ALAGAPPA CHETTIAR GOVERNMENT COLLEGE OF ENGINEERING AND TECHNOLOGY

(An Autonomous Institution Affiliated to Anna University, Chennai)

KARAIKUDI – 630003

PROFESSIONAL READINESS FOR INNOVATION EMPLOYABLITY AND ENTERPRENEURSHIP

IBM PROJECT REPORT

Submitted by

Team ID: PNT2022TMID06091

GOPINATH R - 1915014

ANUSUYA K - 1915003

JANANI S - 1915018

SAVITHA SREE L - 1915033

In partial fulfillment for the award of the degree

Of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

NOVEMBER 2022

ALAGAPPA CHETTIAR GOVERNMENT COLLEGE
OF
ENGINEERING AND TECHNOLOGY

(An Autonomous Institution Affiliated to Anna University, Chennai)

KARAIKUDI – 630003

BONAFIDE CERTIFICATE

Certified that this PROJECT REPORT **“A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM”** is the bonafide work of **GOPINATH R ANUSUYA K JANANI S SAVITHA SREE L** for **IBM NALAIYATHIRAN** in VII semester of B.E., degree course in Computer Science and Engineering branch during the academic year of 2022 - 2023.

Staff-In charge

Dr.C.Uma Rani

Evaluator

Mrs.Syed Suhaila

Head of the Department

Mrs.K.Chandraprabha

ACKNOWLEDGEMENT

We expressour breathless thanks to our **Dr.P.K.PALANI, M.E, Ph.D,** the Dean, Alagappa Chettiar Government College Of Engineering And technologyfor giving constant motivation in succeeding in our goal.

We acknowledge our sincere thanks to Head of the Department **MRS.K.CHANDRAPRABHA, M.E ,**for giving us valuablesuggestion and help towards us throughout thisProject.

We are highly grateful to thank our Project coordinator **DR.C.UMA RANI,** andour Project Evaluator**MRS.SYED SUHAILA,**Department of Computer Scienceand Engineering Alagappa Chettiar Government College Of Engineering And technology, for the coordinating us throughout this Project.

We are very much indebted to thank all the faculty members of Department of Computer science and Engineering in our Institute, for their excellent moral supportand suggestions to completeour Project work successfully.

CONTENTS

1.INTRODUCTION

1.1 PROJECT OVERVIEW:

Handwriting recognition is the ability of a machine to receive and interpret handwritten input from multiple sources like paper documents, photographs, touch screen device etc. Recognition of handwritten and machine characters is an emerging area of research and finds extensive applications in banks, offices and industries. The main aim of this project is to design expert system for , “HCR using Neural Network” that can effectively recognize a particular character of type format using the Artificial Neural Network approach. Neural computing Is comparatively new field, and design components are therefore less well specified than those of other architectures. Neural computers implement data parallelism. Neural computer are operated in way which is completely different from the operation of normal computers. Neural computer are trained (not Programmed) so that given a certain starting state (data input); they either classify the input data into one of the number of classes or cause the original data to evolve in such a way that a certain desirable property is optimized.

1.2 PURPOSE:

This application is useful for recognizing all character (English) given as in input image. Once input image of character is given to proposed system, then it will recognize input character which is given in image. Recognition and classification of characters are done by Neural Network. The main aim of this project is to effectively recognize a particular character of type format using the Artificial Neural Network approach.

2.LITERATURE SURVEY

2.1 EXISTING PROBLEM:

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. Plus, sometimes, characters look very similar, making it hard for a computer to recognise accurately. Joined-up handwriting is another challenge for computers. When your letters all connect, it makes it hard for computers to recognise individual characters. Consider, for instance, an ‘r’ and an ‘n’. Joined up, these letters could be mistaken for an ‘m’. In the case of handwriting recognition from photos, there are also awkward angles to consider. The angle the photo is taken could obscure the character, making it harder for the computer to identify.

2.2 REFERENCES:

1. Isha Vats, Shamandeep Singh, “Offline Handwritten English Numerals Recognition using Correlation Method”, International Journal of Engineering Research and Technology (IJERT): ISSN: 2278-0181 Vol. 3 Issue 6, June 2014. Access Date: 09/07/2015.

2. Gunjan Singh,Sushma Lehri, “ Recognition of Handwritten Hindi

Characters using Back propagation Neural Network”,International Journal of Computer Science and Information Technologies ISSN 0975-9646, Vol. 3 (4) , 2012,4892-4895. Access Date:09/07/2015.

3. S S Sayyad, Abhay Jadhav, Manoj Jadhav, Smita Miraje, Pradip Bele, Avinash Pandhare, ‘Devnagiri Character Recognition Using Neural Networks” ,International Journal of Engineering and Innovative Technology (IJEIT)Volume 3, Issue 1, July 2013. Access Date: 09/07/2015.

4. Shabana Mehfuz,Gauri katiyar, ‘Intelligent Systems for Off-Line Handwritten Character Recognition: A Review” ,International Journal of Emerging Technology and Advanced Engineering Volume 2, Issue 4, April 2012. Access Date: 09/07/2015.

5. Prof. Swapna Borde, Ms. Ekta Shah, Ms. Priti Rawat, Ms. Vinaya Patil, “Fuzzy Based Handwritten Character Recognition System” ,International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622,VNCET 30 Mar’12. Access Date: 09/07/2015.

6. Rahul KALA, Harsh VAZIRANI, Anupam SHUKLA and Ritu TIWARI, “An Overview of Character Recognition Focused on Off-LineHandwriting”,IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS PART APPLICATIONS AND REVIEWS, VOL. 31, NO. 2, MAY 2001. Access Date:09/07/2015.

7. Ms. Seema A. Dongare , Prof. Dhananjay B. Kshirsagar, Ms. Snehal V.

Waghchaure ,“Handwritten Devanagari Character Recognition using Neural Network ” ,IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661, p- ISSN: 2278-8.

8. Mitrakshi B. Patil, Vaibhav Narawade, “Recognition of Handwritten Devnagari Characters through Segmentation and Artificial neural networks” ,International Journal of Engineering Research and Technology (IJERT) Vol. 1 Issue 6, August - 2012. ISSN: 2278-0181. Access Date:09/07/2015.

9. Mandeep Kaur, Sanjeev Kumar, “A RECOGNITION SYSTEM FOR HANDWRITTEN GURMUKHI CHARACTERS”International Journal of Engineering Research and Technology (IJERT) Vol. 1 Issue 6, August - 2012 ISSN: 2278-0181. Access Date:09/07/2015.

10. Miroslav NOHAJ, Rudolf JAKA, “Image preprocessing for optical character recognition using neural networks” Journal of Patter Recognition Research, 2011. Access Date:09/07/2015.

2.3 PROBLEM STATEMENT DEFINITION:

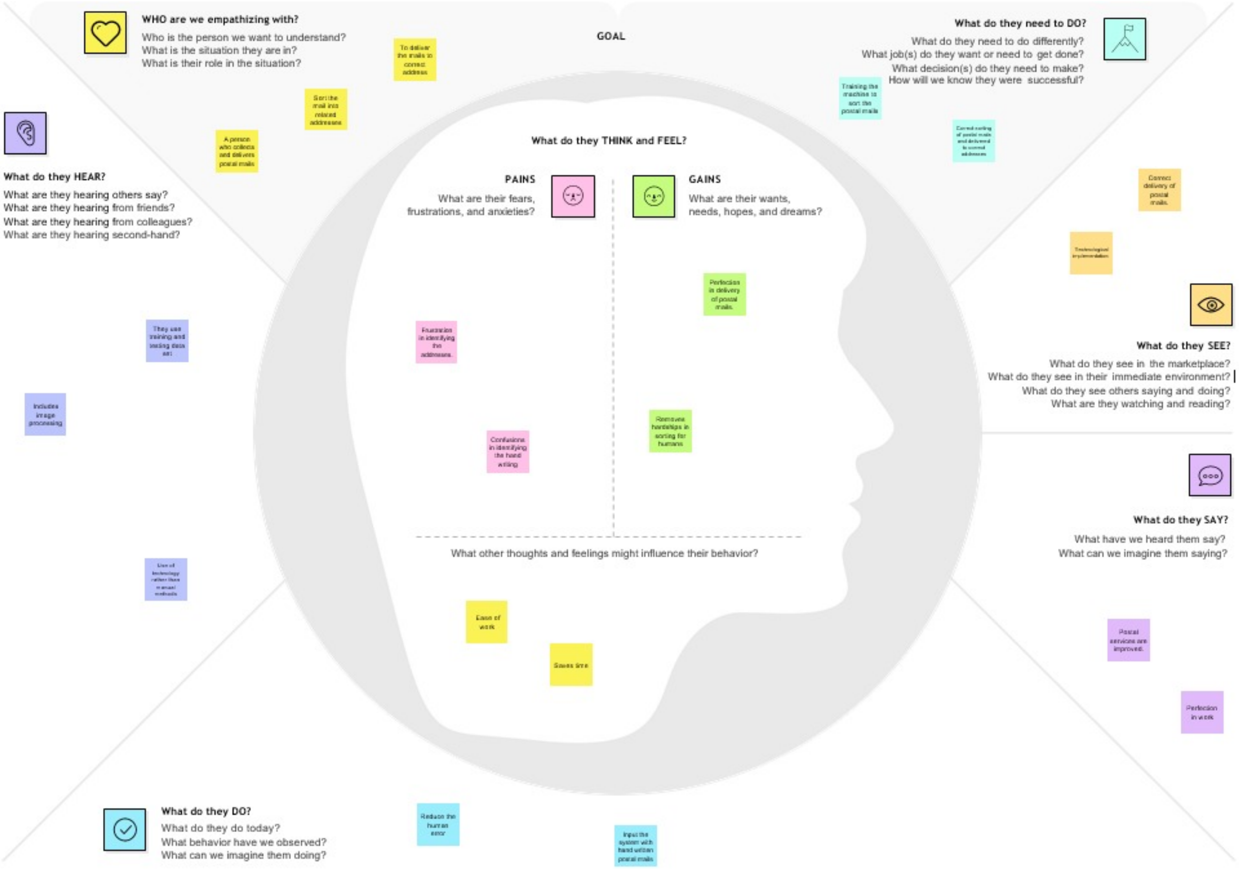
- I am a person who collects and delivers postal mails. I’m trying to sort the mail into related addresses but manual methods leads to mistakes in sorting because of handwritten post codes and addresses which make me feel guilty of wrong delivery of postal mail.
- I am a person who wants to know the prescribed medicines. I’m trying to know the medicine names suggested by the doctor but I am unable to read and understand because of handwritten doctor’s prescription which makes me feel frustrated.

2.3.1 Problem statement

Problem Statement (PS)	**Iam (Customer)**	**I'm trying to**	**But**	**Because**	**Which makes me feel**
PS-1	A person who collects and delivers postal mails	Sort the mail into related addresses	Manual methods leads to mistakes in sorting	Of handwritten post codes and addresses	Guilty of wrong delivery of postal mail.
PS-2	A person who wants to know the prescribed medicines	Know the medicine names suggested by the doctor	I am unable to read and understand	Of handwritten doctor's prescription	Frustrated

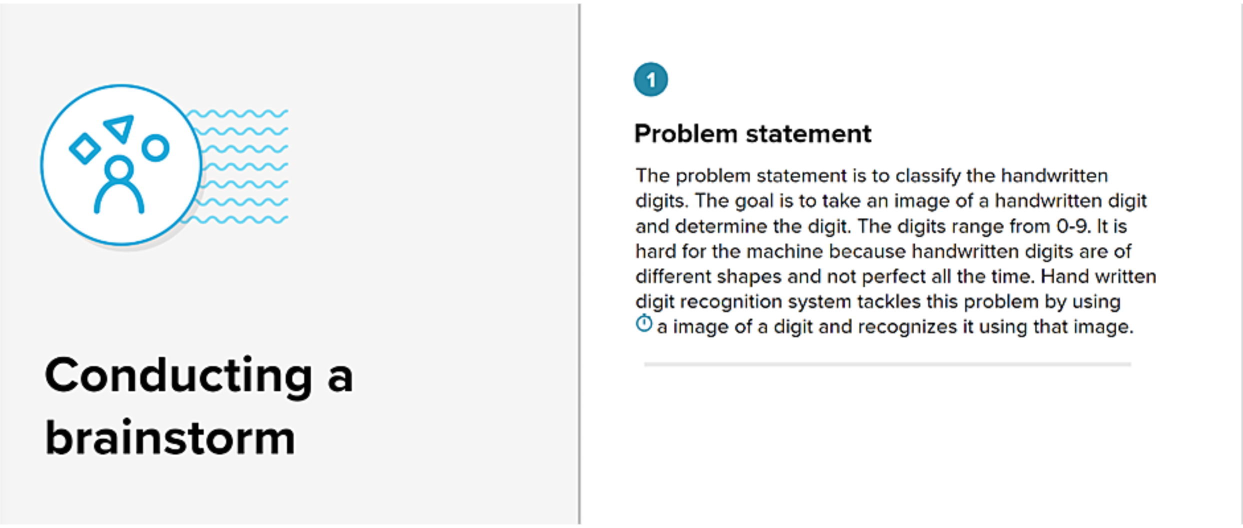
3.IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS:


WHO are we empathizing with?
Who is the person we want to understand?
What is the situation they are in?
What is their role in the situation?
GOAL
What do they need to DO?
What do they need to do differently?
What job(s) do they want or need to get done?
What decision(s) do they need to make?
How will we know they were successful?
To deliver the mails to correct address
Sort the mail into related addresses
A person who collects and delivers postal mails
Training the machine to sort the postal mails
Correct sorting of postal mails and delivered to correct addresses
What do they THINK and FEEL?
PAINS
What are their fears, frustrations, and anxieties?
GAINS
What are their wants, needs, hopes, and dreams?
Correct delivery of postal mails.
Technological implementation
What do they HEAR?
What are they hearing others say?
What are they hearing from friends?
What are they hearing from colleagues?
What are they hearing second-hand?
Perfection in delivery of postal mails.
Frustration in identifying the addresses.
They use training and testing data set
Includes image processing
Confusions in identifying the hand writing
Removes hardships in sorting for humans
What do they SEE?
What do they see in the marketplace?
What do they see in their immediate environment?
What do they see others saying and doing?
What are they watching and reading?
What do they SAY?
What have we heard them say?
What can we imagine them saying?
What other thoughts and feelings might influence their behavior?
Use of technology rather than manual methods
Ease of work
Saves time
Postal services are improved.
Perfection in work
What do they DO?
What do they do today?
What behavior have we observed?
What can we imagine them doing?
Reduce the human error
Input the system with hand written postal mails

3.2 IDEATION & BRAINSTORMING:

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



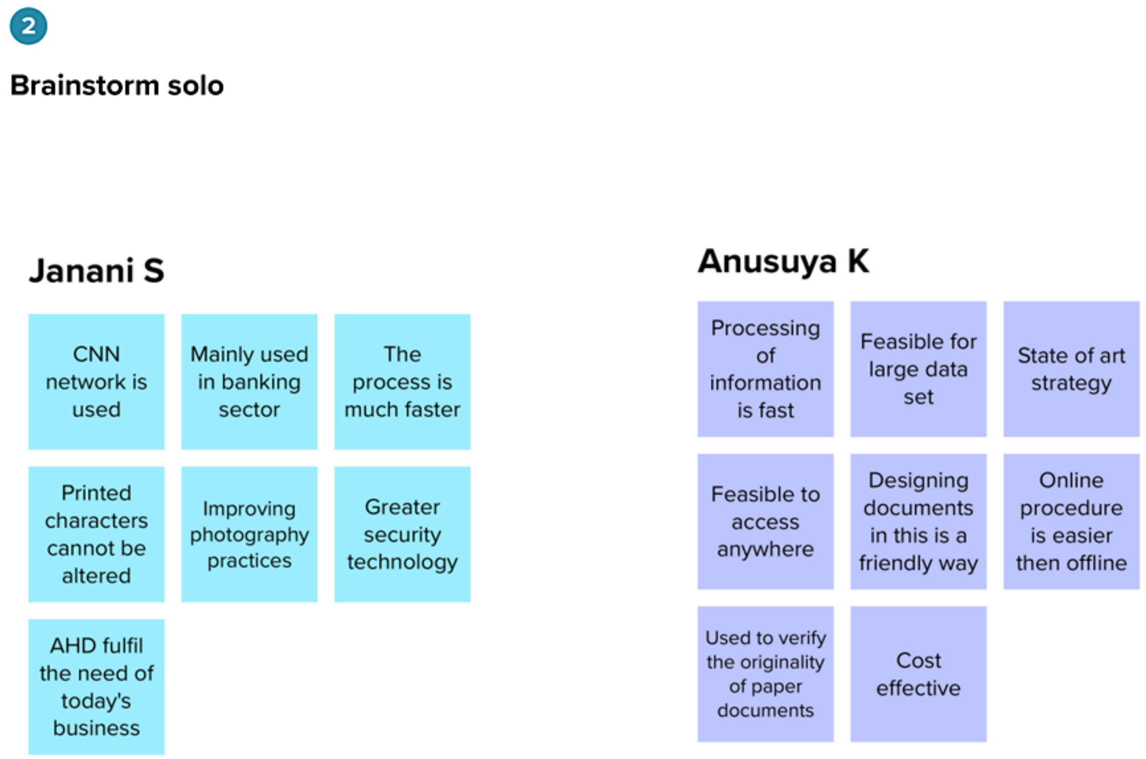
Conducting a brainstorm

1

Problem statement

The problem statement is to classify the handwritten digits. The goal is to take an image of a handwritten digit and determine the digit. The digits range from 0-9. It is hard for the machine because handwritten digits are of different shapes and not perfect all the time. Hand written digit recognition system tackles this problem by using a image of a digit and recognizes it using that image.

Step-2: Brainstorm, Idea Listing and Grouping



2

Brainstorm solo

Janani S

- CNN network is used
- Mainly used in banking sector
- The process is much faster
- Printed characters cannot be altered
- Improving photography practices
- Greater security technology
- AHD fulfil the need of today's business

Anusuya K

- Processing of information is fast
- Feasible for large data set
- State of art strategy
- Feasible to access anywhere
- Designing documents in this is a friendly way
- Online procedure is easier then offline
- Used to verify the originality of paper documents
- Cost effective

Savitha Sree L

Affect training time

Lack of recognition accuracy

Complexity of noise in data

Variation in character styles

Stress on some parts of numbers

Pattern analysis is complex

Heavy tailed distributions

Hand writing must be dark enough

Huge ambiguity of strokes from person to person

Gopinath R

Need to develop an efficient algorithm

There is an probability of potential of collapse

Limited no.of characters

Expensive method of data entry

Poor quality of source document

3

Brainstorm as a group

15 minutes

It involved Machine learning techniques like HMM,SVM etc.

Feature extraction steps varies for every language hence it is not scalable

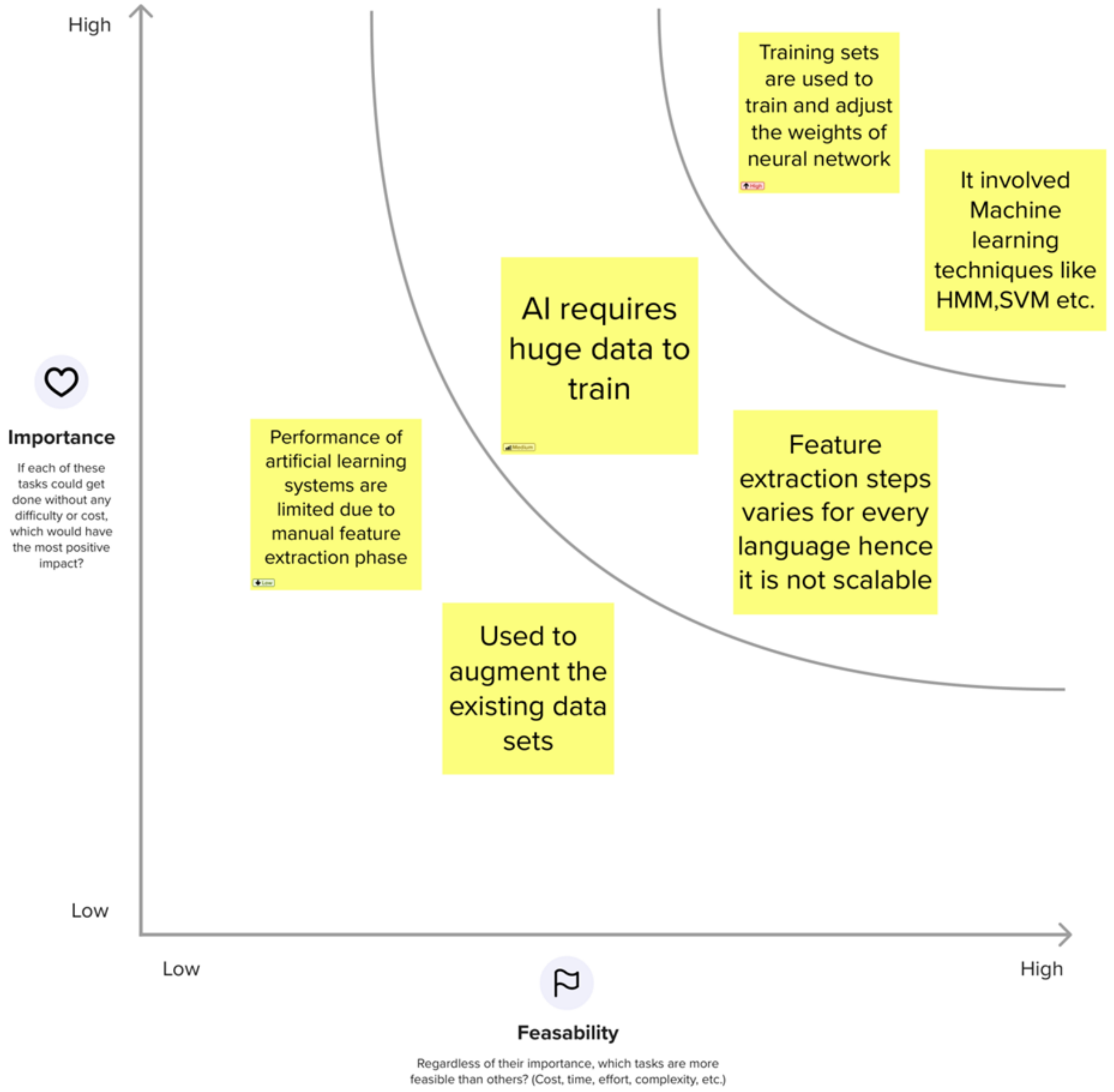
Used to augment the existing data sets

Training sets are used to train and adjust the weights of neural network

Performance of artificial learning systems are limited due to manual feature extraction phase

AI requires huge data to train

Step-3: Idea Prioritization



If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?

Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

Training sets are used to train and adjust the weights of neural network

It involved Machine learning techniques like HMM,SVM etc.

AI requires huge data to train

Feature extraction steps varies for every language hence it is not scalable

Performance of artificial learning systems are limited due to manual feature extraction phase

Used to augment the existing data sets

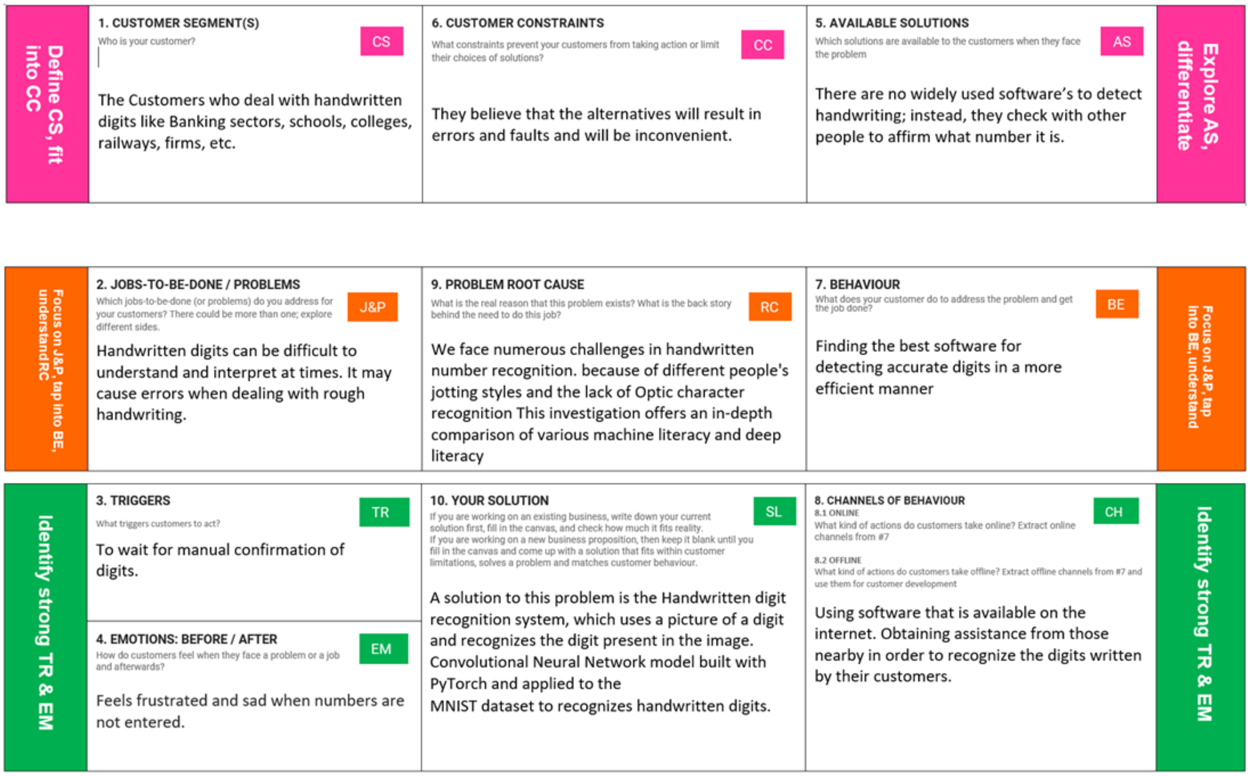
3.3 PROPOSED SOLUTION:

3.3.1 Proposed solution

S.No	Parameter	Description
1.	Problem Statement (Problem to be solved)	Janani who is a cashier needs a way to a quickly enter the account details which are written by account holders so that account holders don't have to wait long
2.	Idea / Solution description	Using MNIST dataset and Convolutional Neural Network to Perform digits recognition. All of those digits can be converted into electronic words in a text document format, and this data only needs a fraction of the physical storage space of the physical copies.
3.	Novelty / Uniqueness	Unlike OCR which recognise all the character, it can accurately recognise the digits.
4.	Social Impact / Customer Satisfaction	It saves time and work load in sectors which uses this technology.

5.	Business Model (Revenue Model)	In banking sector, numerical detail in cheque can be easily recognised. Pin code details are obtainable in Postal System
6.	Scalability of the Solution	Able to distinguish numbers in noisy environment. No restriction on number of digits to be recognised

3.4 PROBLEM SOLUTION FIT:



Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Who is your customer?

The Customers who deal with handwritten digits like Banking sectors, schools, colleges, railways, firms, etc.

6. CUSTOMER CONSTRAINTS CC

What constraints prevent your customers from taking action or limit their choices of solutions?

They believe that the alternatives will result in errors and faults and will be inconvenient.

5. AVAILABLE SOLUTIONS AS

Which solutions are available to the customers when they face the problem

There are no widely used software's to detect handwriting; instead, they check with other people to affirm what number it is.

Explore AS, differentiate

Focus on J&P, tap into BE, understandRC

2. JOBS-TO-BE-DONE / PROBLEMS J&P

Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides.

Handwritten digits can be difficult to understand and interpret at times. It may cause errors when dealing with rough handwriting.

9. PROBLEM ROOT CAUSE RC

What is the real reason that this problem exists? What is the back story behind the need to do this job?

We face numerous challenges in handwritten number recognition. because of different people's jotting styles and the lack of Optic character recognition This investigation offers an in-depth comparison of various machine literacy and deep literacy

7. BEHAVIOUR BE

What does your customer do to address the problem and get the job done?

Finding the best software for detecting accurate digits in a more efficient manner

Focus on J&P, tap into BE, understand

Identify strong TR & EM

3. TRIGGERS TR

What triggers customers to act?

To wait for manual confirmation of digits.

4. EMOTIONS: BEFORE / AFTER EM

How do customers feel when they face a problem or a job and afterwards?

Feels frustrated and sad when numbers are not entered.

10. YOUR SOLUTION SL

If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality.
If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.

A solution to this problem is the Handwritten digit recognition system, which uses a picture of a digit and recognizes the digit present in the image. Convolutional Neural Network model built with PyTorch and applied to the MNIST dataset to recognizes handwritten digits.

8. CHANNELS OF BEHAVIOUR CH

8.1 ONLINE
What kind of actions do customers take online? Extract online channels from #7

8.2 OFFLINE
What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development

Using software that is available on the internet. Obtaining assistance from those nearby in order to recognize the digits written by their customers.

Identify strong TR & EM

4.REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT:

FR No.	**Functional Requirement (Epic)**	**Sub Requirement (Story / Sub-Task)**
FR-1	User Registration	Registration through Gmail
FR-2	User Confirmation	Confirmation via Email
FR-3	Uploading the image	Uploading the handwritten digit image in the format provided
FR-4	Using a web browser	User requires a desktop or mobile browser
FR-5	Image Data	Handwritten digit recognition is the ability of a computer to recognize the human and written digits from different sources like images, papers, touch screens, etc, and classify them into 10 predefined classes This has been a topic of boundless research in the field of deep learning. In the realm of deep learning, this has been the subject of countless studies

FR-6	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-7	MNIST dataset	The MNIST dataset is an acronym that stands for the Modified National Institute of Standards and Technology 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.
FR-8	Cloud Computing	Cloud Computing is defined as a virtual platform that allows you to store and access your data over the internet without any limitations.

4.2 NON FUNCTIONAL REQUIREMENTS:

FR No.	**Non-Functional Requirement**	**Description**

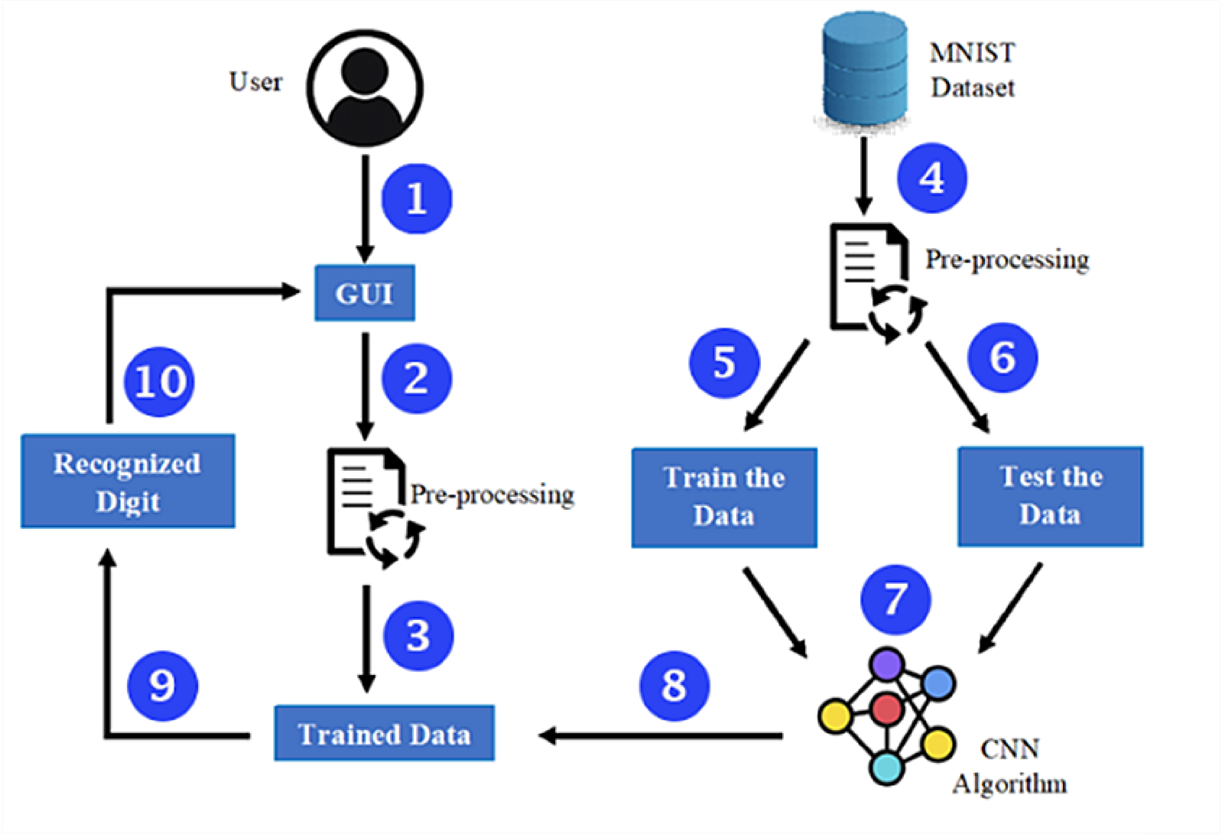
NFR-1	**Usability**	Handwritten character recognition is one of the practically important issues in pattern recognition applications. The applications of digit recognition include postal mail sorting, bank check processing, form data entry, etc. One of the very significant problems in pattern recognition applications is the recognition of handwritten characters. Applications for digit recognition include filling out forms, processing bank checks, and sorting mail.
NFR-2	**Security**	All the data provided by user are secured.
NFR-3	**Reliability**	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 and also this process is 99% accurate

NFR-4	**Performance**	The neural network uses the examples to automatically infer rules for recognizing handwritten digits. Furthermore, by increasing the number of training examples, the network can learn more about handwriting, and so improve its accuracy. There are a number of ways and algorithms to recognize handwritten digits, including Deep Learning/CNN, SVM, Gaussian Naive Bayes, KNN, Decision Trees, Random Forests, etc.
NFR-5	**Availability**	Since we use cloud, the availability of this software is all over the world only Internet facility is needed.
NFR-6	**Scalability**	This process has lot of future technology, and the usage id abundant.

5.PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS:

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. Solution & Technical Architecture User Stories.



Simplified DFD:

1. Get image from user.

2. Pre-processing the given image by specifying the pixel to be

3. Pre-processing image is compared with trained data.

4. MNIST dataset sent to pre-processing.

5. Pre-processed data classified to training data.

6. Pre-processed data classified to testing data.

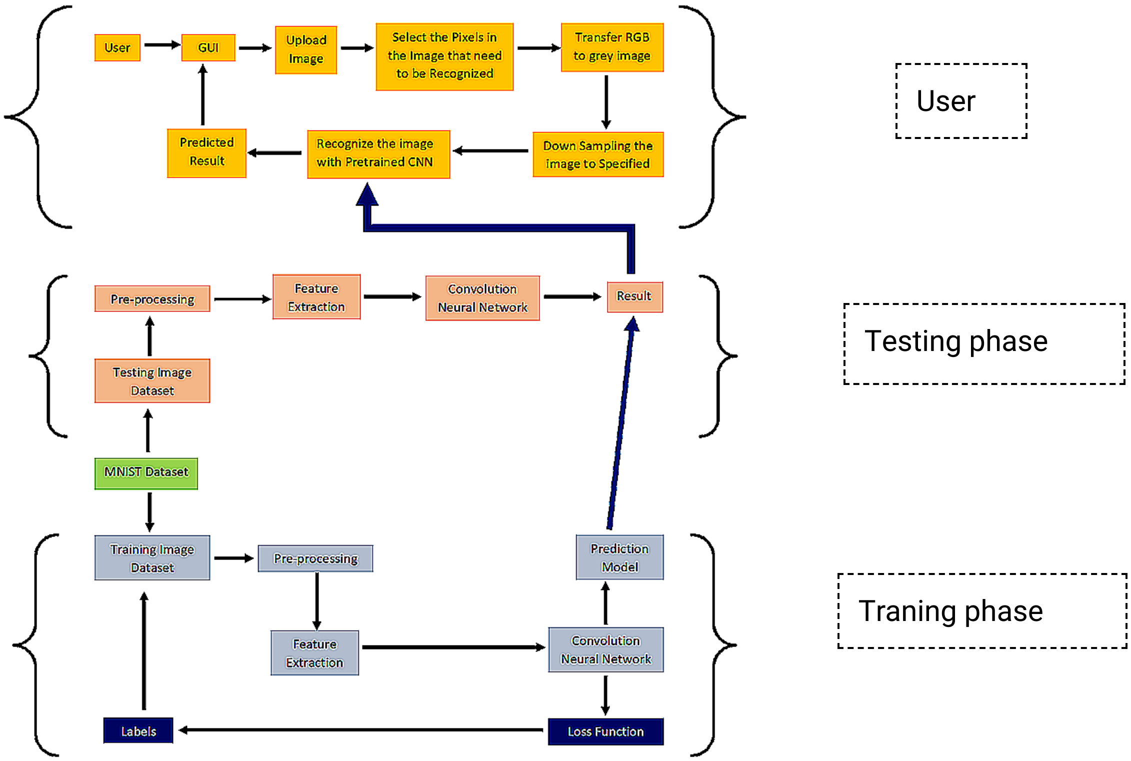
7. Both Training and Testing data are sent to CNN Algorithm.

8. Trained data is obtained from the CNN Model.

9. Image from user is recognized using trained data.

10.Recognized data is sent to the user.

Data Flow Diagram:

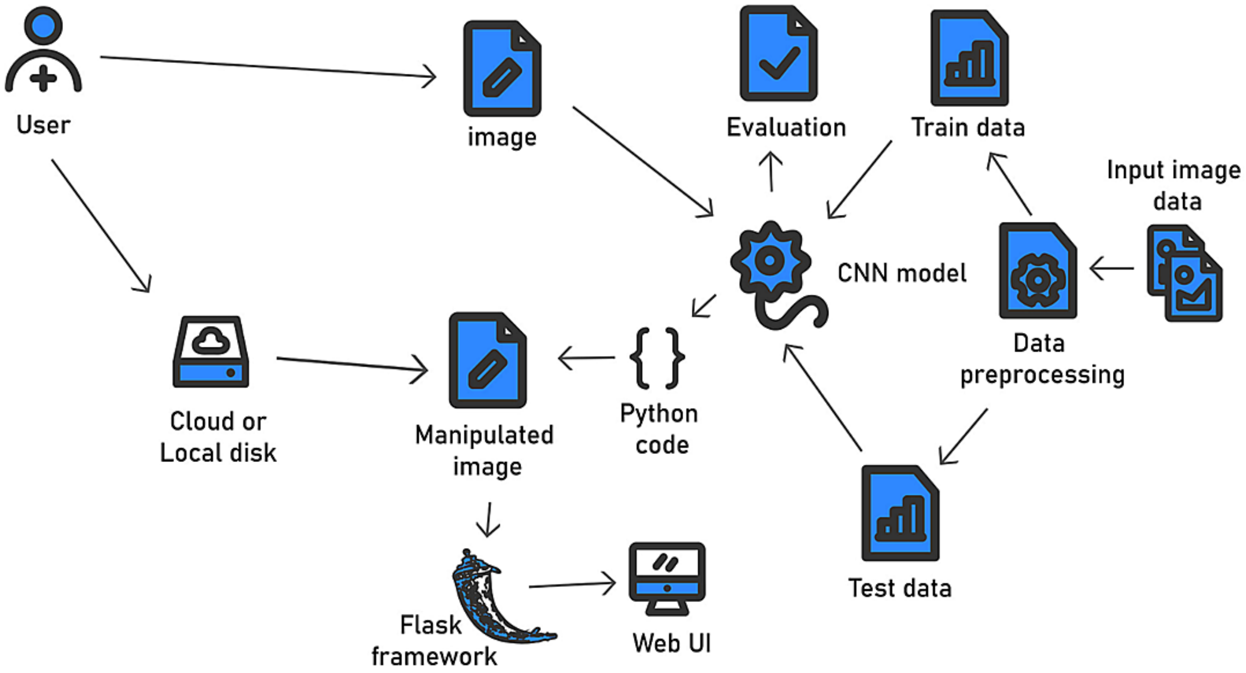


5.2 Solution & Technical Architecture

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

- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behaviour, and other aspects of the software to project stakeholders.
- Define features, development phases, and solution requirements.

Provide specifications according to which the solution is defined, managed, and delivered.



5.3 User stories:

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Cashier in Bank Sector)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	Medium	Sprint-2
	Uploading Image	USN-2	As a user, I can upload the images.	Uploading images (.jpg, .png, etc.,)	High	Sprint-3
	Selecting the part to be recognized	USN-3	User will select the part that need to be recognized.	By Scrolling of mouse	Low	Sprint-3
	Image Processing	USN-4	The background is eliminated from captured image and converted into binary image.	Accurate prediction is done only on the binary image.	Medium	Sprint-3
	Prediction	USN-5	The binary image is predicted as Numerical Digits (1,2,3)., by using CNN model.	Accurate prediction is done using CNN model	High	Sprint-1
	Perform Action	USN-6	Performing action by training the MNIST Dataset.		High	Sprint-1
	Viewing Result	USN-7	Viewing the Recognized digital data of digits.	The resultant Digit is displayed in the web UI.	High	Sprint-4

6.PROJECT PLANNING & SCHEDULING

6.1Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story/ Task	Story Points	Priority	Team Members
Sprint-1	Data Collection	USN-1	Download the Dataset	10	High	Gopinath R , Anusuya K
Sprint-1		USN-2	Image Pre-processing	10	High	Savitha Sree L , Janani S
Sprint-1		USN-3	Import and Configure the Image Data Generator Libraryand Class	10	High	Gopinath R, Janani S, Savitha Sree L, Anusuya K
Sprint-1		USN-4	Apply Image Data Generator Functionality to Train-Set and Test-Set	10	High	Gopinath R, Savith Sree L,
Sprint-2	Model Building	USN-5	Import the Model Building Libraries and Initializing the Model	10	High	Anusuya K, Gopinath R, Savitha Sree L, Janani S
Sprint-2		USN-6	Adding CNN Layers and Dense Layers	10	High	Anusuya K, Janani S, Savitha Sree L, Gopinath R

Sprint	Functional Requirement (Epic)	User Story Number	User Story/ Task	Story Points	Priority	Team Members
Sprint-2		USN-7	Configure the Learning Process	10	High	Anusuya K, Savitha Sree L
Sprint-2		USN-8	Train the Model, Save the Model andTest the Model	10	High	Janani S, Gopinath R, Savitha Sree L, Anusuya K
Sprin-2		USN-9	Image processing of given image	10	High	Gopinath R, Savitha Sree L, Janani S, Anusuya K
Sprint-3	Application Building	USN-10	Create Web Application using HTML, CSS, JavaScript	10	Medium	Gopinath R, Savitha Sree L, Janani S
Sprint-3		USN-11	Build Python code	10	High	Gopinath R, Savitha Sree L, Janani S
Sprint-3		USN-12	Run the Application	10	High	Savitha Sree L, Anusuya K, Gopinath R
Sprint-4	Train The Model on IBM	USN-13	Register for IBM Cloud	10	High	Gopinath R, Anusuya K, Janani S

Sprint-4		USN-14	Train the Model and Test the Model and its Overall Performance	10	High	Savitha Sree L, Anusuya K, Janani S

6.2 Sprint Delivery Schedule

SI.No	Milestone	Activities	Team members
1	Data collection	Download the Dataset	Gopinath R Janani S
2	Data collection	Image Pre-processing	SavithaSree L, Anusuya K
3	Data collection	Import the Image Data Generator Library	Gopinath R Janani S,SavithaSree L, Anusuya K
4	Data collection	Configure Image Data Generator Class	Janani S,SavithaSree L, Anusuya K
5	Data collection	Apply Image Data Generator Functionality to Trainset and Test set	Gopinath R Janani S,SavithaSree L, Anusuya K
6	Model Building	Import the Model Building Libraries	Gopinath R Janani S,
7	Model Building	Initializing the Model	SavithaSree L, Anusuya K
8	Model Building	Adding CNN Layers	Gopinath R Janani S,SavithaSree L,
9	Model Building	Adding Dense Layers	Janani S,SavithaSree L, Anusuya K

10	Model Building	Configure the Learning Process	Gopinath R Janani S,SavithaSree L, Anusuya K
11	Model Building	Train The Model	Janani S,SavithaSree L, Anusuya K
12	Model Building	Save the Model	Gopinath R Janani S,SavithaSree L, Anusuya K
13	Model Building	Test Model	Gopinath R Janani S,SavithaSree L, Anusuya K
14	Application Building	Create HTML Pages	Janani S,SavithaSree L, Anusuya K
15	Application Building	Build Python code	Gopinath R Janani S,SavithaSree L, Anusuya K
16	Application Building	Run the Application	Gopinath R Janani S,SavithaSree L, Anusuya K
17	Train The Model on IBM	Register for IBM Cloud	Janani S,SavithaSree L, Anusuya K
18	Train The Model on IBM	Train Model on IBM	Gopinath R Janani S,SavithaSree L, Anusuya K

7. CODING & SOLUTIONING

7.1 Feature 1

Along with the predicted value our model gives the probability of the other digits.

```
import os
import random
import string
from pathlib import Path
import numpy as np
from tensorflow.keras.models import load_model
from PIL import Image, ImageOps


def recognize(image):
  model=load_model(Path("./model/model.h5"))

  img = Image.open(image).convert("L")
  img_name = image.filename
  img = ImageOps.grayscale(img)
  img = ImageOps.invert(img)
  img = img.resize((28, 28))

  img2arr = np.array(img)
  img2arr = img2arr / 255.0
  img2arr = img2arr.reshape(1, 28, 28, 1)

  results  = model.predict(img2arr)
  best = np.argmax(results,axis = 1)[0]
  pred = list(map(lambda x: round(x*100, 2), results[0]))

  values = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
  others = [(i,pred[i]) for i in range(0,10)]
  best = (pred.index(max(pred)),max(pred))
  #best = others.pop(best)

  return best,others
```

7.2 Feature 2

```
# loading the MNIST dataset

mnist = tf.keras.datasets.mnist
(x_train, y_train), (x_test, y_test) = mnist.load_data()

input_shape = (28, 28, 1)

x_train=x_train.reshape(x_train.shape[0], x_train.shape[1], x_train.shape[2], 1)
x_train=x_train / 255.0
x_test = x_test.reshape(x_test.shape[0], x_test.shape[1], x_test.shape[2], 1)
x_test=x_test/255.0
```

8. TESTING

8.1 Test Cases

S.No	Feature type	Component	Test scenario	Steps to execute	Expected Result	Actual Result	Status
1	Functional	Handwritten image of 9	Check whether the model detects 9	Find the uploaded handwritten image	9	9	Pass
2	Functional	Handwritten image of 4	Check whether the model detects 4	Find the uploaded handwritten image	4	4	Pass
3	Functional	Handwritten image of 0	Check whether the model detects 9	Find the uploaded handwritten image	0	0	Pass

8.2 User Acceptance Testing

Purpose of document :

The purpose of this document is to briefly explain the test coverage and open issues of the project - Emerging Methods for Early Detection of Forest Fires at the time of the release to User Acceptance Testing (UAT).

Defect analysis :

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved.

Stage	Severity 1(High)	Severity 2	Severity 3	Severity 4(Low)	Subtotal
Design	0	1	0	2	3
Coding and solutioning	2	1	2	4	9
Executing	1	1	0	1	3
Total	3	3	2	7	15
Fixed bugs	3	3	2	7	15
Not fixed	0	0	0	0	0

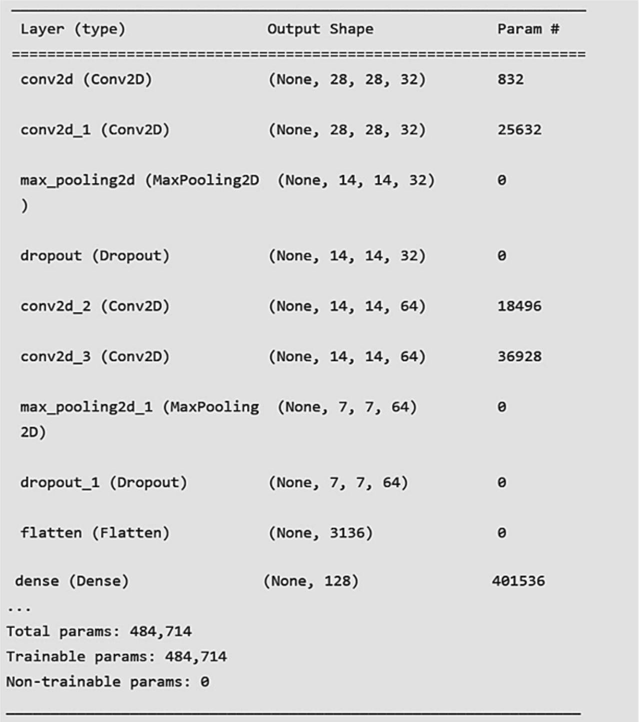
Test Case Analysis :

This report shows the number of test cases that have passed, failed, and untested

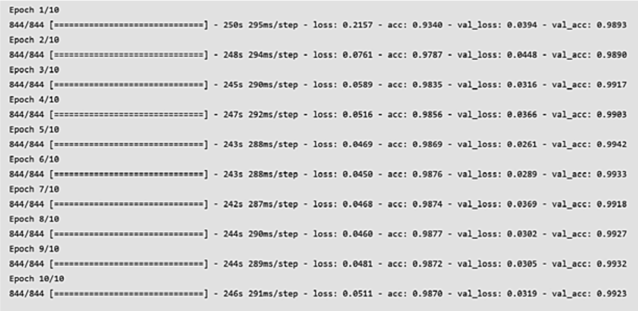
Section	Total test cases	Not tested	Fail	Pass
Pre-processing the images	1	0	0	1
Prediction	2	0	0	2

9. RESULT

9.1. Performance Metric

S.No.	Parameter	Values	Screenshot
1.	Model Summary	Total params: 484,714 Trainable params: 484,714 Non-trainable params: 0	

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 28, 28, 32)        832

 conv2d_1 (Conv2D)           (None, 28, 28, 32)        25632

 max_pooling2d (MaxPooling2D  (None, 14, 14, 32)       0
 )

 dropout (Dropout)           (None, 14, 14, 32)        0

 conv2d_2 (Conv2D)           (None, 14, 14, 64)        18496

 conv2d_3 (Conv2D)           (None, 14, 14, 64)        36928

 max_pooling2d_1 (MaxPooling  (None, 7, 7, 64)         0
 2D)

 dropout_1 (Dropout)         (None, 7, 7, 64)          0

 flatten (Flatten)           (None, 3136)              0

 dense (Dense)               (None, 128)               401536
...
Total params: 484,714
Trainable params: 484,714
Non-trainable params: 0
_________________________________________________________________
```

2.	Accuracy	Training Accuracy - 0.9870 Validation Accuracy – 0.9923	 Epoch 1/10 844/844 [==============================] - 250s 295ms/step - loss: 0.2157 - acc: 0.9340 - val_loss: 0.0394 - val_acc: 0.9893 Epoch 2/10 844/844 [==============================] - 248s 294ms/step - loss: 0.0761 - acc: 0.9787 - val_loss: 0.0448 - val_acc: 0.9890 Epoch 3/10 844/844 [==============================] - 245s 290ms/step - loss: 0.0589 - acc: 0.9835 - val_loss: 0.0316 - val_acc: 0.9917 Epoch 4/10 844/844 [==============================] - 247s 292ms/step - loss: 0.0516 - acc: 0.9856 - val_loss: 0.0366 - val_acc: 0.9903 Epoch 5/10 844/844 [==============================] - 243s 288ms/step - loss: 0.0469 - acc: 0.9869 - val_loss: 0.0261 - val_acc: 0.9942 Epoch 6/10 844/844 [==============================] - 243s 288ms/step - loss: 0.0450 - acc: 0.9876 - val_loss: 0.0289 - val_acc: 0.9933 Epoch 7/10 844/844 [==============================] - 242s 287ms/step - loss: 0.0468 - acc: 0.9874 - val_loss: 0.0369 - val_acc: 0.9918 Epoch 8/10 844/844 [==============================] - 244s 290ms/step - loss: 0.0460 - acc: 0.9877 - val_loss: 0.0302 - val_acc: 0.9927 Epoch 9/10 844/844 [==============================] - 244s 289ms/step - loss: 0.0481 - acc: 0.9872 - val_loss: 0.0305 - val_acc: 0.9932 Epoch 10/10 844/844 [==============================] - 246s 291ms/step - loss: 0.0511 - acc: 0.9870 - val_loss: 0.0319 - val_acc: 0.9923

10.ADVANTAGES & DISADVANTAGES

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 small number of parameters and hence training is relatively easy and fast

4) Unlike many other recognition schemes, it does not rely on some form of pre-normalization of input images, but can handle arbitrary scalings, translations and a limited degree of image rotation.

DISADVANTAGES:

The disadvantage is that it is not done in real time as a person writes and therefore not appropriate for immediate text input.

The main disadvantage of the method is that it requires much more computation than more standard OCR techniques.

11.CONCLUSION

In this paper, the variations of accuracies for handwritten digit were observed for 15 epochs by varying the hidden layers. The accuracy curves were generated for the six cases for the different parameter using CNN MNIST digit dataset. The six cases perform differently because of the various combinations of hidden layers. The layers were taken randomly in a periodic sequence so that each case behaves differently during the experiment. The maximum and minimum accuracies were observed for different hidden layers variation with a batch size of 100. Among all the observation, the maximum accuracy in the performance was found 99.21% for 15 epochs in case 2 (Conv1, pool1, Conv2,
pool2 with 2 dropouts). In digit recognition, this type of higher accuracy will cooperate to speed up the performance of the machine more adequately. However, the minimum accuracy among all observation in the performance was
found 97.07% in case 6 (Conv1, pool1, Conv2, pool2 with 1 dropout). Moreover, among all the cases, the total highest test loss is approximately 0.049449 found in case 3 without dropout and the total lowest test loss is approximately 0.026303 found in case 2 with dropout. This low loss will provide CNN better performance to attain better image resolution and noise processing. In the future, we plan to observe the variation in the overall classification accuracy by varying the number of hidden layers and batch

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. The advancement in this field can help us create an environment of safety, awareness and comfort by using these algorithms in day to day application and high-level application (i.e. Corporate level or Government level). Application-based on artificial intelligence and deep learning is the future of the technological world because of their absolute accuracy and advantages over many major problems

13.APPENDIX

Python:

Python is an interpreted, high-level, general purpose programming language created byGuido VanRossumand first released in 1991, Python's design philosophy emphasizescode Readability with itsnotable use of significant White space. Its language constructs and object oriented approach aim to helpprogrammers write clear, logicalcode for small and large-scale projects. Python is dynamically typedand garbagecollected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming

Keras :

Keras is a powerful and easy-to-use free open source Python library for developing and evaluating deep learning models.

It wraps the efficient numerical computation libraries Theano and TensorFlow and allows you to define and train neural network models in just a few lines of code. It uses libraries such as Python, C#, C++ or standalone machine learning toolkits. Theano and TensorFlow are very powerful libraries but difficult to understand for creating neural networks. Keras is based on minimal structure that provides a clean and easy way to create deep learning modelsbased on TensorFlow or Theano. Keras is designed to quickly define deep learning models. Well, Keras is an optimal choice for deep learning applications.

Steps for creating a keras model:

1)First we must define a network model.
2)Compile it, which transforms the simple sequence of layers into a complex group of matrix operations.
3)Train or fit the network.

To import:
from keras.models import Sequential
fromkeras.layers import Dense, Activation, Dropout

TensorFlow:

TensorFlow is a Python library for fast numerical computing created and released by Google. It is afoundationlibrary that can be used to create Deep Learning models directly or by using wrapper librariesthat simplifythe process built on top of TensorFlow. TensorFlow tutorial is designed for both beginnersand professionals.Our tutorial provides all the basic and advanced concept of machine learning anddeep learning concept such as deep neural network, image processing and sentiment analysis.

TensorFlow is one of the famous deep learning frameworks, developed by Google Team. It is a free andopen -source software library and designed in Python programming language, this tutorial is designedin such a way that we can easily implements deep learning project on TensorFlow in an easy andefficient way. Unlike other numerical libraries intended for use in Deep Learning like Theano,TensorFlow was designed for use both in research and development and in production systems. It canrun on single CPU systems, GPUs as well as mobile devices and largescale distributed systems ofhundreds of machines.

Numpy:

NumPy is a Python library used for working with arrays. It also has functions for working in domain oflinear algebra, Fourier transform, and matrices. Numpy which stands for Numerical Python, is a libraryconsisting of multidimensional array objects and a collection of routines for processing those arrays.Using NumPy, mathematical and logical operations on arrays can be performed. This tutorial explainsthe basics of NumPy such as its architecture and environment. It also discusses the various arrayfunctions, types of indexing, etc. It is an open source project and you can use it freely.

NumPy stands for Numerical Python. NumPyaims to provide an array object that is up to 50x faster than traditional Python lists. The array object inNumPy is called ndarray, it provides a lot of supporting functions that make workingwith ndarray very easy. Arrays are very frequently used in data science, where speed and resources arevery important.

13.1 Source Code

Train the Model On IBM:

```
# importing Required Libraries

import tensorflow as tf
import seaborn as sns
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
from keras.utils.vis_utils import plot_model

# loading the MNIST dataset

mnist = tf.keras.datasets.mnist
(x_train, y_train), (x_test, y_test) = mnist.load_data()

sns.countplot(y_train)

input_shape = (28, 28, 1)

x_train=x_train.reshape(x_train.shape[0], x_train.shape[1], x_train.shape[2], 1)
x_train=x_train / 255.0
x_test = x_test.reshape(x_test.shape[0], x_test.shape[1], x_test.shape[2], 1)
x_test=x_test/255.0

y_train = tf.one_hot(y_train.astype(np.int32), depth=10)
y_test = tf.one_hot(y_test.astype(np.int32), depth=10)

plt.imshow(x_train[100][:,:,0])
```

```
print(y_train[100])

X_train_plot = x_train.reshape(-1, 28, 28)

def Show_example_digits(mono = 'gray'):
   fig = plt.figure(figsize = (16, 16))
   for idx in range(15):
plt.subplot(5, 5,idx+1)
plt.imshow(X_train_plot[idx], cmap = mono)
plt.title("Digit {}".format(y_train[idx]))

plt.tight_layout()
Show_example_digits()

# determine the shape of the input images
inp_shape = x_train.shape[1:]
print(inp_shape)

batch_size = 64
num_classes = 10
epochs = 10

# defining the model
model = tf.keras.models.Sequential([tf.keras.layers.Conv2D(32, (5,5), padding='same',
activation='relu', input_shape=input_shape),
   tf.keras.layers.Conv2D(32, (5,5), padding='same', activation='relu'),
   tf.keras.layers.MaxPool2D(),
tf.keras.layers.Dropout(0.25),
   tf.keras.layers.Conv2D(64, (3,3), padding='same', activation='relu'),
   tf.keras.layers.Conv2D(64, (3,3), padding='same', activation='relu'),
   tf.keras.layers.MaxPool2D(strides=(2,2)),
tf.keras.layers.Dropout(0.25),
tf.keras.layers.Flatten(),
tf.keras.layers.Dense(128, activation='relu'),
tf.keras.layers.Dropout(0.5),
tf.keras.layers.Dense(num_classes, activation='softmax')
])

model.compile(optimizer=tf.keras.optimizers.RMSprop(epsilon=1e-08),
loss='categorical_crossentropy', metrics=['acc'])

# text Description of model
model.summary()

history = model.fit(x_train, y_train,
batch_size=batch_size,
                epochs=epochs,
validation_split=0.1)

# ploting the learning curves
fig, ax = plt.subplots(1,1)
ax.plot(history.history['loss'], color='b', label="Training Loss")
ax.plot(history.history['val_loss'], color='r', label="Validation Loss")
legend = ax.legend(loc='best', shadow=True)

# evaluate the model
loss, accuracy = model.evaluate(x_test, y_test, verbose=0)
print(f'Accuracy: {accuracy*100}')

y_pred = model.predict(x_test)

def draw_output(idx_nums):
plt.figure(figsize = (20, 20))
plt.xticks( range(10) )
    x = np.ceil(np.sqrt(len(idx_nums)))
cnt = 1
    for ph in idx_nums:
plt.subplot(x, x, cnt)
curr_photo = y_test[ph]

plt.xlim(0, 10)
plt.title("Digit: {0}\n idx: {1} ".format(np.argmax(y_test[ph]), ph), fontsize = 10)
plt.bar(range(10), y_pred[ph])

cnt += 1

cnt_error = []
for idx, (a, b) in enumerate(zip(y_test, y_pred)):
    if np.argmax(a) == np.argmax(b): continue
cnt_error.append( (np.argmax(a)) )

cnt_error = np.unique(cnt_error, return_counts = True)
sns.set_style("darkgrid")
plt.figure(figsize = (15, 7))
bar_plot = sns.barplot(cnt_error[0], cnt_error[1], palette="muted")
plt.show()

cnt_ind = 1
list_idx = []
X_val_plot = x_test.reshape( x_test.shape[:-1] )
```

```
fig = plt.figure(figsize=(14, 14))

for idx, (a, b) in enumerate(zip(y_test, y_pred)):
    if np.argmax(a) == np.argmax(b): continue
    if (np.argmax(a) == 2 or np.argmax(a) == 9):
plt.subplot(5, 5, cnt_ind)
plt.imshow(X_val_plot[idx], cmap='gray', interpolation='none')
plt.title('y_true={0}\ny_pred={1}\n ind = {2}'.format(np.argmax(a), np.argmax(b), idx))
plt.tight_layout()
list_idx.append(idx)
cnt_ind += 1

image = x_train[0]
# lets display the image which we want to predict
plt.imshow(np.squeeze(image), cmap='gray')

image.shape[0],image.shape[1],image.shape[2]

# make a prediction
# reshaping the image for model input
image= image.reshape(1,input_shape[0],input_shape[1],input_shape[2])
# predicting the label of image
yhat = model.predict([image])
print('Predicted: {}'.format(np.argmax(yhat)))

# Predict the values from the testing dataset
Y_pred = model.predict(x_test)
# Convert predictions classes to one hot vectors
Y_pred_classes = np.argmax(Y_pred,axis = 1)
# Convert testing observations to one hot vectors
Y_true = np.argmax(y_test,axis = 1)
# compute the confusion matrix
confusion_mtx = tf.math.confusion_matrix(Y_true, Y_pred_classes)

plt.figure(figsize=(10, 8))
sns.heatmap(confusion_mtx, annot=True, fmt='g')

model.save('hand_written_digits_CNN.h5')

model = tf.keras.models.load_model('hand_written_digits_CNN.h5')

image = x_test[100]
# lets display the image which we want to predict
plt.imshow(np.squeeze(image), cmap='gray')
```

```
# make a prediction
# reshaping the image for model input
image= image.reshape(1,input_shape[0],input_shape[1],input_shape[2])
# predicting the label of image
yhat = model.predict([image])
print('Predicted: {}'.format(np.argmax(yhat)))

!tar -zcvf model.tgz hand_written_digits_CNN.h5

!pip install watson-machine-learning-client

!pip install ibm_watson_machine_learning

from ibm_watson_machine_learning import APIClient
wml_credentials = {
   "url":"https://jp-tok.ml.cloud.ibm.com",
   "apikey":"Y0hOkxEIr9-Qwjc7rRJqcboPqPn2GdjCddwHsedqsc8N"
}
client = APIClient(wml_credentials)
client

client.spaces.get_details()

space_id = 'f5b3e32f-adf0-45f4-918c-88ac9ee62536'

client.set.default_space(space_id)

client.software_specifications.list()

software_space_uid=client.software_specifications.get_uid_by_name('tensorflow_rt22.1-py3.9')
software_space_uid

model_details = client.repository.store_model(model='model.tgz',meta_props={
         client.repository.ModelMetaNames.NAME : 'A Novel Method for Handwritten Digit
Recognition System',
   client.repository.ModelMetaNames.TYPE:'tensorflow_2.7',
   client.repository.ModelMetaNames.SOFTWARE_SPEC_UID:software_space_uid
})
model_details

model_id = client.repository.get_model_id(model_details)
model_id

client.repository.download(model_id,'model.tar.gb')
```

app.py

```
from flask import Flask,render_template,request
from PIL import Image, ImageOps
import os
import random
import string
from pathlib import Path
import numpy as np


def random_name_generator(n):
  return ''.join(random.choices(string.ascii_uppercase + string.digits, k=n))

app=Flask(__name__)



@app.route('/')
def home_page():
    return render_template('index.html')



@app.route('/index')
def ai_engine_page():
    return render_template('index.html')

@app.route('/submit',methods=['POST'])
def submit():
    if request.method=='POST':
      image = request.files['image']
img_name = image.filename
file_path = os.path.join('static/data/', img_name)
image.save(file_path)
img = Image.open(image).convert("L")
img = img.resize((255, 255))
img.save(os.path.join('static/thumb/', "255X255_"+img_name))
      best = (9, 73.19)
           # others = [(0, 9.15),(1, 0.35000000000000003), (2, 0.4), (3, 0.0), (4, 109.9), (5,
4.1499999999999995), (6, 3.5), (7, 3.4000000000000004), (8, 3.15), (9, 365.95)]
       others = [(0, 1.83), (1, 0.07), (2, 0.08), (3, 0.0), (4, 21.98), (5, 0.83), (6, 0.7), (7, 0.68), (8,
0.63),(9, 73.19)]
```

```
                return render_template("submit.html", best=best, others=others,
img_name=img_name)


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


if __name__ == '__main__':
app.run(debug=True)
```

CNN.py

```
import os
import random
import string
from pathlib import Path
import numpy as np
from tensorflow.keras.models import load_model
from PIL import Image, ImageOps

def recognize(image):
  model=load_model(Path("./model/model.h5"))

img = Image.open(image).convert("L")
img_name = image.filename
  # img_name = random_name_generator(10) + '.jpg'

  # if not os.path.exists(f".static/data/"):
  #   os.mkdir(os.path.join('./static', 'data'))
  #img.save(Path(f".static/data/{img_name}"))

img = ImageOps.grayscale(img)
img = ImageOps.invert(img)
img = img.resize((28, 28))

  img2arr = np.array(img)
  img2arr = img2arr / 255.0
  img2arr = img2arr.reshape(1, 28, 28, 1)
```

```
results  = model.predict(img2arr)
best = np.argmax(results,axis = 1)[0]

pred = list(map(lambda x: round(x*100, 2), results[0]))

values = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
others = [(i,pred[i]) for i in range(0,10)]
best = (pred.index(max(pred)),max(pred))
#best = others.pop(best)

return best,others
```

13.2 GitHub & Project Demo Link

GitHub:

https://github.com/IBM-EPBL/IBM-Project-7958-1658903961

Project Demo Link:
https://drive.google.com/file/d/1KUmheC21st3aZ28X5vNHztArUx73X-zW/view?usp=drivesdk