reliability**	This software will work reliably for low resolutionimages and not for graphical images.
NFR-4	**Performance**	Handwritten characters in the input image will be recognized with an accuracy of about 90% and more.
NFR-5	**Availability**	This system will retrieve the handwritten text regions only if the image contains written text in it.
NFR-6	**Scalability**	It contains thousands of handwritten digits that havebeen used in the development of programs .

CHAPTER 5

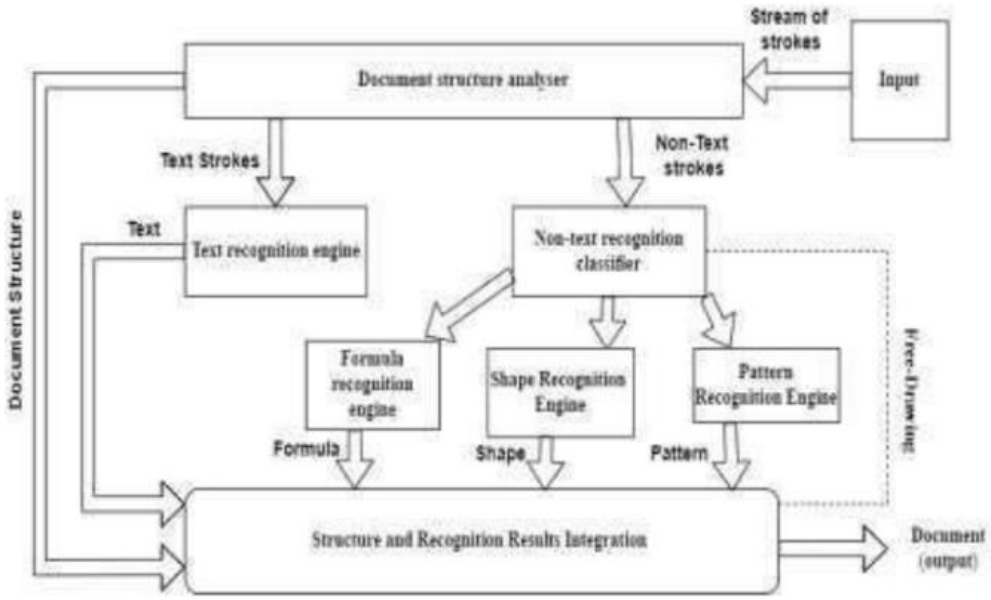
PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS:

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

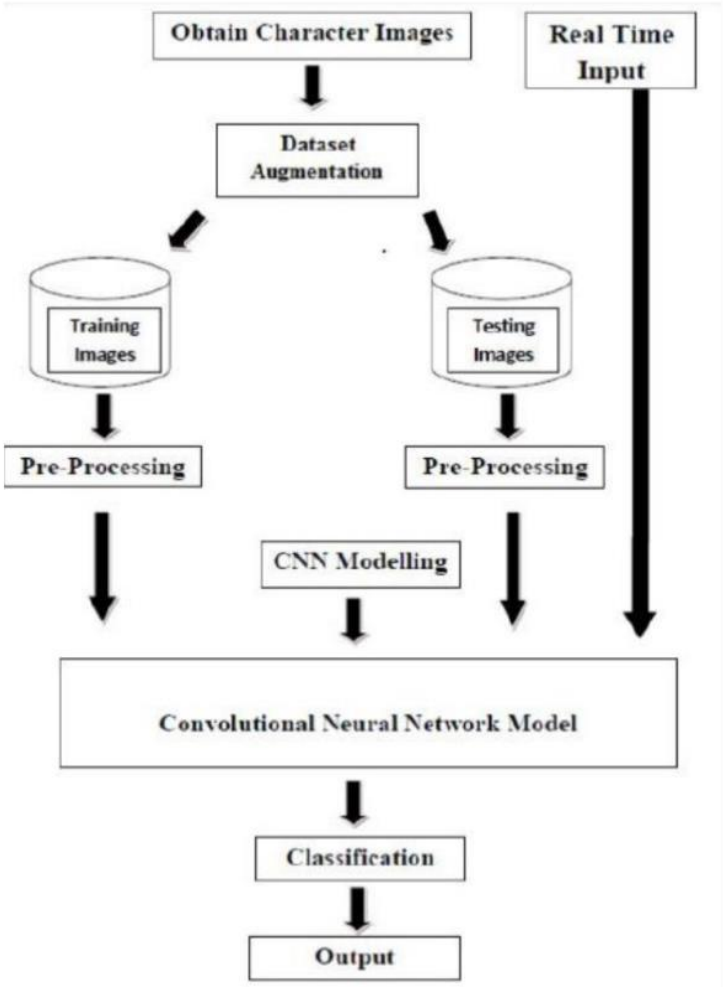
DFD Level-0:

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.

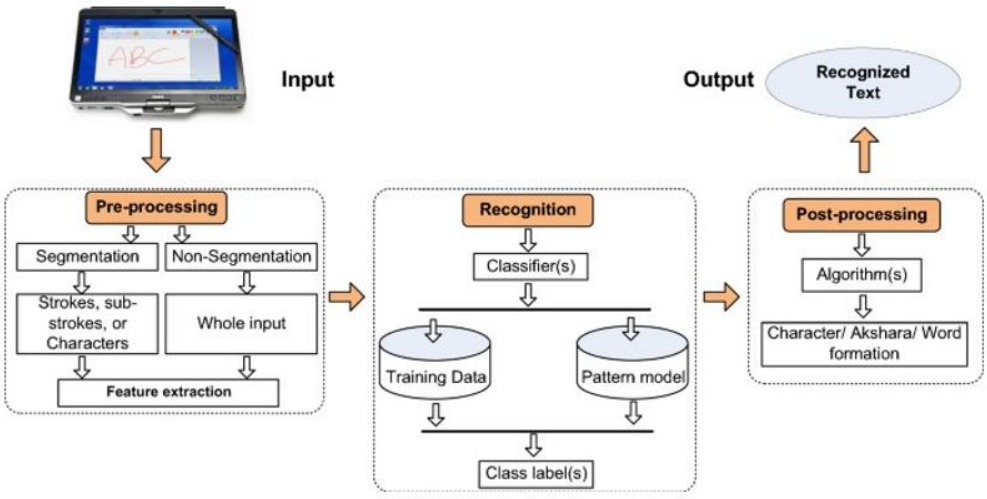


5.2 SOLUTION & TECHNICAL ARCHITECTURE:

SOLUTION ARCHITECTURE:



TECHNICAL ARCHITECTURE:



5.3 COMPONENTS & TECHNOLOGIES

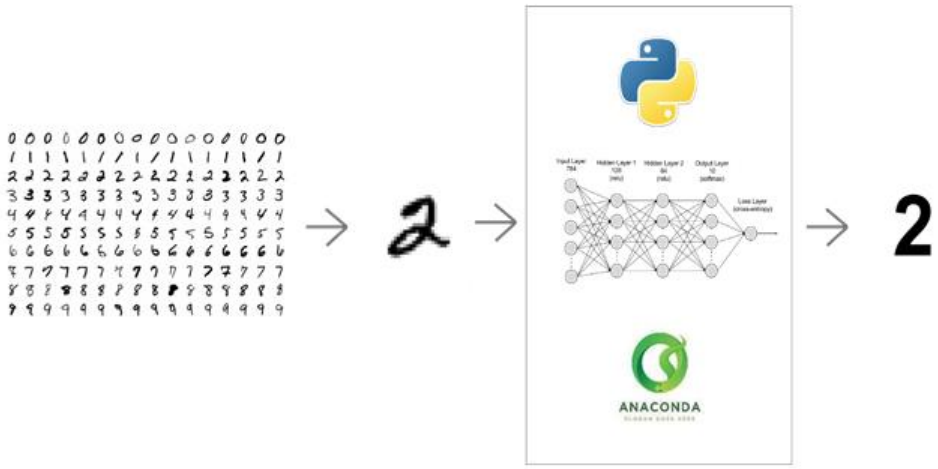
COMPONENTS:

S. No	Component	Description	Technology
1.	User Interface	How the user interacts with application e.g. Web UI, Mobile App, Chatbot etc.	HTML, CSS, JavaScript
2.	Application Logic-1	Logic for a process in the application	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 Cloud	IBM DB2, IBM Cloudant
7.	File Storage	File storage requirements	IBM Block Storage
8.	External API-1	Purpose of External API used in the application	IBM Weather API
9.	External API-2	Purpose of External API used in the application	Aadhar API
10.	Machine Learning Model	Purpose of Machine Learning Model	Object Recognition Model
11.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration Cloud Server Configuration	Local, Cloud Foundry

TECHNOLOGIES:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used	Technology of Open Source framework
2.	Security Implementations	List all the security / access controls implemented, use of firewalls etc.	SHA-256, Encryptions, IAM Controls, OWASP
3.	Scalable Architecture	Justify the scalability of architecture	3 – tier, Micro-services
4.	Availability	Abstract and Figures. The features for handwritten digit recognition have been introduced. These features are based on shape analysis of the digit image and extract slant or slope information. They are effective in obtaining good recognition accuracies	Distributed servers, IBM cloud
5.	Performance	The standard implementations of neural networks achieve an accuracy of ~ (98–99) percent in correctly classifying the handwritten digits.	Number of requests per sec, use of Cache, use of CDN's

MNIST DATASET PROCESSING WITH PYTHON:



5.4 USER STORIES:

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

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Home	USN-1	As a user, I can view the guide and awareness to use this application.	I can view the awareness to use this application and its limitations.	Low	Sprint-1
		USN-2	As a user, I'm allowed to view the guided video to use the interface of this application.	I can gain knowledge to use this application by a practical method.	Low	Sprint-1
		USN-3	As a user, I can read the instructions to use this application.	I can read instructions also to use it in a user-friendly method.	Low	Sprint-2
	Recognize	USN-10	As a user, I can use the web application virtually anywhere.	I can use the application portably anywhere.	High	Sprint-1
		USN-11	As it is an open source, can use it cost freely.	I can use it without any payment to be paid for it to access.	Medium	Sprint-2
		USN-12	As it is a web application, it is installation free	I can use it without the installation of the application or any software.	Medium	Sprint-4
	Predict	USN-13	As a user, I'm Allowed to upload and choose the image to be uploaded	I can upload and choose the image from the system storage and also in any virtual storage.	Medium	Sprint-3

CHAPTER 6

PLANNING AND SCHEDULING

6.1 SPRINT PLANNING AND EXECUTION:

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	Mrithula Baraniya Poonam Akshaya
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	Mrithula Baraniya Poonam Akshaya
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	Mrithula Baraniya Poonam Akshaya
Sprint-2	Add CNN layers	USN-4	Creating the model and adding the input, hidden, and output layers to it.	5	High	Mrithula Baraniya Poonam Akshaya
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	Mrithula Baraniya Poonam Akshaya
Sprint-2	Train & test the model	USN-6	As a user, let us train our model with our image dataset.	6	Medium	Akshaya Mrithula Baraniya Poonam
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	Mrithula Baraniya Poonam Akshaya

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
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	Mrithula Baraniya Poonam Akshaya
Sprint-3		USN-9	As a user, I can know the details of the fundamental usage of the application.	5	Low	Mrithula Baraniya Poonam Akshaya
Sprint-3		USN-10	As a user, I can see the predicted / recognized digits in the application.	5	Medium	Mrithula Baraniya Poonam Akshaya
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	Mrithula Baraniya Poonam Akshaya
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	Mrithula Baraniya Poonam Akshaya

6.2 SPRINT DELIVERY SCHEDULE:

SPRINT	TOTAL STORY POINTS	DURATION	SPRINT START DATE	SPRINT END DATE (PLANNED)	STORY POINTS COMPLETED (AS ON PLANNED DATE)	SPRINT RELEASE DATE (ACTUAL)
Sprint - I	11	6 Days	24 Oct 2022	29 Oct 2022	11	29 Oct 2022
Sprint - II	9	6 Days	31 Oct 2022	05 Nov 2022	9	05 Nov 2022
Sprint - III	10	6 Days	07 Oct 2022	12 Nov 2022	10	12 Nov 2022
Sprint - IV	9	6 Days	14 Nov 2022	19 Nov 2022	9	19 Nov 2022

6.3 REPORT FROM JIRA:

Velocity:

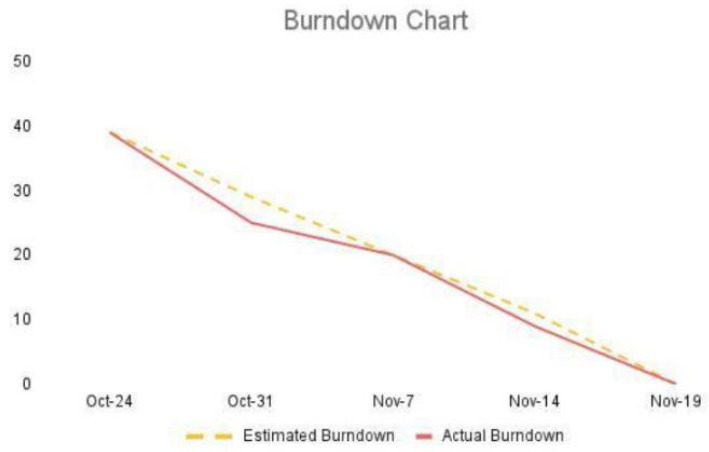
Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint).Let's calculate the team's average velocity (AV) per iteration unit (story points per day):

$$Team\ Velocity = \frac{\sum sprint-I + sprint-II + \ldots}{total\ sprints} = \frac{11+9+10+9}{4} = 9.75$$

$$AV = \frac{team\ velocity}{duration} = \frac{9.75}{6} = 1.625$$

Burndown Chart:

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



CHAPTER 7

CODING AND 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 file from request.files[]object and saves it to the upload folder.

```
import 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-
recognitionsystem/flask_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.

```
<section 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>
```

```
function preview() {
    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 script tag.

```
<script>

$(document).ready(function() {
$('#clear button').on('click', function() {
            $('#image').val('');
            $('#frame').attr('src',"");
        });
    });
```

CHAPTER 8

TESTING

8.1 TEST CASES

Test case ID	Feature Type	Component	Test Scenario	Expected Result	Actual Result	Status
HP_TC_001	UI	Home Page	Verify UI elements or 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 2560x1801 And 768x630	FAIL
HP_TC_003	Functional	Home Page	Check if the 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 the user cannot upload unsupported files	The application should not allow the 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

8.2.1 DEFECT ANALYSIS:

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	1	0	1	0	2
Duplicate	0	0	0	0	0
External	1	0	0	1	2
Fixed	2	0	1	1	4
Not Reproduced	0	0	0	0	0
Skipped	0	0	0	0	0
Won't Fix	0	0	0	0	0
Totals	4	0	2	2	8

8.2.2 TEST CASE ANALYSIS:

Section	Total Cases	Not Tested	Fail	Pass
Home Page	5	0	0	5
Upload Option	7	0	0	7
Image File Upload	5	0	0	5
Redirection to Prediction Page	5	0	0	5
Correct Prediction of Number	10	0	0	10

CHAPTER 9

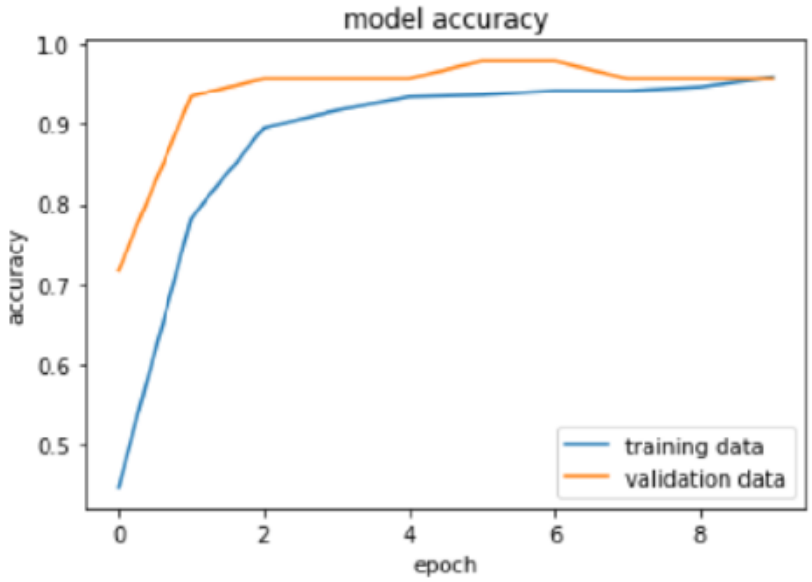
RESULTS

9.1 PERFORMANCE METRICS:

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
_________________________________________________________________
```

2. **Accuracy**



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
- 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 might try with clear image and neat handwriting to get accuracy in digits
- Unclear image will not give accurate results

CHAPTER 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 the 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.

CHAPTER 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 in the roadside, user shall give only the destination.

All of these enhancement 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. Unless of those the most of the visualizing platforms like on which developers are doing some works 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.

APPENDIX

SOURCE CODE

MODEL CREATION:

Import the necessary packages

In [17]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from keras.utils import np_utils
from tensorflow.keras.datasets import mnist
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, Dense, Flatten
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.models import load_model
from PIL import Image, ImageOps
```

Load data

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

Data Analysis

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

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

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

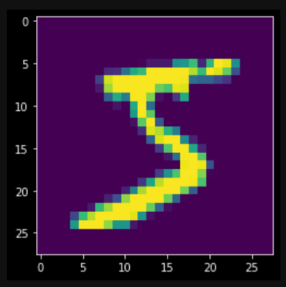
Out[4]:
```
array([[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,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

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

Out[5]: 5

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

Out[6]:



Data Pre-Processing

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

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

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

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

Create model

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

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

Train the model

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

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

Out[12]:

Test the model

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

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

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

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

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

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

Save the model

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

Test the saved model

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

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

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

SCRIPT (JS) – script.js:

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

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

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

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

HOME PAGE (HTML) – home.html:

```
<html>
    <head>
        <meta name="viewport" content="width=device-width, initial-scale=1.0" />
        <title>Handwritten Digit Recognition System</title>
        <link rel="stylesheet" href="{{url_for('static',filename='css/main.css')}}" />
        <script src="https://unpkg.com/feather-icons"></script>
        <script defer src="{{url_for('static',filename='js/script.js')}}"></script>
    </head>
    <body>
        <div class="container">
            <div class="heading">
                <h1 class="heading__main">Handwritten Digit Recognition System</h1><br>
                <h2 class="heading__sub">A Novel Method for Handwritten Digit Recognition System</h2><br><br>
            </div>
            <div>
                <div class="form-wrapper">
                    <form class="upload" action="/predict" method="post" enctype="multipart/form-data">
                        <label id="label" for="upload-image">Upload Image</label>
                        <input type="file" name="photo" id="upload-image" hidden />
                        <button type="submit" id="up_btn"></button>
                    </form>
                    <img id="loading" src="{{url_for('static',filename='images/loading.gif')}}">
                </div>
            </div>
        </div>
    </body>
</html>
```

HOME PAGE (CSS) – style.css:

```
* {
        padding: 0;
        margin: 0;
}

body {
        color: black;
        font-family: "Overpass", sans-serif;
        background-color: rgb(188, 236, 250);
}

.container {
        width: 100%;
        height: 100%;
        display: flex;
        flex-direction: column;
        justify-content: center;
        align-items: center;
        background-color: rgb(188, 236, 250);
}

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

```

```css
.heading .heading__main {
        color: rgb(0, 43, 88);
        font-size: 3rem;
        font-weight: 550;
}

.heading .heading__sub {
        font-size: 1rem;
        color: rgb(97, 97, 97);
}

.upload-container {
        box-shadow: 0 0 20px rgba(111, 183, 35, 0);
        width: 30rem;
        height: 15rem;
        padding: 1.5rem;
}

.form-wrapper {
        background-color: rgba(255, 0, 0, 0);
        width: 100%;
        height: 100%;
        display: flex;
        justify-content: center;
        align-items: center;
}

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

.form-wrapper .upload {
        display: flex;
        justify-content: center;
        align-items: center;
        width: 15rem;
        text-align: center;
        height: -webkit-fit-content;
        height: -moz-fit-content;
        height: fit-content;
        border-radius: 6px;
        color: white;
        background-color: rgb(36, 113, 255);
        box-shadow: 0 5px 10px rgb(1, 75, 153);
}

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

.form-wrapper .upload label {
        font-size: 1rem;
        font-weight: 600;
        color: white;
        height: 100%;
        width: 100%;
        padding: 15px;
        display: block;
}

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

@media screen and (max-width: 700px) {
        .upload-container {
                height: 20rem;
                width: 18rem;
                margin-top: 3.5rem;
                margin-bottom: -8rem;
        }

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

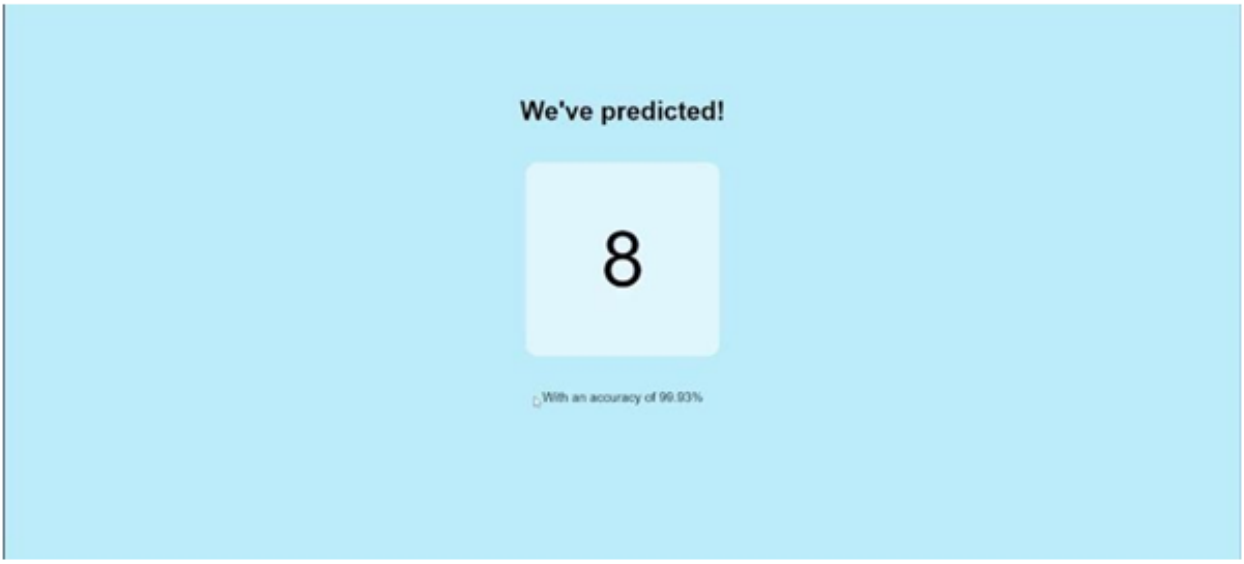
PREDICTPAGE (HTML) – predict.html:

```
<html>
        <head>
        <meta name="viewport" content="width=device-width, initial-scale=1.0" />
                <title>Handwritten Digit Recognition System</title>
                <link rel="stylesheet" href="{{url_for('static',filename='css/predict.css')}}" />
                <meta name="viewport" content="width=device-width, initial-scale=1.0" />
        </head>
        <body>
                <div class="container"><br><br><br>
                        <h1>We've predicted!</h1>
                        <div class="result-wrapper">
                                <div class="result-container">
                                        <div class="value">{{best.0}}</div>
                                </div>
                        </div>
                        <div class="accuracy"><br>With an accuracy of {{best.1}}%<br></div>
                </div>
        </body>
</html>
```

FLASK APP – flask.py:

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

app=Flask(__name__)

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


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


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

SCREENSHOTS:



GITHUB LINK:

https://github.com/IBM-EPBL/IBM-Project-25783-1659973203

Team Id : PNT2022TMID53487

Project Name : A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

TEAM MEMBERS:

Akshaya K B	190801004
Baraniya S	190801008
Mrithula Angeline M J	190801046
Poonam B N	190801053