PROJECT REPORT

A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

Submitted By

PNT2022TMID24591

KRITHIKA B - 210419104033
SUBIKSHA U B - 210419104169
SADHANA L - 210419104138
SHIVAKAR V - 210419104157

TABLE OF CONTENTS

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.

1.2 Purpose

Digit Recognition system are capable of recognizing the digits from different sources like emails, bank cheques, papers, images, etc. and in different real-world scenarios for online handwriting recognition on computer, tablets or systems, recognize number plates of vehicles, processing bank cheque amounts, numeric enteries in forms filled up by hand (tax forms) and so on.

2. LITERATURE SURVEY

2.1 Existing problem

The different architectures of CNN, hybrid CNN,CNN - RNN and CNNHMM models, and domain - specific recognition system, are not thoroughly inquired and evolutionary algorithms are not clearly explored for optimizing CNN learning parameters ,the number of layers, learning rate and kernel sizes of convolutional filters.

2.2 References

[1] Kai Ding, Zhibin Liu, Lianwen Jin, Xinghua Zhu, “A Comparative study of GABOR feature and gradient feature for handwritten chinese character recognition”, International Conference on Wavelet Analysis and Pattern Recognition, pp. 1182- 1186, Beijing, China, 2-4 Nov. 2007.

[2] Pranob K Charles, V.Harish, M.Swathi, CH. Deepthi, "A Review on the Various Techniques used for Optical Character Recognition", International Journal of Engineering Research and Applications, Vol. 2,Issue 1, pp. 659-662, Jan-Feb 2012.

[3] Bhatia Neetu, “Optical Character Recognition Techniques”, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 4, Issue 5, May 2014.

[4] Liana M. Lorigo and Venu Govindaraju, “Offline Arabic Handwriting Recognition: A Survey”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 28 Issue 5, May 2006.

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 analyse images easily. Also, recognize the different elements present in the images.

- the goal is to correctly identify digits from a dataset of tens of thousands of handwritten images and experiment with different algorithms to learn first-hand what works well and how techniques compare

- 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 .

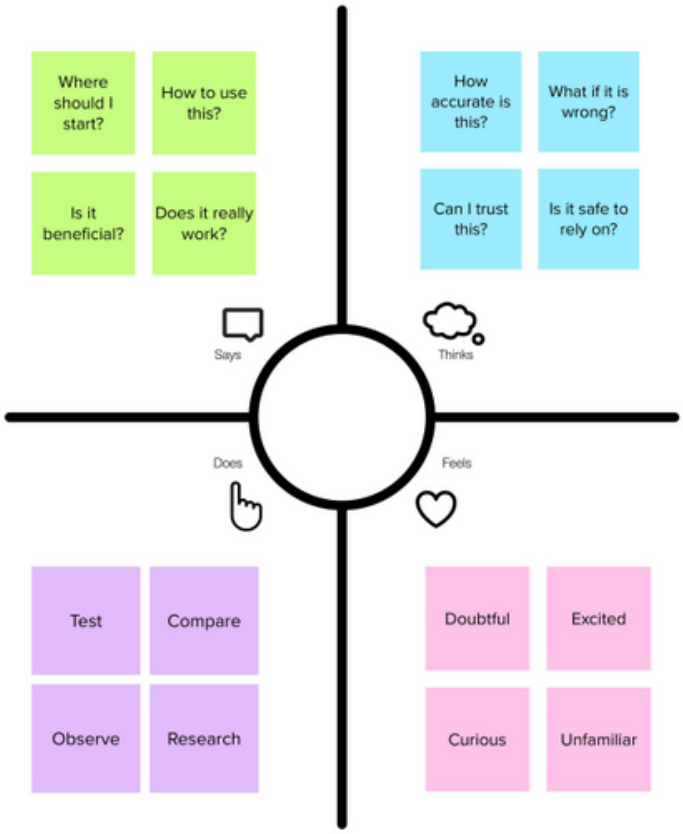
- Handwriting number recognition is a challenging problem researchers had been research into this area for so long especially in the recent years

QUESTION	DESCRIPTION
What does the problem affect?	Handwriting recognition tends to have problems when it comes to accuracy. People can struggle to read others' handwriting. How, then, is a computer going to do it? The issue is that there's a wide range of handwriting – good and bad. This makes it tricky for programmers to provide enough examples of how every character might look.
What are the boundaries of the problem?	As the manually written digits aren't of a comparable size, thickness, position and direction, numerous difficulties need to be taken into consideration to decide the problem of handwritten digit recognition and it also involves the difficulty of visual pattern recognition.
What is the issue?	The handwritten digits are not always of the same size, width, orientation and justified to margins as they differ from writing of person to person, so the general problem would be while classifying the digits due to the similarity between digits such as 1 and 7, 5 and 6, 3 and 8, 2 and 5, 2 and 7, etc.
When does the issue occur?	Perhaps the most obvious problem when processing handwritten forms during the data capture process is poor

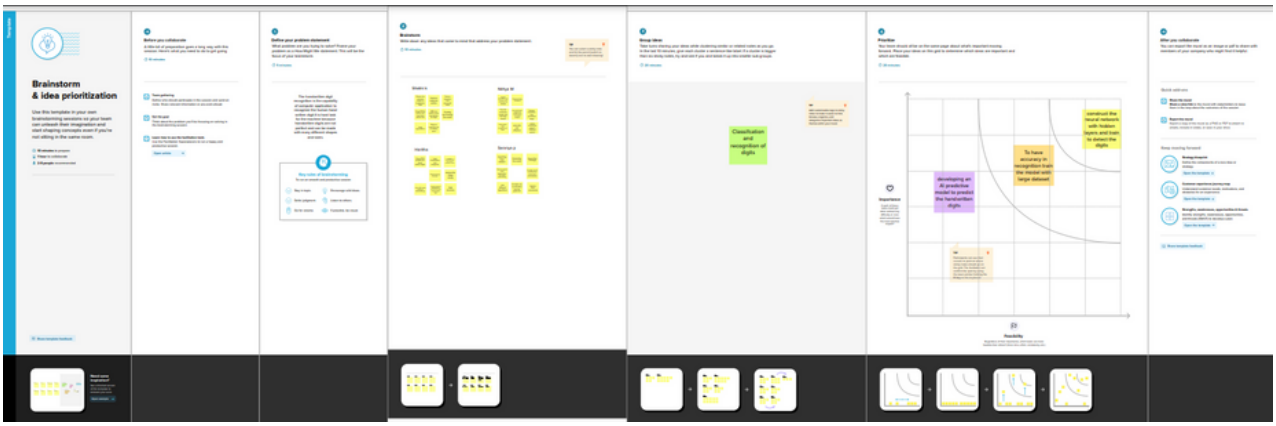
	quality or illegible handwriting. We all know the old stereotype about doctors' handwriting, so trying to perform accurate data capture and validation on this type of form-filling may result in little meaningful data being extracted.
Where is the issue occurring?	During the data capture validation stages of any forms processing activity,all required text fields are processed which involves recognition and extracting the written characters.
Why is it important that we fix the problem?	The high variance in handwriting styles across people and poor quality of the handwritten text compared to printed text pose significant hurdles in converting it to machine readable text. Nevertheless it's a crucial problem to solve for multiple industries like healthcare, insurance and banking.

3. IDEATION & PROPOSED SOLUTION

3.1Empathy Map Canvas



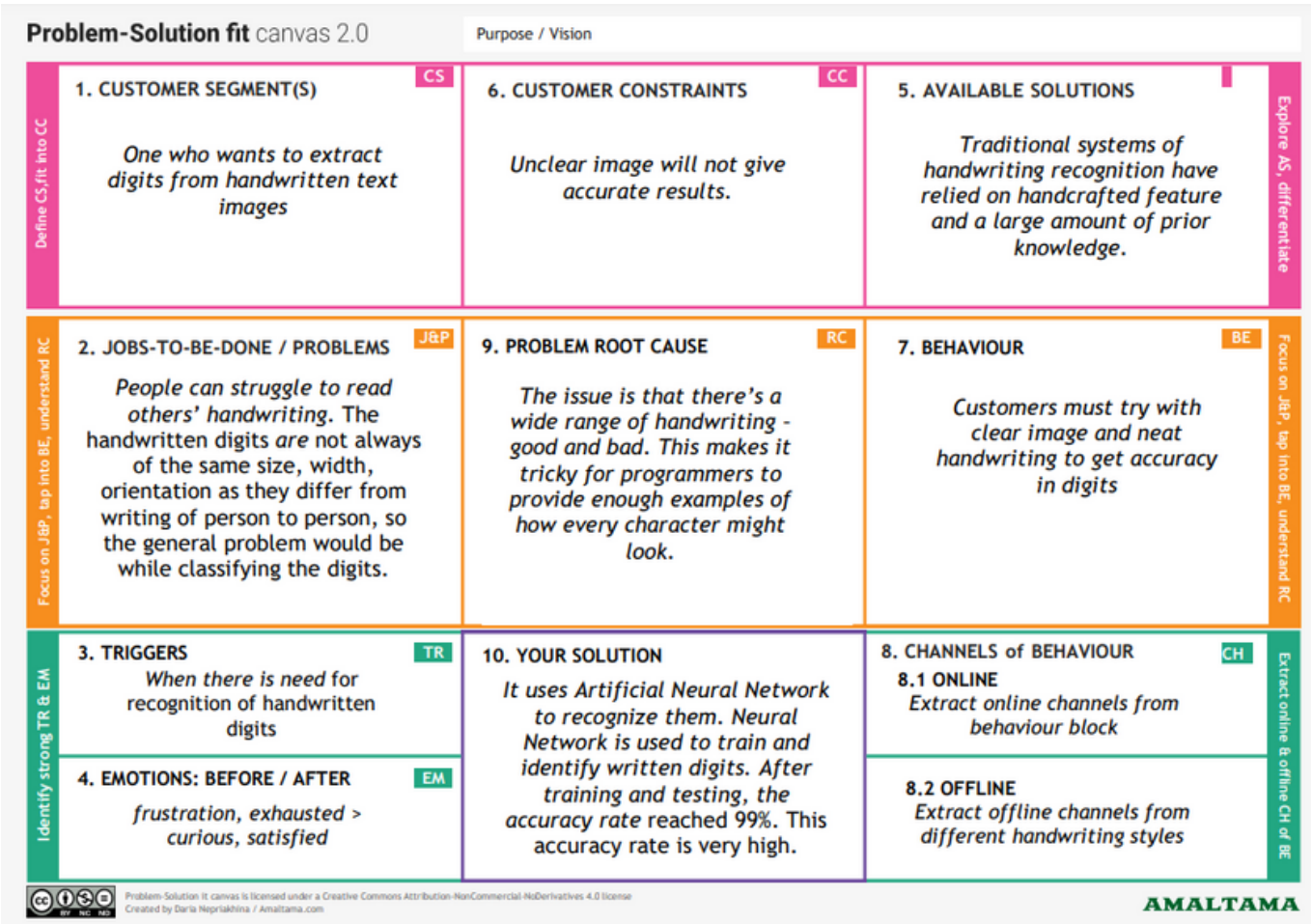
3.2 Ideathon & Brainstorming



3.3 Proposed Solution

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

3.4 Problem Solution Fit



4. REQUIREMENT ANALYSIS

4.1 Functional requirement

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

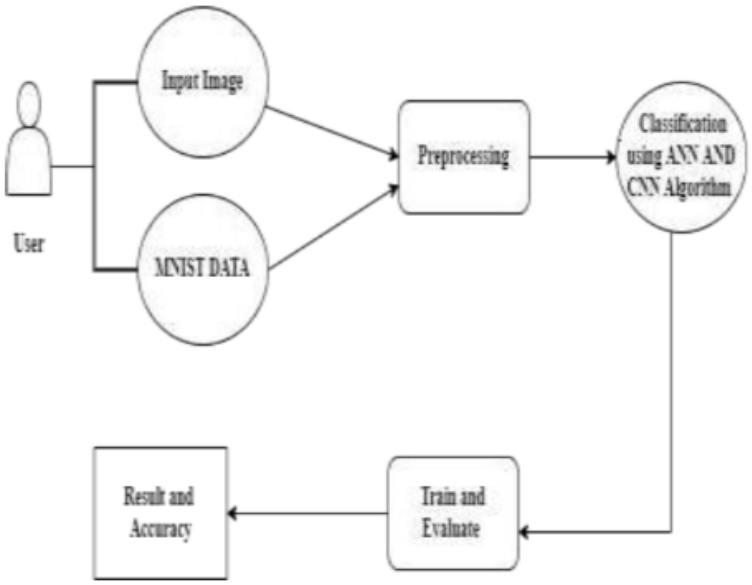
4.1 Non-Functional requirement

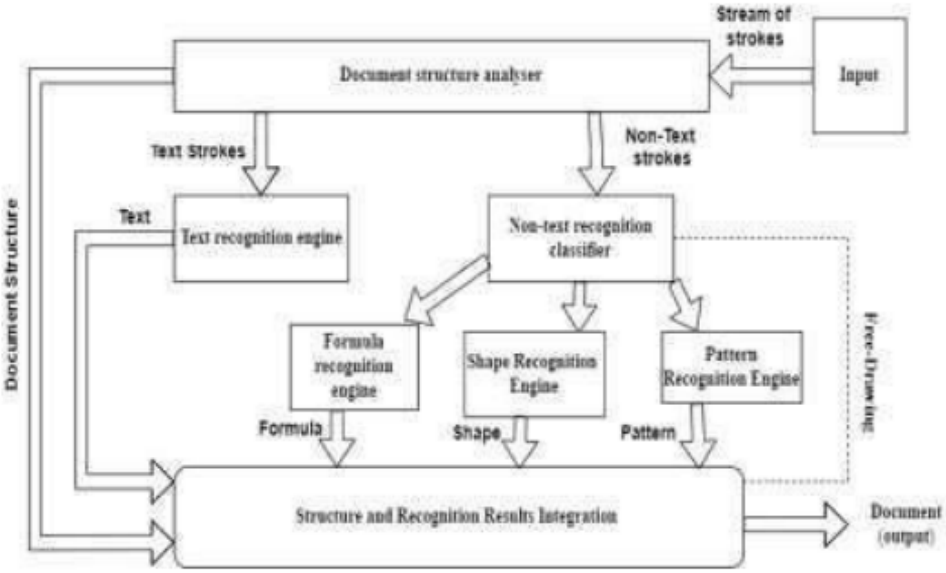
FR No	Non-Functional Requirement	Description
NFR-1	**Usability**	One of the very significant problems in pattern recognition applications is the recognition of handwritten characters. Applications for digit recognition include filling out forms, processing bank checks, and sorting mail.
NFR-2	**Security**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

5. PROJECT DESIGN

5.1 Data Flow Diagrams

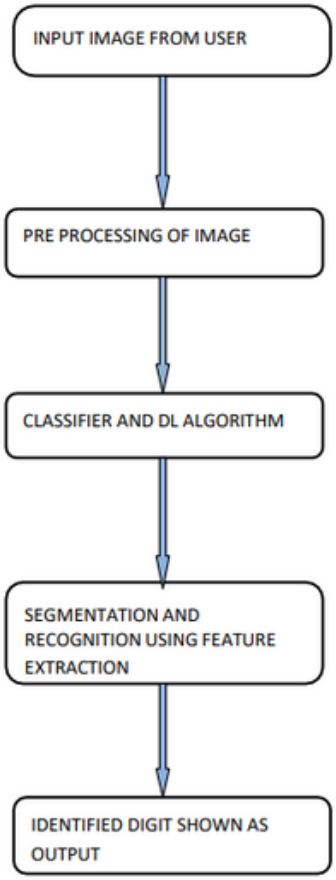
Data Flow Diagrams: A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where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





5.2 Solution & Technical Architecture

Solution Architecture



Dataset

The MNIST data collection, which contains 70000 handwritten digits, is frequently utilized for this recognition method. In order to train these photos and create a deep learning model, we use artificial neural networks. A web application is developed that allows users to upload pictures of handwritten numbers. The model examines this picture. The 60,000 training and 10,000 testing labeled handwritten digit images in the MNIST Handwritten Digit Recognition Dataset.

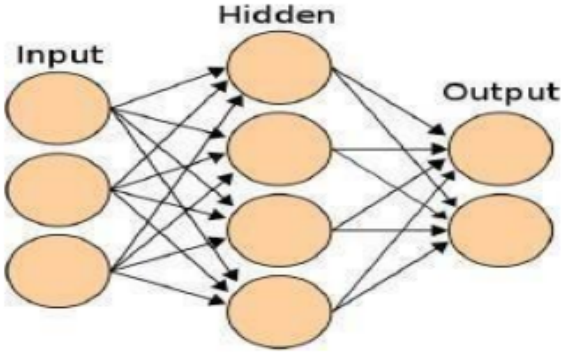
APPROACH:

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

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

- The hidden layer: The network's nonlinear ties are provided by hidden units termed activations that make up the hidden layer. Depending on our needs, there can be a variety of concealed layers.

- The output layer: The nodes in this stratum are referred to as output units. It gives us access to the neural network's final prediction, which may be used to make final predictions.



METHODOLOGY:

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

WORKING:

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

CONVOLUTION LAYER :

The foundational component of a CNN is the convolutional layer. The parameters of the layer are a set of learnable filters (or kernels) that cover the entire depth of the input volume but have a narrow receptive field. Each filter is convolved across the width and height of the input volume during the forward pass, computing the dot product between each filter entry and the input to create a two-dimensional activation map of the filter. As a result, the network picks up filters that turn on when it detects a certain kind of feature at a particular spatial location in the input.

FEATURE EXTRACTION :

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

TENSORFLOW :

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

Technology Architecture

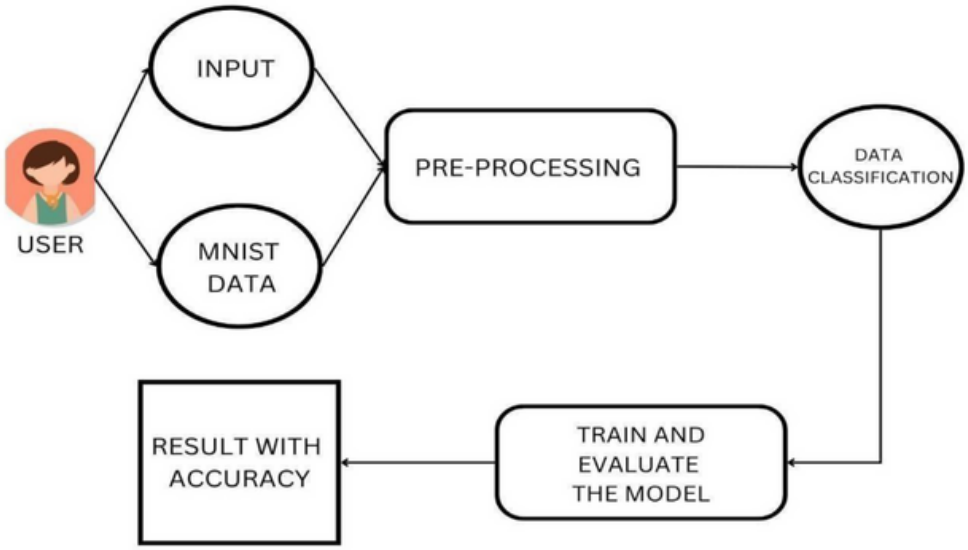


Table -1: Components & Technologies

S.NO	Component	Description	Technology
1.	User Interface	Allows the user to enter the input and recognise the input using GUI.	HTML,CSS, JavaScript
2.	Digit Prediction	Here the digit given as a input is predicted.	Keras,CNN.
3.	Representation	Skeleton, counters, pixels or others	Java / Python
4.	Segmentation	Task of clustering parts of an image together that belong to the same object class.	Convolutional neural networks & super pixels.
5.	Machine Learning Model	Purpose of Machine Learning Model is to train and test the data and predict the user input.	Classification.
6.	Infrastructure	Application deployment on local system Local server Configuration: Intel core i5/i3 10th Generation.	HTML, CSS
7.	Neural network	Automatically infer rules for recognizing handwritten digits	Convolutional neural network

Table – 2: Application Characteristics:

S.NO	Characteristics	Description	Technology
1.	Pre-processing	Data pre-processing is a process of preparing the raw data and making it suitable for a machine learning model.	Real time online handwritten character recognition system, based on an ensemble of neural networks.
2.	Open-Source Frameworks	Enables developers to develop complex code and web application quickly.	Open source-Jupyter, anaconda navigator, flask framework.
3.	Dataset	It Contains 60,000 training images.	MNIST
4.	Security Implementations	After predicting the data, we don't store any data so we can't manipulate it in future.	Encryption
5.	Performance	Neural networks achieve an accuracy of 98-99 percent in correctly classifying the handwritten digits.	Convolutional Neural Networks.

5.3 User Stories

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

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

6. PROJECT PLANNING & SCHEDULING

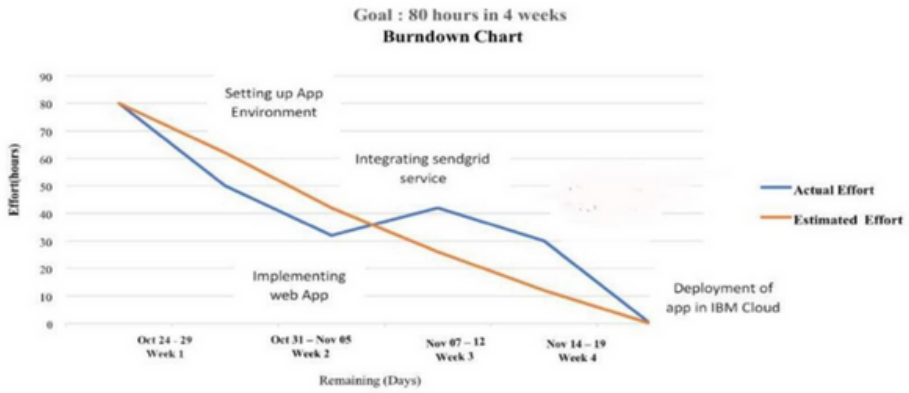
6.1 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority
Sprint-1	Dashboard	USN-1	As a user, they can see the information regarding the prediction of handwritten digit recognition.	2	High
Sprint-1	Launch	USN-2	On clicking the launch button, it will redirect the user to a page where the images to be predicted can be uploaded.	2	High
Sprint-2	Upload	USN-3	Users can select the image from the local storage.	2	High
Sprint-3	Predict	USN-4	Once the image is uploaded, it will predict the respective image.	2	High
Sprint-4	Display	USN-5	The predicted image will be displayed with the accuracy chart.	2	High

6.2 Sprint Delivery Schedule

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

6.3 REPORTS FROM JIRA



7. CODING & SOLUTIONING

```
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 = 'D:/ibm/data'


app = Flask(__name__)
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER

model = load_model("./DigitRecog_IBM_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)
```

8. TESTING

8.1Test Cases

Test case ID	Feature Type	Component	Test Scenario	Expected Result	Actual Result	Status
Homepage_TC_OO1	Functional	Home Page	Verify user is able to see the Homepage when clicked on the link	Home Page should be displayed.	Working as expected	Pass
Homepage_TC_OO2	UI	Home Page	Verify the UI elements in Homepage	Application should show below UI elements: a.choose file button b.predict button c.clear button	Working as expected	Pass
Homepage_TC_OO3	Functional	Home Page	Verify user is able to choose file from the local system and click on predict	Choose file popup screen must be displayed and user should be able to click on predict button	Working as expected	Pass
Homepage_TC_OO4	Functional	Home page	Verify user able to select invalid file format	Application won't allow to attach formats other than ".png, .jiff, .pjp, .jpeg, .jpg, .pjpeg"	Working as expected	Pass
Predict_TC_OO5	Functional	Predict page	Verify user is able to navigate to the predict to and view the predicted result	User must be navigated to the predict page and must view the predicted result	Working as expected	Pass

8.2 User Acceptance Testing

Defect Analysis

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

Test Case Analysis

Section	Total Cases	Not Tested	Fail	Pass
Client Application	5	0	0	5
Security	5	0	0	5
Final Report Output	5	0	0	5
Performance	5	0	0	5

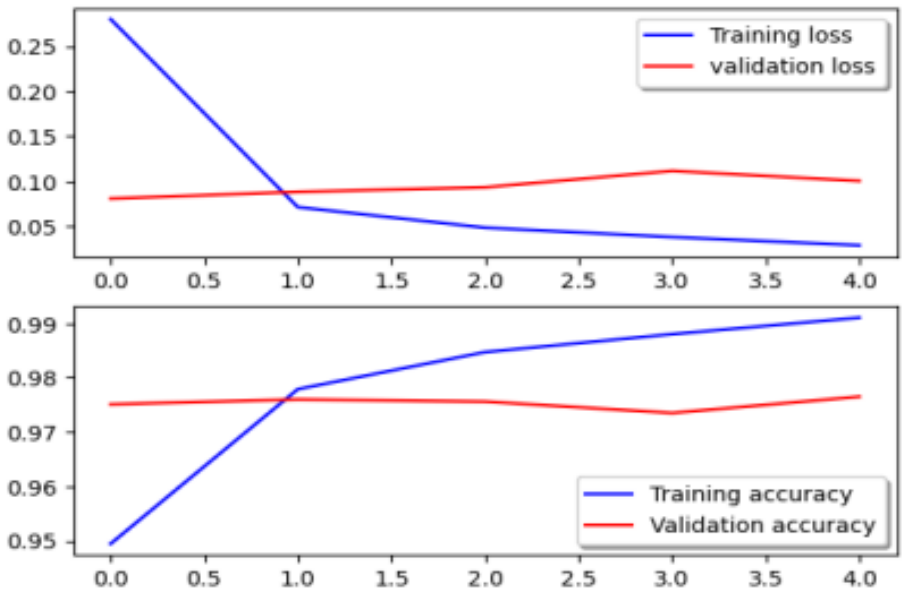
9. RESULTS

9.1Performance Metrics

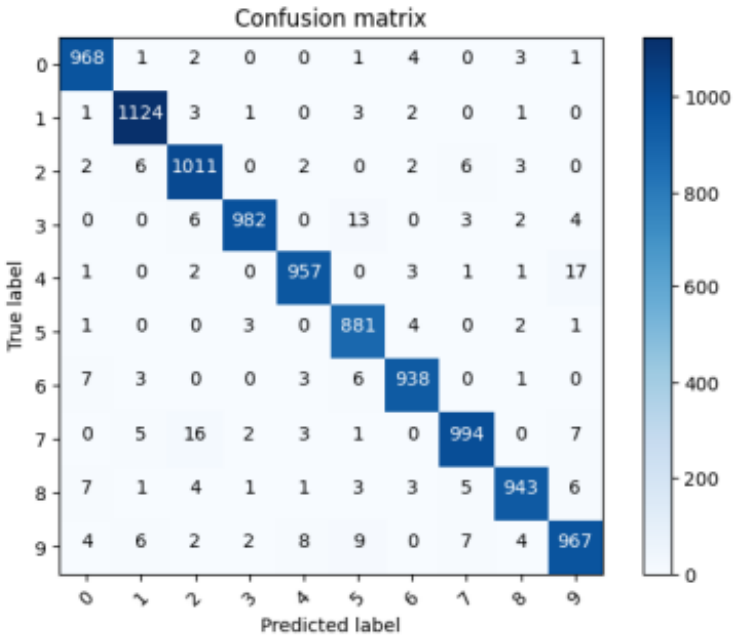
Model Summary:

```
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
_________________________________________________________________
None
```

Accuracy:



Confusion Matrix:



Classification Report:

	precision	recall	f1-score	support
0	0.98	0.99	0.98	980
1	0.98	0.99	0.99	1135
2	0.97	0.98	0.97	1032
3	0.99	0.97	0.98	1010
4	0.98	0.97	0.98	982
5	0.96	0.99	0.97	892
6	0.98	0.98	0.98	958
7	0.98	0.97	0.97	1028
8	0.98	0.97	0.98	974
9	0.96	0.96	0.96	1009
accuracy			0.98	10000
macro avg	0.98	0.98	0.98	10000
weighted avg	0.98	0.98	0.98	10000

10.ADVANTAGES & DISADVANTAGES

Advantages

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

Disadvantages

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

11.CONCLUSION

This project demonstrated a web application that uses machine learning to recognie 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 realworld scenarios such as recognizing number plates of vehicles, processing bank cheque
amounts, numeric entries in forms filled up by hand (tax forms) and so on. There is so much
room for improvement, which can be implemented in subsequent versions.

12.FUTURE SCOPE

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

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

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