Project Report Format

1. **INTRODUCTION**
    - 1.1 Project Overview
    - 1.2 Purpose
2. **LITERATURE SURVEY**
    - 2.1 Related works
    - 2.2 References
    - 2.3 Problem Statement Definition
3. **IDEATION & PROPOSED SOLUTION**
    - 3.1 Empathy Map Canvas
    - 3.2 Ideation & Brainstorming
    - 3.3 Proposed Solution
    - 3.4 Problem Solution fit
4. **REQUIREMENT ANALYSIS**
    - 4.1 Functional requirement
    - 4.2 Non-Functional requirements
5. **PROJECT DESIGN**
    - 5.1 Data Flow Diagrams
    - 5.2 Solution & Technical Architecture
    - 5.3 User Stories
6. **PROJECT PLANNING & SCHEDULING**
    - 6.1 Sprint Planning & Estimation
    - 6.2 Sprint Delivery Schedule
    - 6.3 Reports from JIRA
7. **CODING & SOLUTIONING (Explain the features added in the project along with code)**
    - 7.1 Feature 1
    - 7.2 Feature 2
    - 7.3 Database Schema (if Applicable)
8. **TESTING**
    - 8.1 Test Cases
    - 8.2 User Acceptance Testing
9. **RESULTS**
    - 9.1 Performance Metrics

INTRODUCTION:

Handwriting digits and character recognitions have become increasingly important in today's digitized world due to their practical applications in various day to day activities. It can be proven by the fact that in recent years, different recognition systems have bee developed or proposed to be used in different fields where high classification efficiency is needed. Systems that are used to recognize Handwriting letters, characters, and digits help people to solve more complex tasks that otherwise would be time-consuming and costly.

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.

1.3. A Good example is the use of automatic processing systems used in banks process bank cheques. Without automated bank cheque

processing systems, the bank would be required to employ many

employees who may not be as efficient as the computerized

Processing system.

The human visual system is primarily involved whenever individuals

are reading Handwriting characters, letters, words, or digits. It seems

effortless whenever one is reading handwriting, but it is not as easy

as people believe. A human can make sense of

what they see based on what their brains have been

taught, although everything is done unconsciously. A

human may not appreciate how difficult it is to solve

Handwritting.

2. LITERATURE SURVEY:

2.1 EXISTING PROBLEM

* The main objective of this research is design an expert system for

Handwriting character recognition using neural network approach.

* To address the issue of accuracy in Handwriting

character recognition systems by developing a system that will use

efficient technology for recognizing Handwriting characters and words

from image media.

* To investigate and demonstrate the usefulness of neural

network technology in development of efficient Handwriting characterr

recognition systems.

2.2 Reference:

[1] Fischer, A., Frinken, V., Bunke, H.: Hidden Markov

models for off-line cursive handwriting

recognition, in C.R. Rao (ed.): Handbook of

Statistics 31, 421 – 442, Elsevier, 2013.

[2] Frinken, V., Bunke, H.: Continuous handwritten

script recognition, in Doermann, D., Tombre, K.

(eds.): Handbook of Document Image Processing

and Recognition, Springer Verlag, 2014.

[3] S. Günter and H. Bunke. A new combination scheme for HMM-based classifiers and its application to handwriting recognition. In Proc. 16th Int. Conf. on Pattern Recognition, volume 2, pages 332–337. IEEE, 2002.

[4] U.-V. Marti and H. Bunke. Text line segmentation and word recognition in a system for general writer independent handwriting recognition. In Proc. 6th Int. Conf. on Document Analysis and Recognition, pages 159– 163.

[5] M. Liwicki and H. Bunke, “Iam-ondb - an on-line English sentence database acquired from the handwritten text on a whiteboard,” in ICDAR,

2005.

[6] A. Graves and J. Schmidhuber, “Offline handwriting recognition with multidimensional recurrent neural networks,” in Advances in neural information processing systems, 2009, pp. 545–552.

[7] Nafiz Arica, and Fatos T. Yarman-Vural, —Optical Character Recognition for Cursive Handwriting,‖ IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.24, no.6, pp. 801-113, June 2002.

[8] Anita Pal and Davashankar Singh,”Handwritten English Character Recognition Using Neural

Network", International Journal of ComputerScience and Communication, pp: 141- 144, 2011.

[9] Sandhya Arora, "Combining Multiple Feature

Extraction Techniques for Handwritten Devnagari

Character Recognition", IEEE Region 10

Colloquium and the Third ICIIS, Kharagpur,

INDIA, December 2008.

[10]Om Prakash Sharma, M. K. Ghose, Krishna Bikram

Shah, "An Improved Zone Based Hybrid Feature

Extraction Model for Handwritten Alphabets

Recognition Using Euler Number", International

Journal of Soft Computing and Engineering

(ISSN: 2231 - 2307), Vol. 2, Issue 2, pp. 504- 508,

May 2012.

[11]N. Venkata Rao and Dr. A.S.C.S.Sastry - Optical

Character Recognition Technique Algorithms‖-

2016 Journal of Theoretical and Applied

Information Technology.

[12]Ganapathy, V., & Liew, K. L. (2008). Handwritten

Character Recognition Using Multiscale Neural

Network Training Technique. World Academy of

Science, Engineering and Technology

International Journal of Computer and

Information Engineering, 2(3), 638-643.

[13]Grady, J. O. (2010). System Requirements

Analysis. Amsterdam, Netherlands: Elsevier.

[14]Gribaudo, M. (2013). Theory and Application of

Multi-Formalism Modeling. Hershey, PA: IGI

Global

2.3 Problem Statement Definition:

Handwritten recognition system has the capacity to identify and understand handwritten digits or characters automatically .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 PyTorch library over the MNIST dataset to recognize handwritten digits .

3. IDEATION AND PROPOSED SOLUTION:

3.1 Empathy map canvas:

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

Creating an effective 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 persective along with his or her goals and challenges.

3.2 Ideation and brainstorming:

Template

Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

10 minutes to prepare
1 hour to collaborate
2-8 people recommended

Share template feedback

Before you collaborate

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

10 minutes

A **Team gathering**
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B **Set the goal**
Think about the problem you'll be focusing on solving in the brainstorming session.

C **Learn how to use the facilitation tools**
Use the Facilitation Superpowers to run a happy and productive session.

Open article →

1 Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

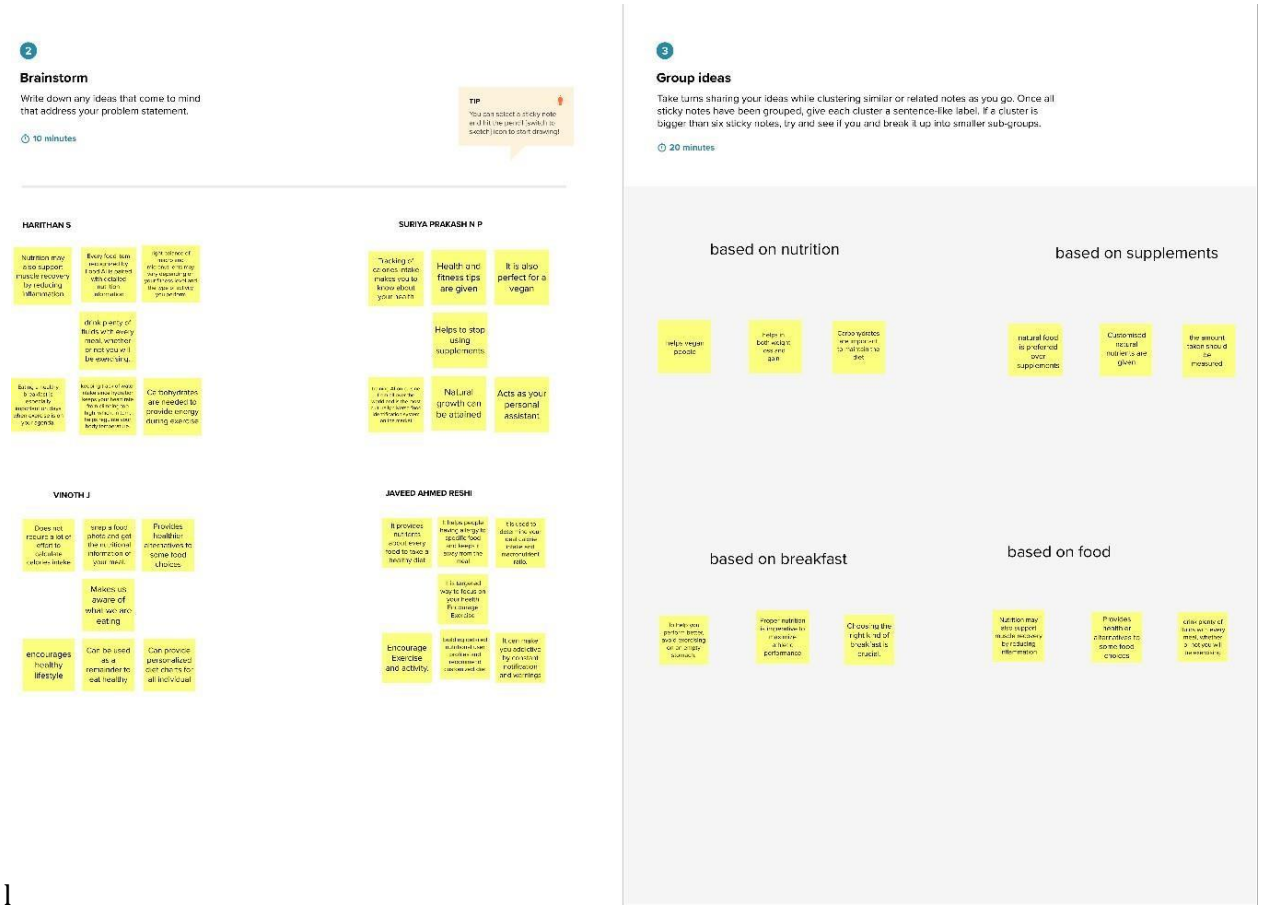
5 minutes

PROBLEM
How might we [your problem statement]?

Key rules of brainstorming

To run an smooth and productive session

- Stay in topic.
- Defer judgment.
- Go for volume.
- Encourage wild ideas.
- Listen to others.
- If possible, be visual.



l

3.3 Proposed solution

Parameter Description

1. Problem Statement (Problem to be solved)

Making the computers capable of identifying and understanding handwritten digits or characters automatically.

2. Idea / Solution description

We use artificial neural networks to train these images and build a deep learning model.

3. Novelty / Uniqueness

Image of handwritten notes are uploaded using web application, this model is analyzed by the model and the detected result is returned on to UI.

4. Social Impact / Customer Satisfaction

There is a need in real time applications for handwritten digit recognition. Our app helps to overcome it and also gets the feedback from the customers.

5. Business Model (Revenue Model)

Platform is free to afford. For watermark free output document the customers want to afford separately.

6. Scalability of the Solution

Address the problem by using online handwriting data collection for a large scale production online handwriting recognition system.

3.4 Problem solution fit:

4.REQUIREMENT ANALYSIS

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Getting input from the user.	The handwritten digit is obtained as input from the user as an image uploading or writing on the canvas.
FR-2	Data preprocessing.	Upgrades the image to make it ready for the segmentation by performing some tasks on the input image.
FR-3	Segmentation and data feature extraction.	Segment the MNIST dataset images using the edge detection technique and remove the redundancy from the data.
FR-4	Classification and Recognition.	Passing the feature vectors as individual input to the classifiers or the neural networks such as the CNN model.
FR-5	Prediction.	The deep learning model is trained and tested using the MNIST dataset with the accuracy greater than 90%
FR-6	Evaluation.	Ensure that the digit is correctly recognised by the model and produces accurate output.

Non-functional Requirements:

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

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	To identify and understand handwritten digits automatically with high accuracy.
NFR-2	**Security**	Ensures security, since uploaded images are not stored in any database.
NFR-3	**Reliability**	User-friendly web interface for the system. Process confidential information without data leakage.
NFR-4	**Performance**	High, since artificial neural networks are used to train the images and build deep learning model. Fast prediction using CNN algorithm.
NFR-5	**Availability**	Using web application, anyone can easily access the system, making it highly available for web and mobile browsers.
NFR-6	**Scalability**	Performs well even if the count of input handwriting increased, since MNIST dataset is used for the recognition process. Low time consumption.

5.PROJECT DESIGN

5.1 DATA FLOW DIAGRAM

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

DFD Level-0

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

DFD Level-1

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

DFD Level-2

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

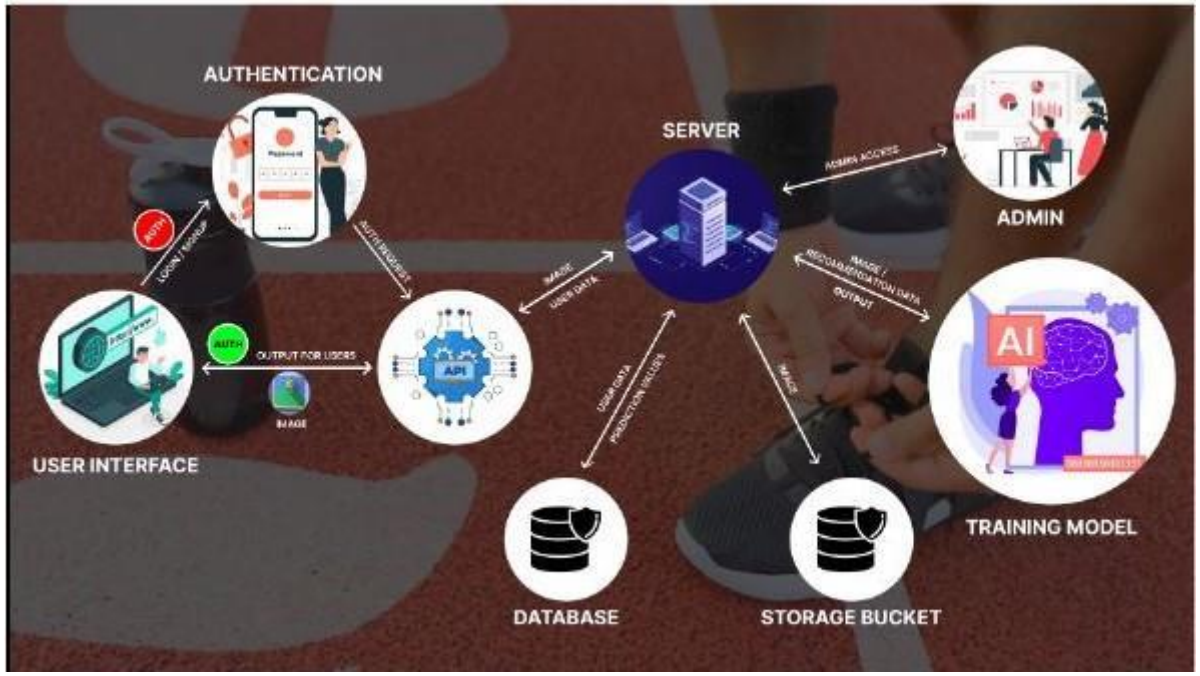
5.2 SOLUTION AND TECHNICAL ARCHITECTURE:

Solution Architecture:

Solution architecture is a complex process – with many sub-processes – that bridges

the gap between business problems and technology solutions.

Solution Architecture Diagram:



5.3 USER STORIES:

The six cases perform differently because of the various shrinkage convolutional neural network for adaptive noise reduction," IEEE Signal Processing Letters, vol. 25, no. 2, pp. 224-228, 2018.combinations of hidden layers. The layers were taken [17] C. C. Park, Y. Kim, and G. Kim, "Retrieval of sentence randomly in a periodic sequence so that each case behaves sequences for an image stream via coherence recurrent differently during the experiment. The maximum and convolutional networks," IEEE transactions on pattern analysis minimum accuracies were observed for different hidden and machine intelligence, vol. 40, no. 4, pp. 945-957, 2018.layers variation with a batch size of 100. Among all the [18] Y. Yin, J. Wu, and H. Zheng, "Ncfm: Accurate handwritten digits recognition using convolutional neural networks," in 2016

observation, the maximum accuracy in the performance was International Joint Conference on Neural Networks (IJCNN), found 99.21% for 15 epochs in case 2 (Conv1, po

The six cases perform differently because of the various shrinkage convolutional neural network for adaptive noise reduction," IEEE Signal Processing Letters, vol. 25, no. 2, pp. 224-228, 2018.combinations of hidden layers. The layers were taken [17] C. C. Park, Y. Kim, and G. Kim, "Retrieval of sentence randomly in a periodic sequence so that each case behaves sequences for an image stream via coherence recurrent differently during the experiment. The maximum and convolutional networks," IEEE transactions on pattern analysis minimum accuracies were observed for different hidden and machine intelligence, vol. 40, no. 4, pp. 945-957, 2018.layers variation with a batch size of 100. Among all the [18] Y. Yin, J. Wu, and H. Zheng, "Ncfm: Accurate handwritten digits recognition using convolutional neural networks," in 2016 observation, the maximum accuracy in the performance was International Joint Conference on Neural Networks (IJCNN), found 99.21% for 15 epochs in case 2.

6. PROJECT PLANNING AND SCHEDULING:

6.1 Sprint Planning and Estimation:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data preprocessing	USN-1	Collect and load the dataset, scaling and wrangling the data and split into train and testdata.	20	Medium	Gopika ShreeS, Kavitha P
Sprint-2	Model building	USN-2	Building the deep learning model with CNN to recognize the handwritten digit with higheraccuracy.	10	High	Nafeesa Fathia S, Kokila M
Sprint-2		USN-3	Training and Testing the model with the splitimage dataset and refine the accuracy.	5	Medium	Nafeesa Fathima S, Gopika ShreeS

Spriint-2		USN-4	Model save and used for further integration withthe Flask web user interface.	5	Medium	Kokila M, Kavita P.
Sprint-3	UI Application	USN-5	Building a Flask web application interface to upload the handwritten image by clicking theupload button.	5	Medium	Kokila M, Gopika ShreeS.
Sprint-3		USN-6	Integrating the Flask web application with thesaved deep learning model.	10	High	Nafee sa Fathia S, Kavita P.
Sprint-3		USN-7	As a user, I can see the predicted digits with theaccuracy in the web application.	5	Medium	Kavitha P.

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint 4	Train the model	USN-8	Train the model on IBM cloud and deploy theFlask web application with scoring and point.	20	High	Nafeesa Fathima S, Gopika ShreeS, Kokila M

6.2.Sprint Delivery Schedule:

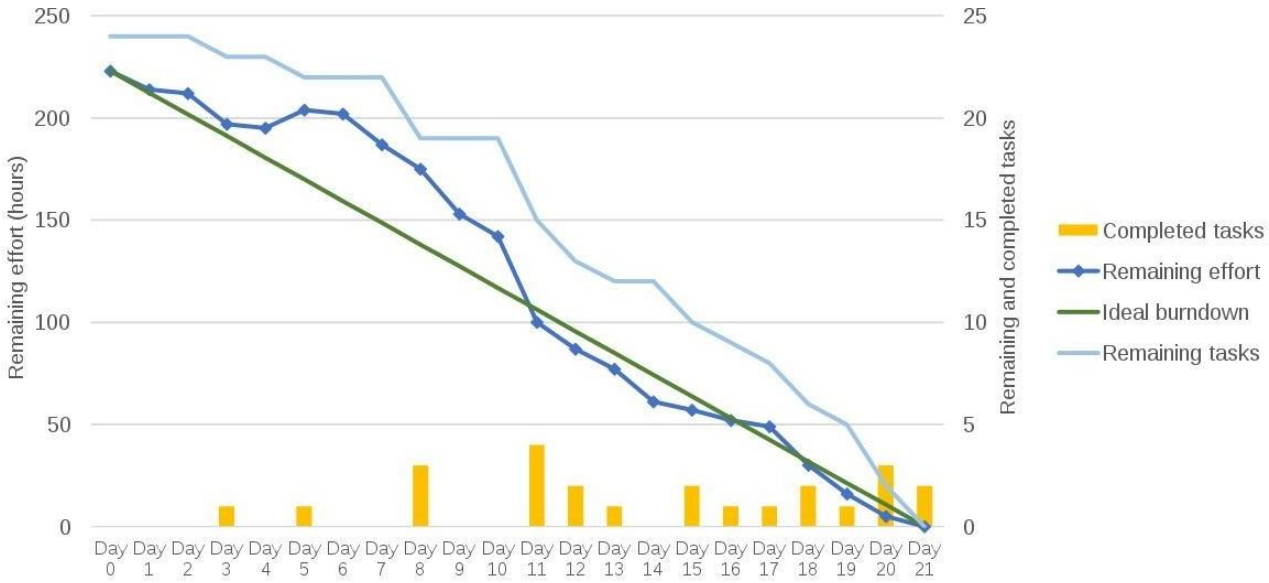
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6	24 Oct	29 Oct 2022	20	29 Oct 2022

		Days	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

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

6.3.Reports from JIRA:

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



7. CODING AND SOLUTION:

7.1 Feature 1

Depending on the features given to the classifier, it accumulates knowledge base for classification purposes. In case of a binary image, when translated into an array and used as an attribute, no information is given to the classifier about the order of the attributes. It would in fact be irrelevant if all the patterns are shuffled in the same manner and presented to the classifier for classification purposes.

7.2 Feature 2

If such information is given to the classifier, its performance can be improved. One such feature (Pixel Count Feature) is obtained by counting row-wise, number of black pixels present and doing same column-wise, thus obtaining two profiles. For example the row profile □ of dimensions (1xN) can be obtained from complemented binary image □ of □×□ pixels, where 0s and 1s represent white and black pixels.

7.3Database Schema

The MNIST database contains 60,000 training images and 10,000 testing images. Half of the training set and half of the test set were taken from NIST's training dataset, while the other half of the training set and the other half of the test set were taken from NIST's testing dataset. The original creators of the database keep a list of some of the methods tested on it. In their original paper, they use a support-vector machine to get an error rate of 0.8%.

8. TESTING

8.1 Test Cases

Sensitivity Sensitivity:

can be defined as the effectiveness of classifier to identify positive labels. This is also known as recall. Sensitivity = (TP)/ (TP+FN)

Specificity:

This is defined as the effectiveness of classifier to correctly identify negative labels. Specificity = (TN) / (FP + TN) Both Sensitivity and specificity lie between 0 and 1, 1 is an ideal value for each of them. We calculate balanced accuracy as taking an average of sensitivity and specificity.

Prevalence:

Well, how often does the “yes” condition actually occur in our sample? Prevalence = (TP + FN) / N N is the sum of all conditions i.e. TP, FN, FP, TN.

Positive predicted values :

The portion of correct value results in labels identified as positive.
Positive_predicted_value = (Sensitivity * Prevalence) / ((Sensitivity * prevalence) + (1 — specificity) * (1 — prevalence))

Negative predicted values:

The proportion of correct results in the label identified as negative.
Negative_predicted_values = Specificity *(1 — prevalence) / (((1-sensitivity)*prevalence) + (specificity * (1 — prevalence)))

8.2 User Acceptance Testing:

Detection rate:

Detection rate is the division of true positives by the total number of conditions. DR = TP / N.

Expected accuracy:

Also considered as random chance among the conditions
Expected_accuracy = ((TP + FN) * (TP+FP) + (FP+TN) * (FN+TN)) / N
Where N is sum of all conditions i.e. TP, FN, FP, and TN.

Kappa statistic:

The Kappa statistic (or value) is a metric that compares an observed accuracy with an expected accuracy (say random chance). Kappa =

(Observed accuracy — expected_accuracy) / (1 — expected_accuracy)

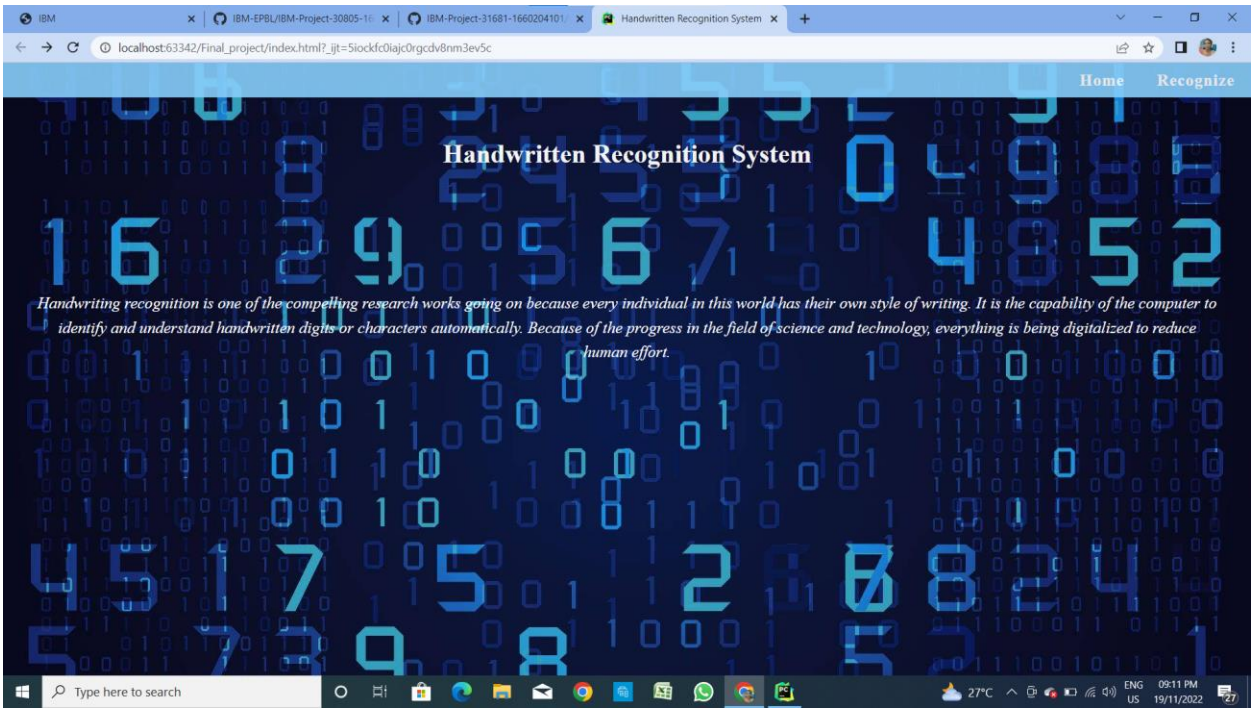
There are a lot of performance metrics for a classifier to show how well it performs in these statistical situations.

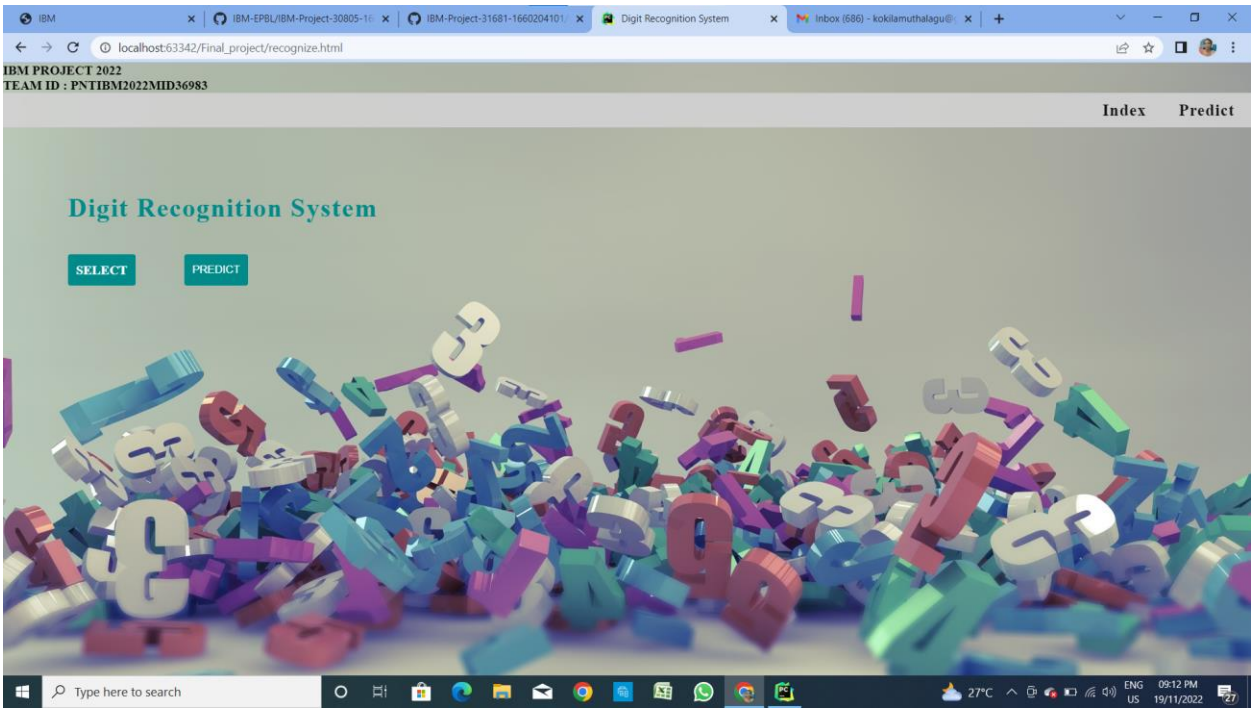
9. RESULTS

9.1 Performance metrics:

For research purpose, or applying the classifiers to real scenario problems. Accuracy and speed of recognition are considered the better measure.

Talking about the different classifiers one by one now.





10. ADVANTAGES

This approach has many advantages: 1) 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; 2) the generative models can perform recognition driven segmentation; 3) the method involves a relatively.

DISADVANTAGES

This implementation is good because it doesn't require a training system but the main disadvantage is the problem of classifying the strokes. There are problems with clockwise and counter-clockwise digits.

11.CONCLUSION:

In this research, we have implemented three models for handwritten digit recognition using MNIST datasets, based on deep and machine learning algorithms. We compared them based on their characteristics to appraise the most accurate model among them. Support vector machines are one of the basic classifiers that’s why it’s faster than most algorithms and in this case, gives the maximum training accuracy rate but due to its simplicity, it’s not possible to classify complex and ambiguous images as accurately as achieved with MLP and CNN algorithms. We have found that CNN gave the most accurate results for handwritten digit recognition. So, this makes us conclude that CNN is best suitable for any type of prediction problem including image data as an input. Next, by comparing execution time of the algorithms we have concluded that increasing the number of epochs without changing the configuration of the algorithm is useless because of the limitation of a certain model.

12.FUTURE SCOPE:

The future development of the applications based on algorithms of deep and machine learning is practically boundless. In the future, we can work on a denser or hybrid algorithm than the current set of algorithms with more manifold data to achieve the solutions to many problems. In future, the application of these algorithms lies from the public to high-level authorities, as from the differentiation of the algorithms above and with future development we can attain high-level functioning applications which can be used in the classified or government agencies as well as for the common people, we can use these algorithms in hospitals application for detailed medical diagnosis, treatment and monitoring the patients, we can use it in surveillances system to keep tracks of the suspicious activity under the system, in fingerprint and retinal scanners, database filtering applications, Equipment checking for national forces and many more problems of both major and minor category.

13.APPENDIX:

Source code:

IMPORTING LIBRARIES

```
#for working with arraysimport numpy as np#open source used for both ML and DL for computationimport tensorflow#mnist datasetfrom tensorflow.keras.datasets import mnist#it is a plain stack of layersfrom
```

```
tensorflow.keras.models import Sequential#A Layer consists of a tensor- in tensor-out computat ion functionfrom tensorflow.keras import layers#Dense-Dense Layer is the regular deeply connected layers#faltten -used fot flattening the input or change the dimensionfrom tensorflow.keras.layers import Dense, Flatten #Convolutional Layerfrom tensorflow.keras.layers import Conv2D#Optimizerfrom keras.optimizers import Adam #Used for one-hot encodingfrom keras. utils import np_utils#for data visualizationimport matplotlib.pyplot as plt
```

LOADING DATA

In [4]:

```
#splitting the mnist data into train and test(x_train, y_train), (x_test, y_test)=mnist.load_data()
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz
11490434/11490434 [==============================] - 0s 0us/step
```

In [5]:

```
#shape is used for give the dimension values #60000-rows 28x28-pixelsprint(x_train.shape)print(x_test.shape)
```

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

In [6]:

```
x_train[0]
```

Out[6]:

```
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, 253, 251,  93,  82,  82,  56,  39,   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],
[  0,   0,   0,   0,   0,   0,   0,   0,   0,  14,   1, 154, 253,
  90,   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, 139, 253,
 190,   2,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,  11, 190,
 253,  70,   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,  35,
 241, 225, 160, 108,   1,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0],
```

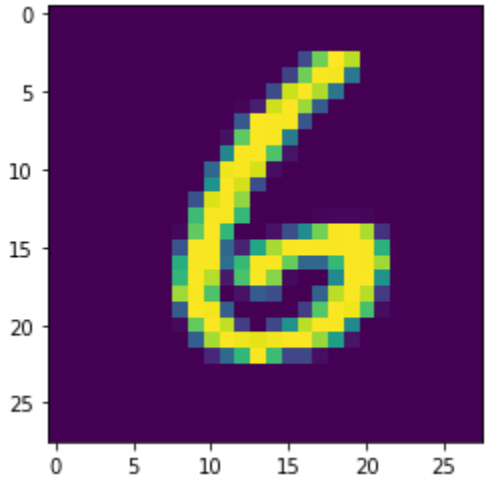
```
[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
  81, 240, 253, 253, 119,  25,   0,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,  45, 186, 253, 253, 150,  27,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0,  16,  93, 252, 253, 187,   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, 249, 253, 249,  64,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,  46, 130, 183, 253, 253, 207,   2,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,  39,
 148, 229, 253, 253, 253, 250, 182,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,  24, 114, 221,
 253, 253, 253, 253, 201,  78,   0,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,  23,  66, 213, 253, 253,
 253, 253, 198,  81,   2,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,  18, 171, 219, 253, 253, 253, 253,
 195,  80,   9,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,  55, 172, 226, 253, 253, 253, 253, 244, 133,
  11,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0, 136, 253, 253, 253, 212, 135, 132,  16,   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]], dtype=uint8)
```

In [7]:

```
#Plotting the imageplt.imshow(x_train[6000])
```

Out[7]:



In [8]:

```
np.argmax(y_train[6000])
```

Out[8]:

0

Reshaping dataset

In [9]:

```
#Reshaping to format which CNN expects (batch, height, width,
channels)x_train=x_train.reshape (60000, 28, 28,
```

```
1).astype('float32')x_test=x_test.reshape (10000, 28, 28, 1).astype
('float32')
```

In [10]:

```
#Storing number of classes in a variablenumber_of_classes = 10
```

In [11]:

```
#converts the output in binary formaty_train = np_utils.to_categorical
(y_train, number_of_classes)y_test = np_utils.to_categorical (y_test,
number_of_classes)
```

Add CNN Layers

In [12]:

```
#create modelmodel=Sequential ()
```

In [13]:

```
#adding modeL Layermodel.add(Conv2D(64, (3, 3), input_shape=(28, 28, 1),
activation='relu'))model.add(Conv2D(32, (3, 3), activation = 'relu'))
```

In [14]:

```
#flatten the dimension of the imagemodel.add(Flatten())
```

In [15]:

```
#output layer with 10 neuronsmodel.add(Dense(number_of_classes,activation =
'softmax'))
```

Compiling the model

In [16]:

```
#Compile modelmodel.compile(loss= 'categorical_crossentropy',
optimizer="Adam", metrics=['accuracy'])x_train = np.asarray(x_train)y_train
= np.asarray(y_train)
```

Training the model

In [19]:

```
#fit the modelmodel.fit(x_train, y_train, validation_data=(x_test, y_test),
epochs=5, batch_size=32)
```

```
Epoch 1/5
1875/1875 [==============================] - 189s 101ms/step - loss: 0.0247 - accuracy: 0.9923 - val_loss: 0.1239 - val_accuracy: 0.9763
Epoch 2/5
1875/1875 [==============================] - 185s 99ms/step - loss: 0.0227 - accuracy: 0.9931 - val_loss: 0.1384 - val_accuracy: 0.9766
Epoch 3/5
1875/1875 [==============================] - 184s 98ms/step - loss: 0.0192 - accuracy: 0.9944 - val_loss: 0.1439 - val_accuracy: 0.9734
Epoch 4/5
1875/1875 [==============================] - 186s 99ms/step - loss: 0.0167 - accuracy: 0.9949 - val_loss: 0.1475 - val_accuracy: 0.9786
Epoch 5/5
1875/1875 [==============================] - 187s 100ms/step - loss: 0.0153 - accuracy: 0.9956 - val_loss: 0.2663 - val_accuracy: 0.9695
```

Out[19]:

In [20]:

```
# Final evaluation of the modelmetrics = model.evaluate(x_test, y_test, verbose=0)print("Metrics (Test loss &Test Accuracy) : ")print(metrics)
```

```
Metrics (Test loss &Test Accuracy) :
[0.2662925720214844, 0.9695000052452087]
```

In [24]:

```
prediction=model.predict(x_test[6000:6001])print(prediction)
```

```
1/1 [==============================] - 0s 21ms/step
[[1.90929128e-22 3.87971029e-33 1.21386323e-17 4.28218216e-09
  1.16075036e-07 1.51229324e-11 4.69814482e-31 3.22366759e-07
  1.32163915e-08 9.99999523e-01]]
```

In [32]:

```
#printing our Labels from first 4 imagesimport numpy as npprint(np.argmax(prediction, axis=1))
```

```
[9]
```

In [33]:

```
#Printing the actual labelsnp.argmax(y_test[6000:6001])
```

Out[33]:

```
9
```

Save the model

In [35]:

```
# Save the modelmodel.save('handwritten/mnistCNN.h5')
```

In [37]:

In [ ]:

GITHUB AND PROJECT DEMO LINK:

https://github.com/IBM-EPBL/IBM-Project-30805-1660190183/blob/38d72e1346d52be7ecaf5df6b09201e54e672780/Project%20development%20phase/Sprint%202/Sprint_2.ipynb