IBM PROJECT

A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

DOMAIN - ARTIFICIAL INTELLIGENCE

Submitted by:

Kaviyarasu B - 510419104039
Abinesh R G - 510419104002
Chandru C - 510419104019
Meiyazhagan V - 510419104046

Table of Contents

1. INTRODUCTION

1.1 Project Overview

- 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.
- 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.

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.

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. 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

Paper 1: Handwritten digit recognition by combined classifiers
Author : M. Breukelen; Robert P. W. Duin; David M. J. Tax; J. E. den Hartog

Summary: Classifiers can be combined to reduce classification errors. We did experiments on a data set consisting of different sets of features of handwritten digits. Different types of classifiers were trained on these feature sets.

The performances of these classifiers and combination rules were tested. The best results were acquired with the mean, median and product combination rules.

The product was best for combining linear classifiers, the median for k-NN classifiers. Training a classifier on all features did not result in less errors

.

Paper 2 : **A trainable feature extractor for handwritten digit recognition**
Author: Fabien Lauer, Ching Y. Suen, Gérard Bloch

Summary: This article focuses on the problems of feature extraction and the recognition of handwritten digits. A trainable feature extractor based on the LeNet5 convolutional neural network architecture is introduced to solve the first problem in a black box scheme without prior knowledge on the data. In order to increase the recognition rate, new training samples are generated by affine transformations and elastic distortions. Experiments are performed on the well-known MNIST database to validate the method and the results show that the system can outperform both SVMs and LeNet5 while providing performances comparable to the best performance on this database. Moreover, an analysis of the errors is conducted to discuss possible means of enhancement and their limitations.

Paper 3 : **Handwritten digit recognition by neural networks with singlelayer training**
Author : S. Knerr; L. Personnaz; G. Dreyfus

Summary: It is shown that neural network classifiers with single-layer training can be applied efficiently to complex real-world classification problems such as the recognition of handwritten digits. The STEPNET procedure, which decomposes the problem into simpler subproblems which can be solved by linear separators, is introduced. Provided appropriate data representations and learning rules are used, performance comparable to that obtained by more complex networks can be achieved. Results from two different databases are presented: an European database comprising 8700 isolated digits and a zip code database from the US Postal Service comprising 9000 segmented digits. A hardware implementation of the classifier is briefly described

2.3 Problem Statement Definition

For years, the traffic department has been combating traffic law violators. These offenders endanger not only their own lives, but also the lives of other individuals.

Punishing these offenders is critical to ensuring that others do not become like them. Identification of these offenders is next to impossible because it is impossible for the average individual to write down the license plate of a reckless driver.

Therefore, the goal of this project is to help the traffic department identify these offenders and reduce traffic violations as a result.

3. IDEATION & PROPOSED SOLUTION

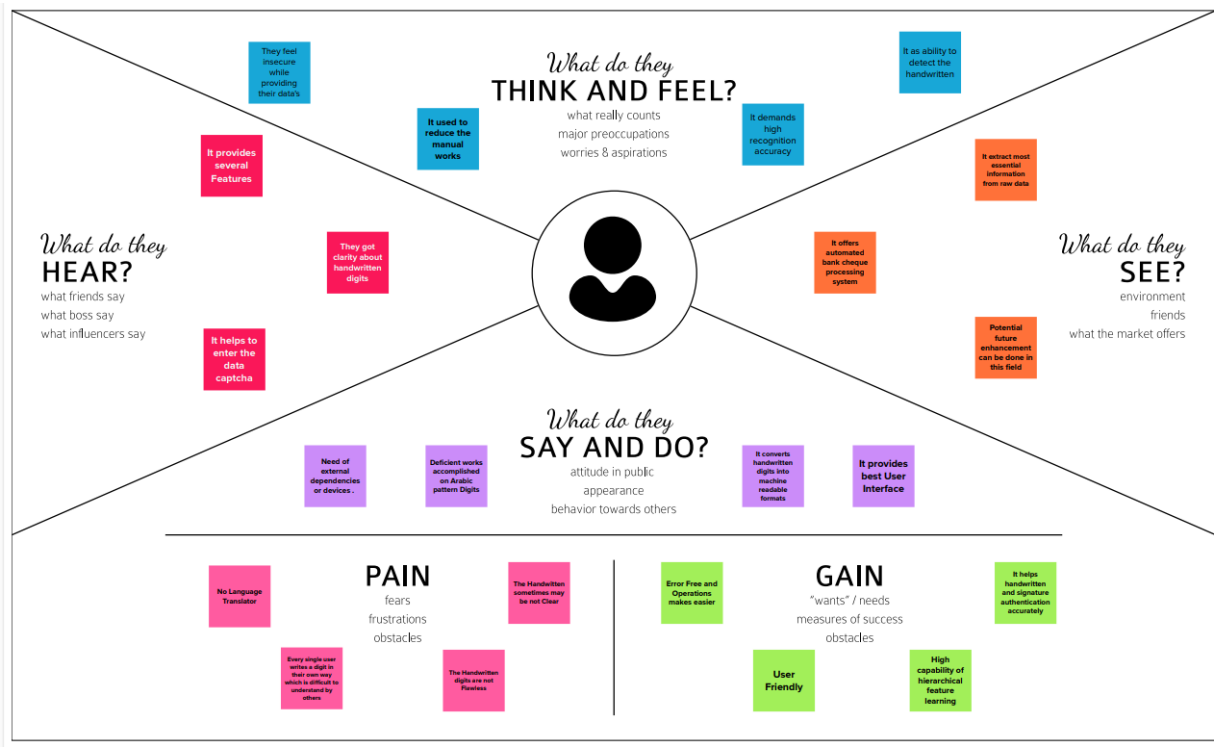
3.1 Empathy Map Canvas

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviors and attitudes. It is a useful tool to helps teams better understand their users.

Creating an effecting solution requires understanding the true problem and the person who is experiencing it.

The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.

Example:



3.2 Ideation & Brainstorming

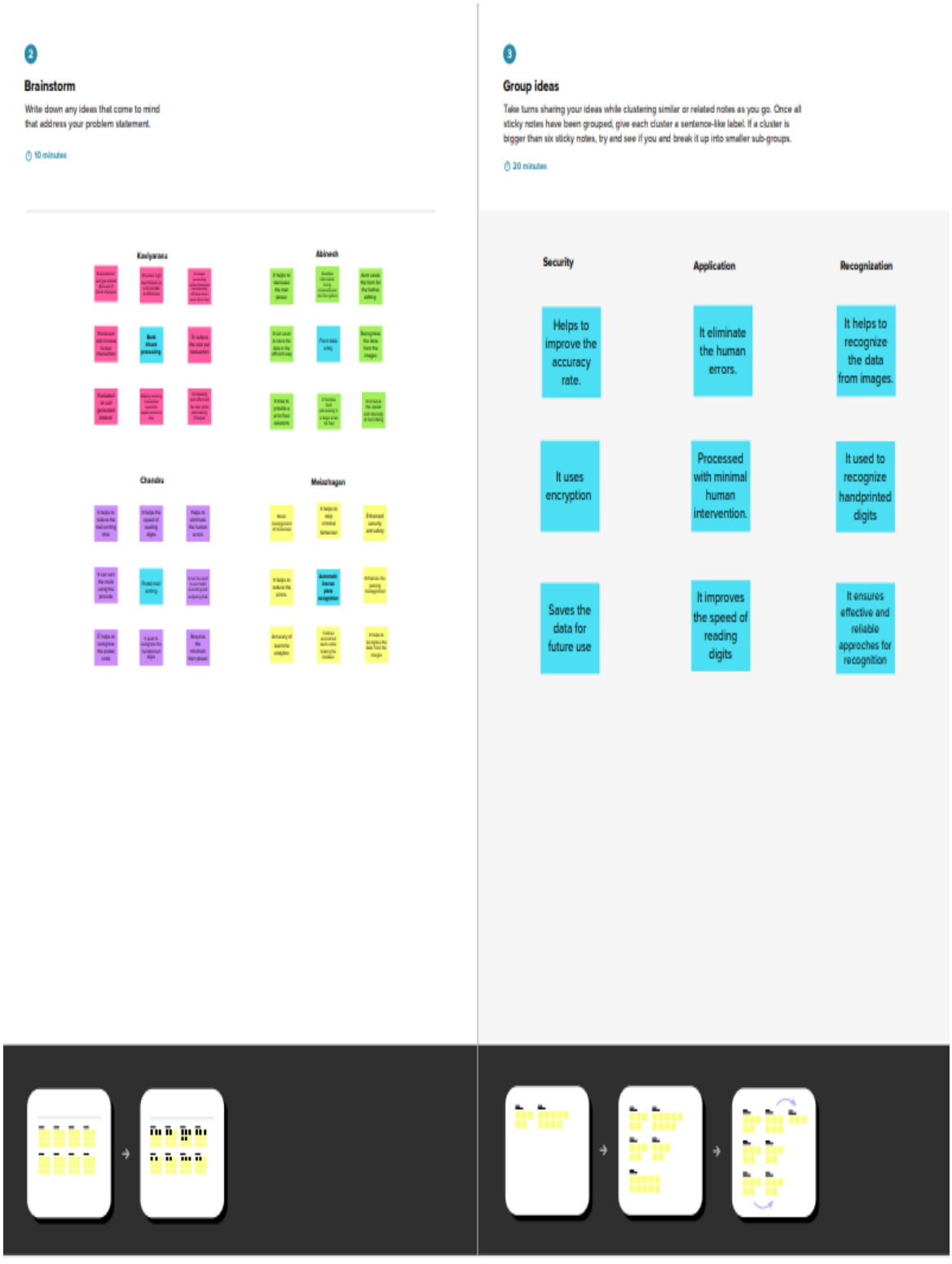
Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving.

Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.

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.

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

Step-2: Brainstorm, Idea Listing and Grouping



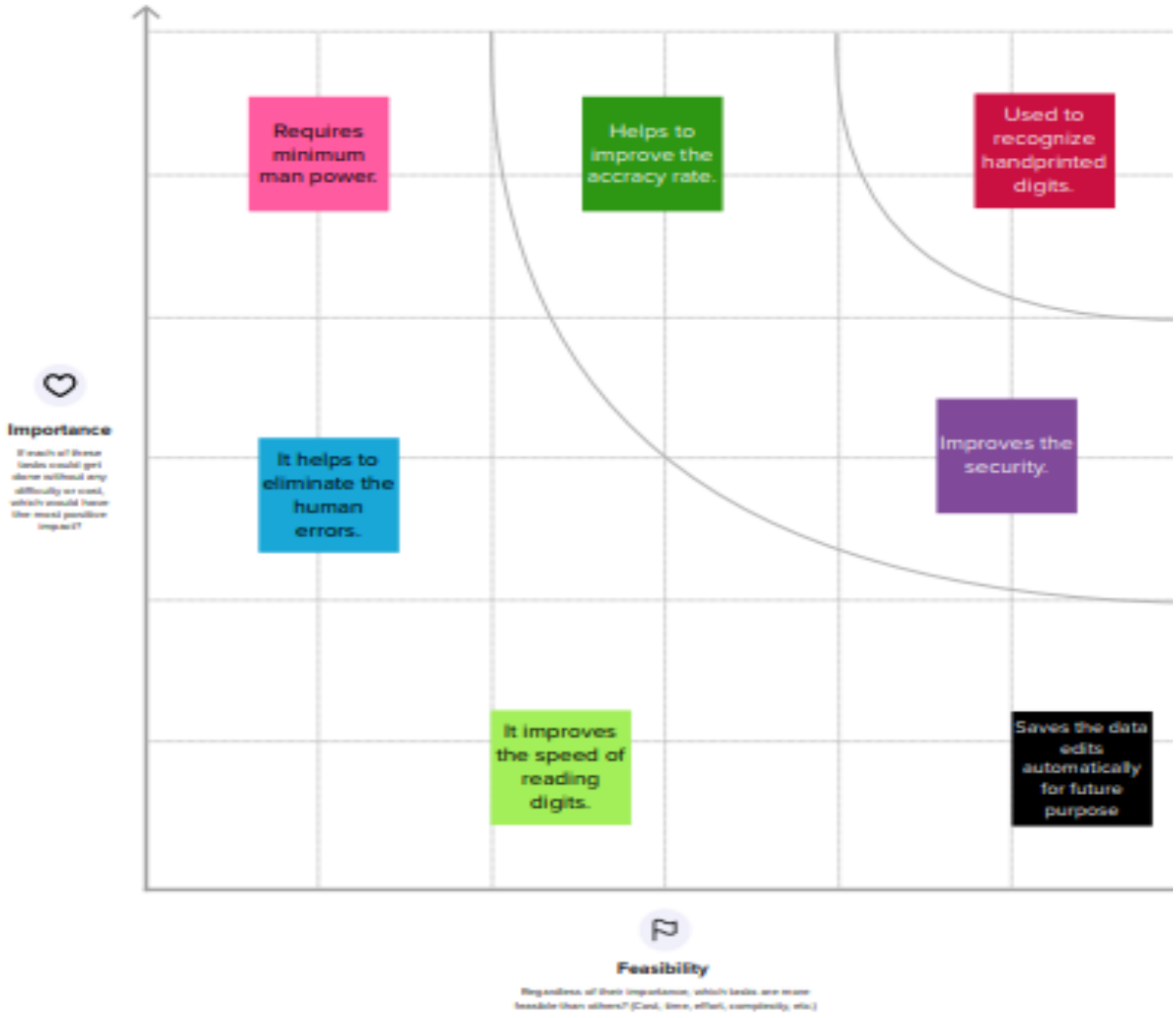
Step-3: Idea Prioritization

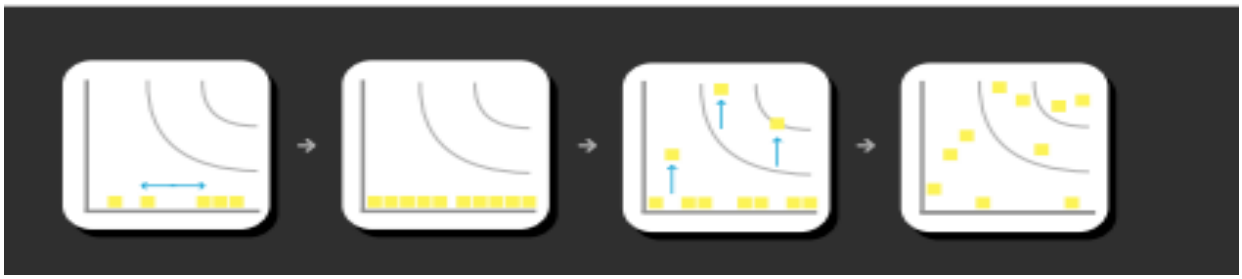
4

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

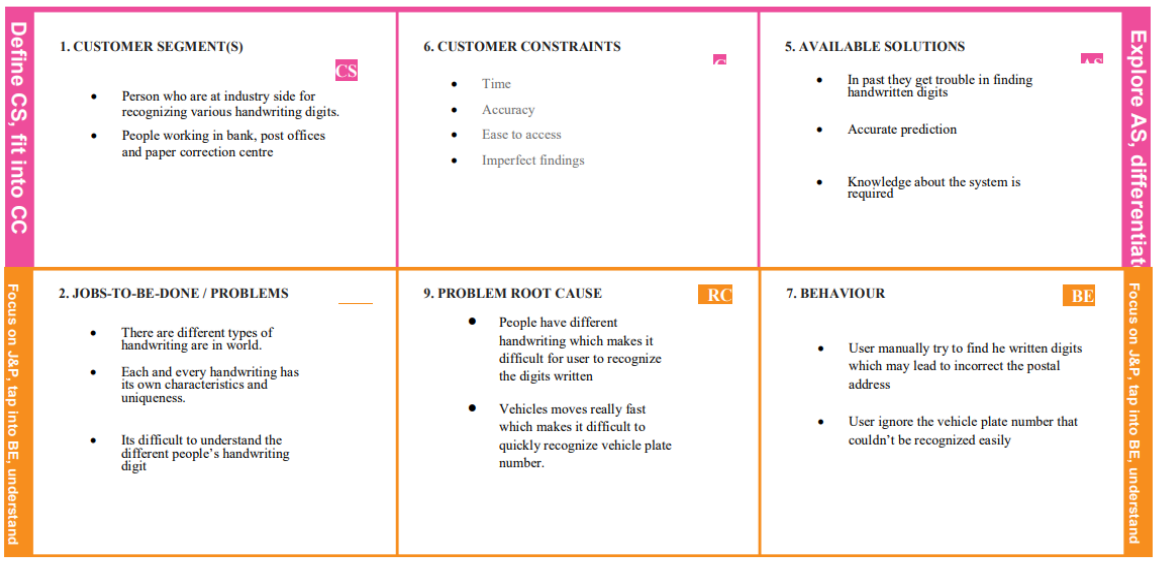




3.3 Proposed Solution.

S.No.	Parameter	Description
1.	Problem Statement	To create an application that recognizes handwritten digits.
2.	Idea / Solution description	The application takes an image as the input and accurately detects the digits in it.
3.	Novelty / Uniqueness	Instead of recognising every text, the application accurately recognizes only the digits.
4.	Social Impact / Customer Satisfaction	This application reduces the manual tasks that need to be performed. This improves productivity in the workplace.
5.	Business Model (Revenue Model)	This application can be integrated with traffic surveillance cameras to recognizes vehicle number plates. This application can be integrated with postal systems to recognizes the pin codes effectively.
6.	Scalability of the Solution	This application can easily be scaled to accept multiple inputs and process them parallelly to further increase efficiency

3.4 Problem Solution fit :



4. REQUIREMENT ANALYSIS

4.1 Functional requirement :-

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Input	GUI allows the user to input image by browsing the device storage
FR-2	Model	The MNIST dataset should be trained using CNN to create a trained model
FR-3	Prediction	The trained model has to be tested by using the test data provided by MNIST and the accuracy of the model should be above 90%
FR-4	Evaluation	Ensure that the output produced by the model is Correct

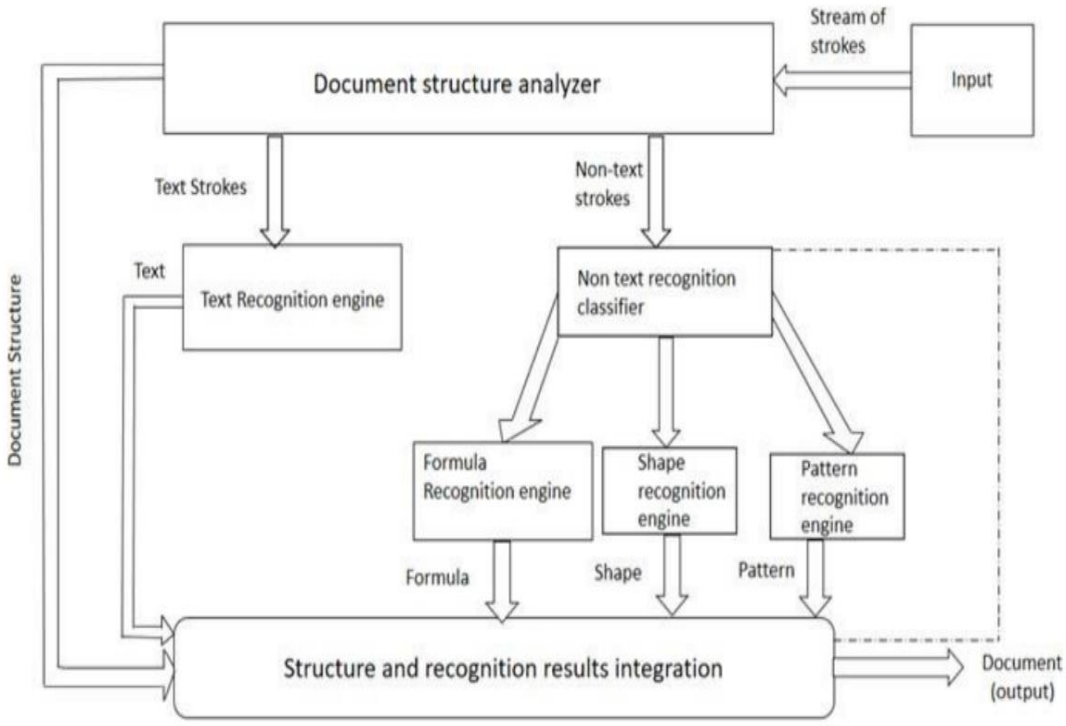
4.2 Non-Functional requirements :-

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Can predict digits with accuracy. The model can be used in bank check processing, data entry etc
NFR-2	**Security**	It ensures security as the uploaded image is not stored in any database
NFR-3	**Reliability**	Can process confidential information without data leakage as the data is never stored in any database
NFR-4	**Performance**	Improvement in fast prediction. We use CNN algorithm for accurate prediction
NFR-5	**Availability**	Available for web and mobile browsers
NFR-6	**Scalability**	Helps many individuals with low time consumption and high accuracy

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, and where data is stored.

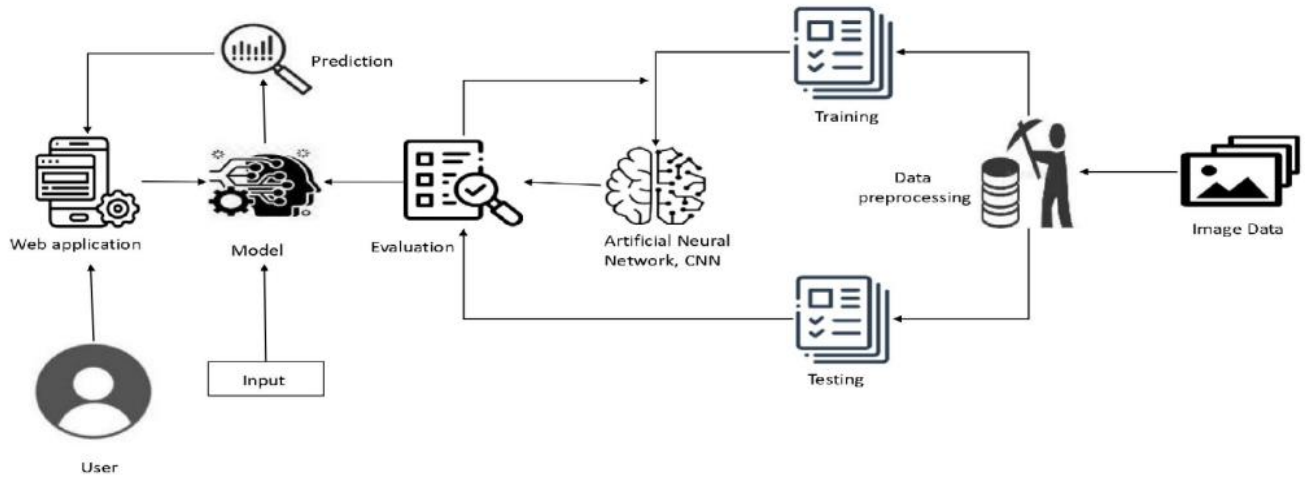


5.2 Solution & Technical Architecture

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions.

- Find the best tech solution to solve existing business problems
- Describe the structure, characteristics, behavior, and other aspects of software to project stakeholders
- Define features, development phases, and solution requirements
- Provide specifications according to which the solution is defined, managed, and delivered.

Architecture diagram :-



5.3 User Stories:

User Type	Functional Requirement (Epic)	User Story Number	User Story I Task	Acceptance criteria	Priority	Release
Customer (Web user)	Home	USN-1	In the Home Page, I can view the guidelines of how to use the website	I can view the guidelines	low	Sprint-1
	Dashboard	USN-2	As a user, I can see Home Page & Prediction Page	I can access the dashboard	Low	Sprint-2
	Choose Input	USN-3	In Prediction Page, I can upload an image of handwritten digit for prediction	I can upload my input by browsing the device storage	Medium	Sprint-3

		USN-4	As a user, I can get an accuracy r	I can get different forms	High	Sprint-4
	Recogni ze	USN-5	As a user, I can see that the GUI processing the input using trained model	I can perform handwritten digit prediction	High	Sprint-1
	Predicti on	USN-6	As a user, I can get accuracy rate by pressing the predict button	I can get the accuracy of the output	Medi um	Sprint-1
Cust omer (Mob ile user)	Home	USN-7	As a user, I can access application in mobile phone	I can access the dashboard with mobile	Medi um	Sprint-1
	Recogni ze	USN-8	I can upload input and retrieve output with accuracy by using the mobile	I can upload input image and get output with a mobile device	High	Sprint-2
Tran scrip tion anal yst	Pre Processi ng	USN-1	Noise in the digital handwritt en image can be reduced.	It uses noise filters.	High	Sprint-1
		USN-2	Blurred image can be modified.	Sobel filter can be used to sharpen the image.	High	Sprint-3
	Feature Extracti on	USN-3	How the features can be identified.	By extracting the foreground image from	Low	Sprint-2

				background image.		
		USN-4	How shape edges can be detected.	Curves of the letters can be found.	Medium	Sprint-1
		USN-5	How words are recognized based on sizes.	By identifying the size of the word.	High	Sprint-3
	Prediction	USN-6	How letters are predicted.	By comparing the features of each letter with the features of actual letters.	High	Sprint-4

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation:-

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 need to collect the data with different handwriting to train the model	6	High	Abinesh R G Meiyazhagan V
Sprint-1	Importing libraries	USN-2	As a user, I have to implement necessary libraries in python packages.	4	Low	Abinesh R G Meiyazhagan V
Sprint-1	Data preprocessing	USN-3	As a user, I can load the dataset, handle the missing values, scale and split the data.	10	Medium	Abinesh R G Meiyazhagan V

Sprint-2	Add the CNN layers	USN-5	Add input convolutional layer, maxpooling layer, flatten , hidden and output layers to the model.	5	High	Abinesh R G Kaviyarasu B
Sprint- 2	Compile the model	USN-6	As a user, compile the model for trained dataset.	2	Medium	Abinesh R G Kaviyarasu B
Sprint-2	Train and test the model	USN-7	As a user, train and test the model for the dataset collected and data are validated.	4	High	Abinesh R G Kaviyarasu B
Sprint-2	Save the model	USN-8	As a user, the compiled data are saved and integrated with an android application or web application.	2	Low	Abinesh R G Kaviyarasu B
Sprint-3	Building UI application	USN-9	As a user upload the input image that contains handwritten digits.	10	Medium	Kaviyarasu B Chandru C
Sprint-3		USN-10	As a user, I can provide the fundamental details about the usage of application to customer.	5	Low	Kaviyarasu B Chandru C
		USN-11	As a user, I can see the predicted or recognized digits in the application.	5	Medium	Kaviyarasu B Chandru C
Sprint-4	Train the model on IBM	USN-12	As a user train the model in IBM cloud and integrate the results.	10	High	Chandru C Meiyazhagan V

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	31 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	6 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	13 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

7. CODING & SOLUTIONING

```
from unittest import result
from flask import Flask,render_template,request,redirect,url_for
#
import numpy as np
from PIL import Image
from tensorflow.keras.models import load_model
#
# from tensorflow.k
```

```python
@app.route('/predict',methods=["POST","GET"])
def predict():
    if request.method == "POST":
        print(request.files['image'])
        img = Image.open(request.files['image'].stream).convert("L")
        img = img.resize((28,28))
        imgToArr = np.array(img)
        imgToArr = imgToArr.reshape(1,28,28,1)
        pred = model.predict([imgToArr])
        print(pred)
        y_pred = np.argmax(pred,axis=1)
        print("The image is "+str(y_pred))
        #return redirect('/output',message = y_pred)
        return redirect( url_for('.output',number = str(y_pred[0])))
    if request.method=="GET":
        return render_template('web.html')
```

```python
@app.route('/output',methods=["GET"])
def output():
    val = request.args.get('number')
    if  val :
        print(val)

        return render_template('result.html',result = val)
    return redirect('/')
if __name__=="__main__":
    model = load_model('Sprint 3\models\mnistCNN.h5')
# Show the model architecture
    app.run(debug=True)
```

8. TESTING

8.1.Test Cases

Test case ID	Feature Type	Component	Test Scenario	Expected Result	Actual Result	Status
HP_TC_001	UI	Home Page	Verify UI elements in the Home Page	The Home page must be displayed properly	Working as expected	PASS
HP_TC_002	UI	Home Page	Check if the UI elements are displayed properly in different screen sizes	The Home page must be displayed properly in all sizes	The UI is not displayed properly in screen size 2560 x 1801 and 768 x 630	FAIL
HP_TC_003	Functional	Home Page	Check if user can upload their file	The input image should be uploaded to the application successfully	Working as expected	PASS
HP_TC_004	Functional	Home Page	Check if user cannot upload unsupported files	The application should not allow user to select a non image file	User is able to upload any file	FAIL
HP_TC_005	Functional	Home Page	Check if the page redirects to the result page once the input is given	The page should redirect to the results page	Working as expected	PASS

BE_TC_001	Functional	Backend	Check if all the routes are working properly	All the routes should properly work	Working as expected	PASS
M_TC_001	Functional	Model	Check if the model can handle various image sizes	The model should rescale the image and predict the results	Working as expected	PASS
M_TC_002	Functional	Model	Check if the model predicts the digit	The model should predict the number	Working as expected	PASS
M_TC_003	Functional	Model	Check if the model can handle complex input image	The model should predict the number in the complex image	The model fails to identify the digit since the model is not built to handle such data	FAIL
RP_TC_001	UI	Result Page	Verify UI elements in the Result Page	The Result page must be displayed properly	Working as expected	PASS
RP_TC_002	UI	Result Page	Check if the input image is displayed properly	The input image should be displayed properly	The size of the input image exceeds the display container	FAIL
RP_TC_003	UI	Result Page	Check if the result is displayed properly	The result should be displayed properly	Working as expected	PASS
RP_TC_004	UI	Result Page	Check if the other predictions are displayed properly	The other predictions should be displayed properly	Working as expected	PASS

8.2.User Acceptance Testing

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

9.RESULTS

9.1. Performance Metrics

```
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
_________________________________________________________________
```

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

10.ADVANTAGES & DISADVANTAGES

ADVANTAGES:

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

DISADVANTAGES

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

11.CONCLUSION

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.

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.

13. APPENDIX

13.1.Source Code

App.py :

```
from unittest import result
from flask import Flask,render_template,request,redirect,url_for
#
import numpy as np
from PIL import Image
from tensorflow.keras.models import load_model
#
# from tensorflow.k
```

```
app = Flask(__name__)


@app.route('/',methods=["GET"])
def index():
    return render_template('index.html')
@app.route('/predict',methods=["POST","GET"])
def predict():
    if request.method == "POST":
        print(request.files['image'])
        img = Image.open(request.files['image'].stream).convert("L")
        img = img.resize((28,28))
        imgToArr = np.array(img)
        imgToArr = imgToArr.reshape(1,28,28,1)
        pred = model.predict([imgToArr])
        print(pred)
        y_pred = np.argmax(pred,axis=1)
        print("The image is "+str(y_pred))
        #return redirect('/output',message = y_pred)
        return redirect( url_for('.output',number = str(y_pred[0])))
    if request.method=="GET":
        return render_template('web.html')


@app.route('/output',methods=["GET"])
def output():
    val = request.args.get('number')
    if  val :
        print(val)
        return render_template('result.html',result = val)
    return redirect('/')


if __name__=="__main__":
    model = load_model('Sprint 3\models\mnistCNN.h5')
# Show the model architecture
    app.run(debug=True)
```

Index.html :

```
<!DOCTYPE html>
<head>
    <link rel="stylesheet" href="/static/homePage.css"/>

</head>
```

```
<body class="body"  >
    <div class="contentWrapper">
        <h1>Digit recognition using CNN</h1>
        <p>  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.</p>
        <p> 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.
            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.</p>
            <p>
            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>
        <a href="{{url_for('predict')}}" class="recogniseBtn">Recognise</a>
    </div>
</body>
```

Result.html

```
<!DOCTYPE html>
<head>
    <link rel="stylesheet" href="../static/homePage.css"/>
<script src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.5.0/Chart.min.js"></script>
    <script>


        var color=['#FFEC00', '#FF7300','#FF0000','#52D726','#007ED6','#7CDDDD', '#7CDDDD',
'#F54F52','#F54F52']
        var barChart = {
          type: 'bar',
          data: {
            labels: color,
            datasets: [
              {
                label: "Numbers",
                backgroundColor: color; data: '{{result}}'



          },
```

```
        options: {
          responsive:true
        }
      };
      window.onload = function() {
          var bar = document.getElementById('myChart').getContext('2d');
          window.myPie = new Chart(bar, barChart);
      };
    </script>
</head>

<body class="body"  >
    <div class="contentWrapper">
        <center><h1>The Predicted number is</h1></center>
        {% if result %}
            <p><h3>output predicted : {{result}}</h3></p>
        {% else %}
            <p>No output Predicted Please enter a proper Image</p>
{% endif %}
        <canvas id="myChart" width="400" height="400"></canvas>
    </div>
</body>
```

13.2 GITHUB Link :

https://github.com/IBM-EPBL/IBM-Project-50744-1660923058.git

13.3 Project Demo Link:

https://youtu.be/bey92qQCUfg