A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

IBM-Project-1006-1658334398

A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

NALAIYA THIRAN PROJECT BASED LEARNING ON PROFESSIONAL READLINESS FOR INNOVATION, EMPLOYNMENT AND ENTERPRENEURSHIP

A PROJECT REPORT

AKILA K (710019104003)
KEERTHIKA S (710019104019)
NIVETHA S (710019104027)
SWETHA S (710019104302)

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

ANNA UNIVERSITY REGIONAL CAMPUS, COIMBATORE.

COIMBATORE – 641 046.

INDEX

1. INTRODUCTION
1.1 Project Overview

Handwritten digit recognition is one of the most important issues that should be addressed in the field of machine learning and it plays a very vital role in learning about applications that implement pattern recognition as it is a very practical approach. There are many different handwritten digit recognition applications like cheque processing in banks, data entry in forms, sorting mails, etc.

To meet the ever growing daily demands related to the IT industry, machine learning is one of the most impactful concepts to be emphasized upon. To understand machine learning more effectively and implement it in problem solving, we undertook this problem of building a handwritten digit recognition system using machine learning. This problem will allow us to learn machine learning techniques from the ground up and it also has various use cases that may prove to be beneficial for individuals or organizations. Different types of learning models are included within machine learning. These are as follows:

A. Supervised Learning

Supervised learning comprises various algorithms such as Naive Bayes, K-Nearest Neighbors, Random Forest, Decision Trees, Linear Regression, Support Vector Machine.

In supervised learning approach, during the training process a model is trained with the help of a labeled dataset in which two variables namely ‘input’ and ‘output’ are mapped to each other.

B. Unsupervised Learning

Unsupervised learning comprises various algorithms such as Neural Network, Anomaly Detection, K-Mean Clustering, Multivariate Analysis.

In unsupervised learning approach, during the training process the model is trained with the help of an unlabeled dataset, be it classified or non classified. Input is not mapped with the output during the learning process instead the training input dataset is categorized under classes which are then used to predict the output for testing dataset.

C. Reinforcement Learning

Reinforcement learning comprises various algorithms such as Negative, Positive, Q Learning, Markov Decision Process. In reinforcement learning, decision making is done sequentially. An intelligent agent acts upon an environment on the basis of collective rewards and tries to maximize these rewards.

1.2 Purpose

Handwritten character recognition is one of the practically important issues in pattern recognition applications. The applications of digit recognition include in postal mail sorting, bank check processing, form data entry, etc. The main problem lies within the ability on developing an efficient algorithm that can recognize hand written digits, which is submitted by users by the way of a scanner, tablet, and other digital devices. This paper presents an approach to off-line handwritten digit recognition based on different machine learning techniques. The main objective of this paper is to ensure the effectiveness and reliability of the approached recognition of handwritten digits. Several machines learning algorithms (i.e. Multilayer Perceptron, Support Vector Machine, Naïve Bayes, Bayes Net, Random Forest, J48, and Random Tree) have been used for the recognition of digits using WEKA. The experimental results showed that the highest accuracy was obtained by Multilayer Perceptron with the value of 90.37%.

2. LITERATURE SURVEY

2.1 Existing Problem

An early notable attempt in the area of character recognition research is by Grimsdale in 1959. The origin of a great deal of research work in the early sixties was based on an approach known as analysisby-synthesis method suggested by Eden in 1968. The great importance of Eden's work was that he formally proved that all handwritten characters are formed by a finite number of schematic features, a point that was implicitly included in previous works.

This notion was later used in all methods in syntactic (structural) approaches of character recognition.

1. K. Gaurav, Bhatia P. K., his paper deals with the various pre-processing techniques involved in the character recognition with different kind of images ranges from a simple handwritten form based documents and documents containing colored and complex background and varied intensities.In this, different preprocessing techniques like skew detection and correction, image enhancement techniques of contrast stretching, binarization, noise removal techniques, normalization and segmentation, morphological processing techniques are discussed.

2. Sandhya Arora, used four feature extraction techniques namely, intersection, shadow feature, chain code histogram and straight line fitting features. Shadow features are computed globally for character image while intersection features, chain code histogram features and line fitting features are computed by dividing the character image into different segments. On experimentation with a dataset of 4900 samples the overall recognition rate observed was 92.80% for Devanagari characters.

3. Brakensiek, J. Rottland, A. Kosmala, J. Rigoll, in their paper a system for off-line cursive handwriting recognition is described which is based on Hidden Markov Models (HMM) using discrete and hybrid modelling techniques. Handwriting recognition experiments using a discrete and two different hybrid approaches, which consist of a discrete and semi-continuous structures, are compared. It is found that the recognition rate performance can be improved of a hybrid modelling technique for HMMs, which depends on a neural vector quantizer (hybrid MMI), compared to discrete and hybrid HMMs, based on tired mixture structure (hybrid - TP), which may be caused by a relative small data set.

4. R. Bajaj, L. Dey, S. Chaudhari, employed three different kinds of features, namely, the density features, moment features and descriptive component features for classification of Devanagari Numerals. They proposed multi classifier connectionist architecture for increasing the recognition reliability and they obtained 89.6% accuracy for handwritten Devanagari numerals.

5. G. Pirlo and D. Impedovo in his work on , presented a new class of membership functions, which are called Fuzzymembership functions (FMFs), for zoning-based classification. These FMFs can be easily adapted to the specific characteristics of a classification problem in order to maximize classification performance. In this research, a realcoded genetic algorithm is presented to find, in a single optimization procedure, the optimal FMF, together with the optimal zoning described by Voronoi tessellation. The experimental results, which are carried out in the field of handwritten digit and character recognition, indicate that optimal FMF performs better than other membership functions based on abstract level, ranked-level, and measurement-level weighting models, which can be found in the literature.

6. Sushree Sangita Patnaik and Anup Kumar Panda May 2011, this paper proposes the implementation of particle swarm optimization (PSO) and bacterial foraging optimization (BFO) algorithms which are intended for optimal harmonic compensation by minimizing the undesirable losses occurring inside the APF itself. The efficiency and effectiveness of the implementation of two approaches are compared for two different conditions of supply. The total harmonic distortion (THD) in the source current which is a measure of APF performance is reduced drastically to nearly 1% by employing BFO. The results demonstrate that BFO outperforms the conventional and PSO based approaches by ensuring excellent functionality of APF and quick prevail over harmonics in the source current even under unbalanced supply.

7. M. Hanmandlu, O.V. Ramana Murthy have presented in their study the recognition of handwritten Hindi and English numerals by representing them in the form of exponential membership functions which serve as a fuzzy model. The recognition is carried out by modifying the exponential membership functions fitted to the fuzzy sets. These fuzzy sets are derived from features consisting of normalized distances obtained using the Box approach. The membership function is modified by two structural parameters that are estimated by optimizing the entropy subject to the attainment of membership function to unity. The overall recognition rate is found to be 95% for Hindi numerals and 98.4% for English numerals.

8. Renata F. P. Neves have proposed SVM based offline handwritten digit recognition. Authors claim that SVM outperforms the Multilayer perceptron classifier. Experiment is 12 carried out on NIST SD19 standard dataset. Advantage of MLP is that it is able to segment non-linearly separable classes. However, MLP can easily fall into a region of local minimum, where the training will stop assuming it has achieved an optimal point in the error surface. Another hindrance is defining the best network architecture to solve the problem, considering the number of layers and the number of perceptron in each hidden layer. Because of these disadvantages, a digit recognizer using the MLP structure may not produce the desired low error rate.

2.2 References

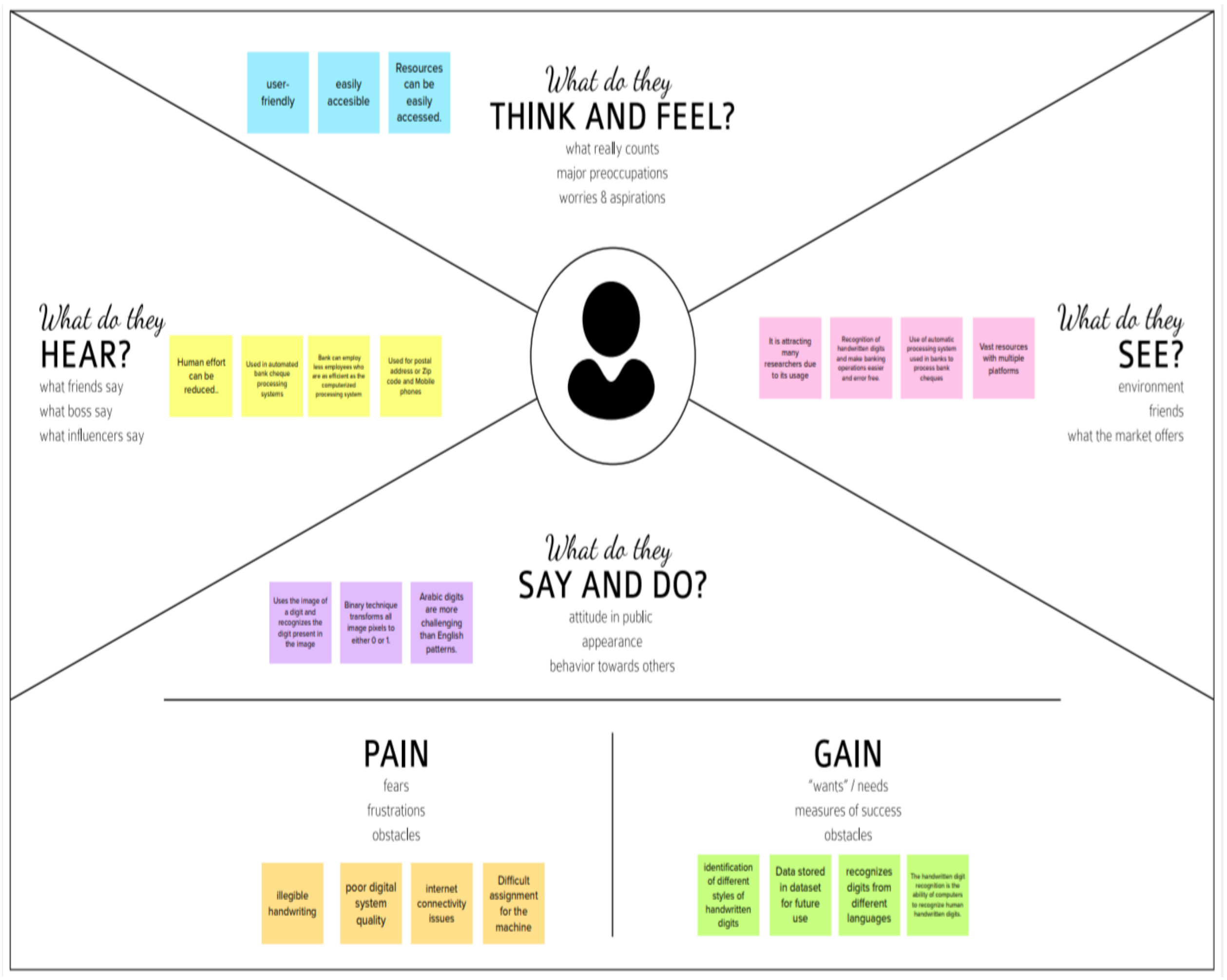
- https://www.geeksforgeeks.org/python-tkintertutorial/
- https://medium.com/the-andela-way/applyingmachine-learning-to-recognize handwrittencharacters-babcd4b8d705
- https://nanonets.com/blog/handwritten-characterrecognition/
- https://www.tensorflow.org/learn
- R. Alhajj and A. Elnagar, "Multiagents to separating handwritten connected digits," in IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, vol. 35, no. 5, pp. 593-602, Sept. 2005. https://doi.org/10.1109/TSMCA.2005.843389
- https://towardsdatascience.com/useful-plots-todiagnose-your-neural-network-521907fa2f45
- https://towardsdatascience.com/the-best-machinelearning-algorithm-for-handwritten-digits-recognition2c6089ad8f09
- https://www.datacamp.com/community/tutorials/convolutional-neural-networks-python
- https://www.analyticsvidhya.com/blog/2021/01/building-a-cnn-model-with-95-accuracy/
- https://www.analyticsvidhya.com/blog/2020/10/whatis-the-convolutional-neural-network-architecture/
- https://www.kdnuggets.com/2018/11/top-pythondeep-learning-libraries.html
- https://analyticsindiamag.com/7-types-classificationalgorithms/
- https://towardsdatascience.com/image-recognitionwith-machine-learning-on-python-image-processing3abe6b158e9a

2.3 Problem Statement Definition

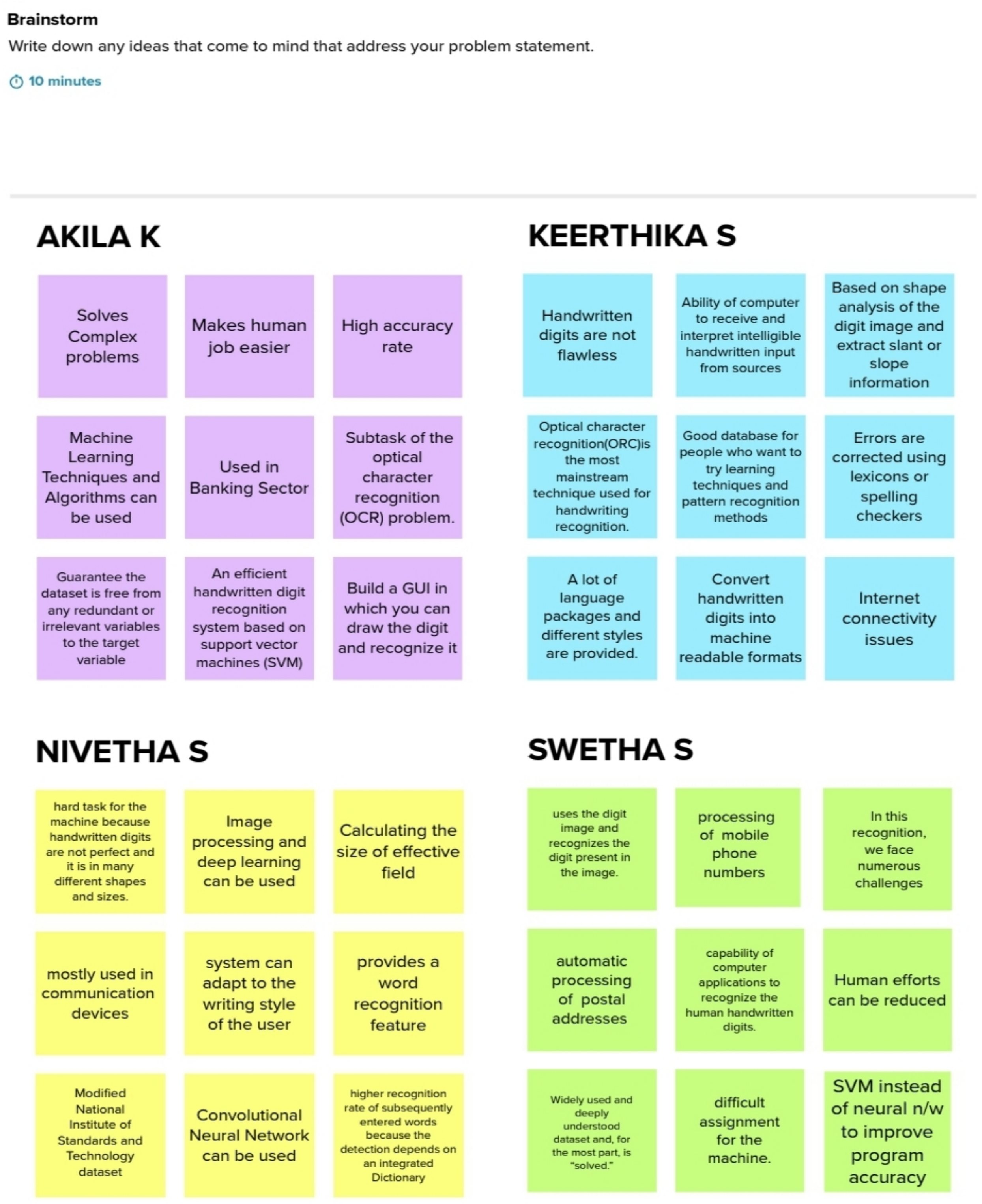
The goal of this project is to create a model that will be able to recognize and determine the handwritten digits from its image by using the concepts of Convolution Neural Network. Though the goal is to create a model which can recognize the digits, it can be extended to letters and an individual’s handwriting. The major goal of the proposed system is understanding Convolutional Neural Network, and applying it to the handwritten recognition system.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas


user-friendly
easily accesible
Resources can be easily accessed.
What do they THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations
What do they HEAR?
what friends say
what boss say
what influencers say
Human effort can be reduced..
Used in automated bank cheque processing systems
Bank can employ less employees who are as efficient as the computerized processing system
Used for postal address or Zip code and Mobile phones
It is attracting many researchers due to its usage
Recognition of handwritten digits and make banking operations easier and error free.
Use of automatic processing system used in banks to process bank cheques
Vast resources with multiple platforms
What do they SEE?
environment
friends
what the market offers
Uses the image of a digit and recognizes the digit present in the image
Binary technique transforms all image pixels to either 0 or 1.
Arabic digits are more challenging than English patterns.
What do they SAY AND DO?
attitude in public
appearance
behavior towards others
PAIN
fears
frustrations
obstacles
illegible handwriting
poor digital system quality
internet connectivity issues
Difficult assignment for the machine
GAIN
"wants" / needs
measures of success
obstacles
identification of different styles of handwritten digits
Data stored in dataset for future use
recognizes digits from different languages
The handwritten digit recognition is the ability of computers to recognize human handwritten digits.

3.2 Ideation & Brainstorming



Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

AKILA K

- Solves Complex problems
- Makes human job easier
- High accuracy rate
- Machine Learning Techniques and Algorithms can be used
- Used in Banking Sector
- Subtask of the optical character recognition (OCR) problem.
- Guarantee the dataset is free from any redundant or irrelevant variables to the target variable
- An efficient handwritten digit recognition system based on support vector machines (SVM)
- Build a GUI in which you can draw the digit and recognize it

KEERTHIKA S

- Handwritten digits are not flawless
- Ability of computer to receive and interpret intelligible handwritten input from sources
- Based on shape analysis of the digit image and extract slant or slope information
- Optical character recognition(ORC)is the most mainstream technique used for handwriting recognition.
- Good database for people who want to try learning techniques and pattern recognition methods
- Errors are corrected using lexicons or spelling checkers
- A lot of language packages and different styles are provided.
- Convert handwritten digits into machine readable formats
- Internet connectivity issues

NIVETHA S

- hard task for the machine because handwritten digits are not perfect and it is in many different shapes and sizes.
- Image processing and deep learning can be used
- Calculating the size of effective field
- mostly used in communication devices
- system can adapt to the writing style of the user
- provides a word recognition feature
- Modified National Institute of Standards and Technology dataset
- Convolutional Neural Network can be used
- higher recognition rate of subsequently entered words because the detection depends on an integrated Dictionary

SWETHA S

- uses the digit image and recognizes the digit present in the image.
- processing of mobile phone numbers
- In this recognition, we face numerous challenges
- automatic processing of postal addresses
- capability of computer applications to recognize the human handwritten digits.
- Human efforts can be reduced
- Widely used and deeply understood dataset and, for the most part, is "solved."
- difficult assignment for the machine.
- SVM instead of neural n/w to improve program accuracy

Group ideas

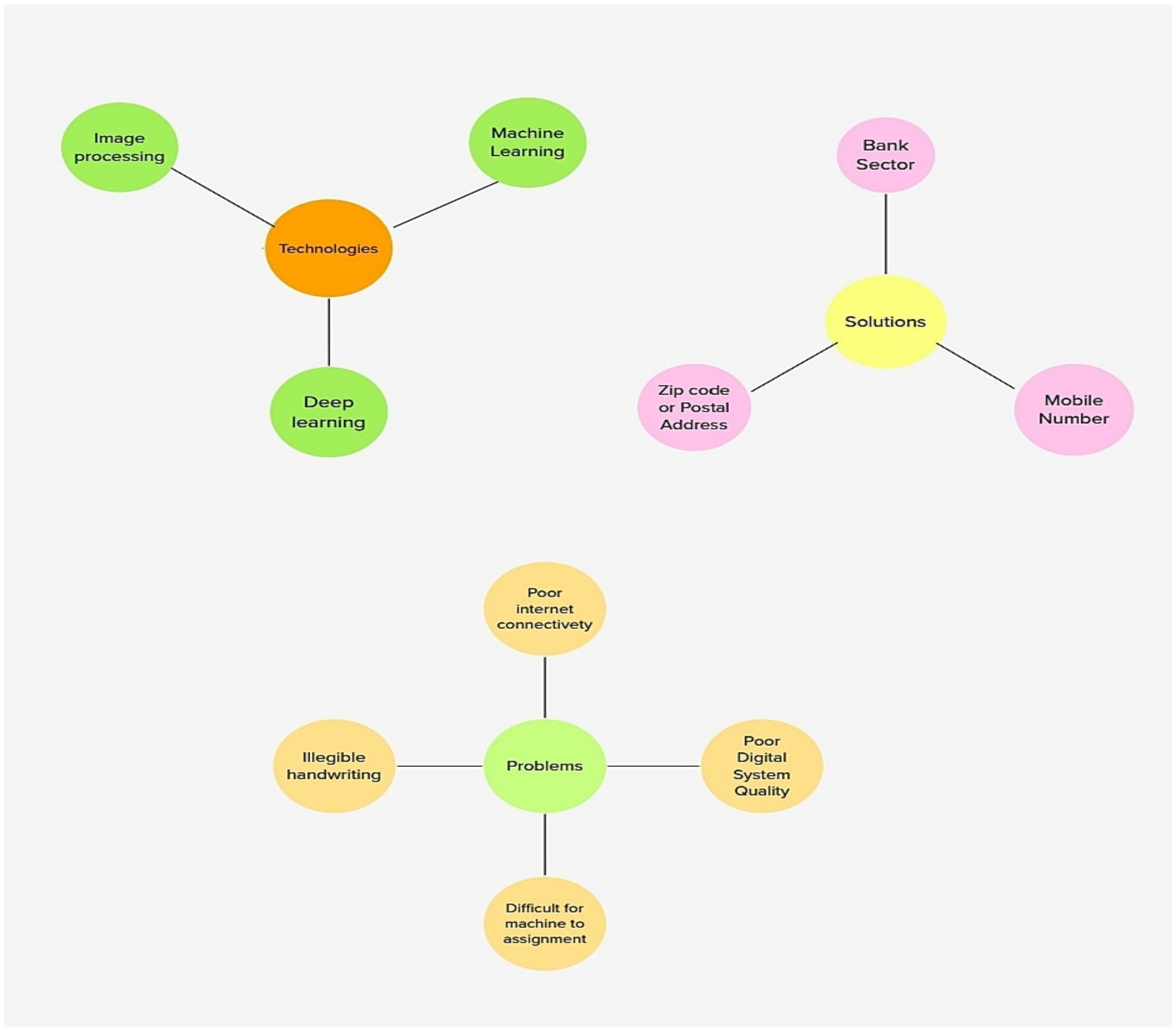
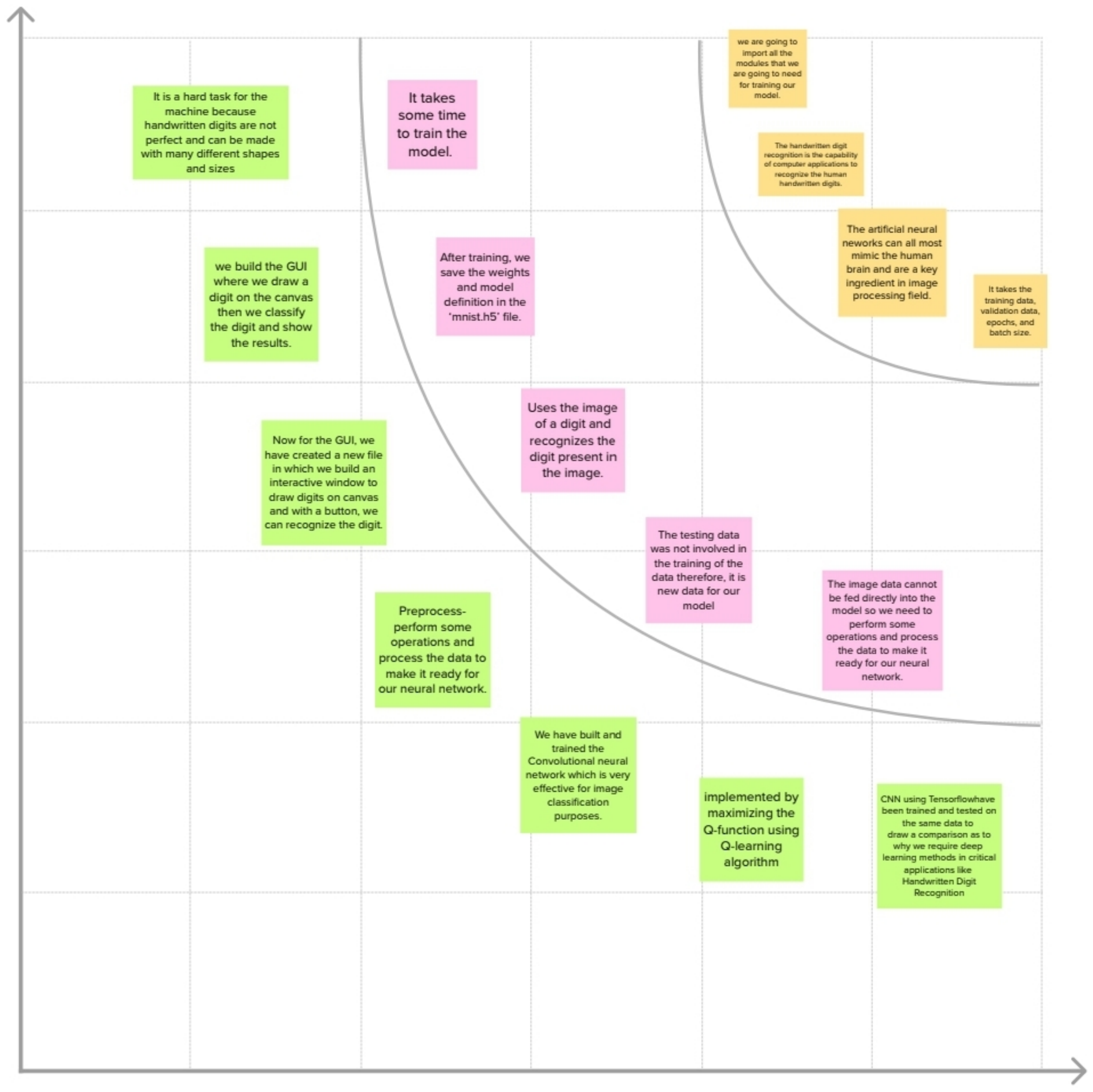


Image processing
Machine Learning
Technologies
Deep learning
Bank Sector
Solutions
Zip code or Postal Address
Mobile Number
Poor internet connectivety
Illegible handwriting
Problems
Poor Digital System Quality
Difficult for machine to assignment

Prioritize

⏱ 20 minutes


It is a hard task for the machine because handwritten digits are not perfect and can be made with many different shapes and sizes

It takes some time to train the model.

we are going to import all the modules that we are going to need for training our model.

The handwritten digit recognition is the capability of computer applications to recognize the human handwritten digits.

we build the GUI where we draw a digit on the canvas then we classify the digit and show the results.

After training, we save the weights and model definition in the 'mnist.h5' file.

The artificial neural neworks can all most mimic the human brain and are a key ingredient in image processing field.

It takes the training data, validation data, epochs, and batch size.

Uses the image of a digit and recognizes the digit present in the image.

Now for the GUI, we have created a new file in which we build an interactive window to draw digits on canvas and with a button, we can recognize the digit.

The testing data was not involved in the training of the data therefore, it is new data for our model

The image data cannot be fed directly into the model so we need to perform some operations and process the data to make it ready for our neural network.

Preprocess- perform some operations and process the data to make it ready for our neural network.

We have built and trained the Convolutional neural network which is very effective for image classification purposes.

implemented by maximizing the Q-function using Q-learning algorithm

CNN using Tensorflowhave been trained and tested on the same data to draw a comparison as to why we require deep learning methods in critical applications like Handwritten Digit Recognition

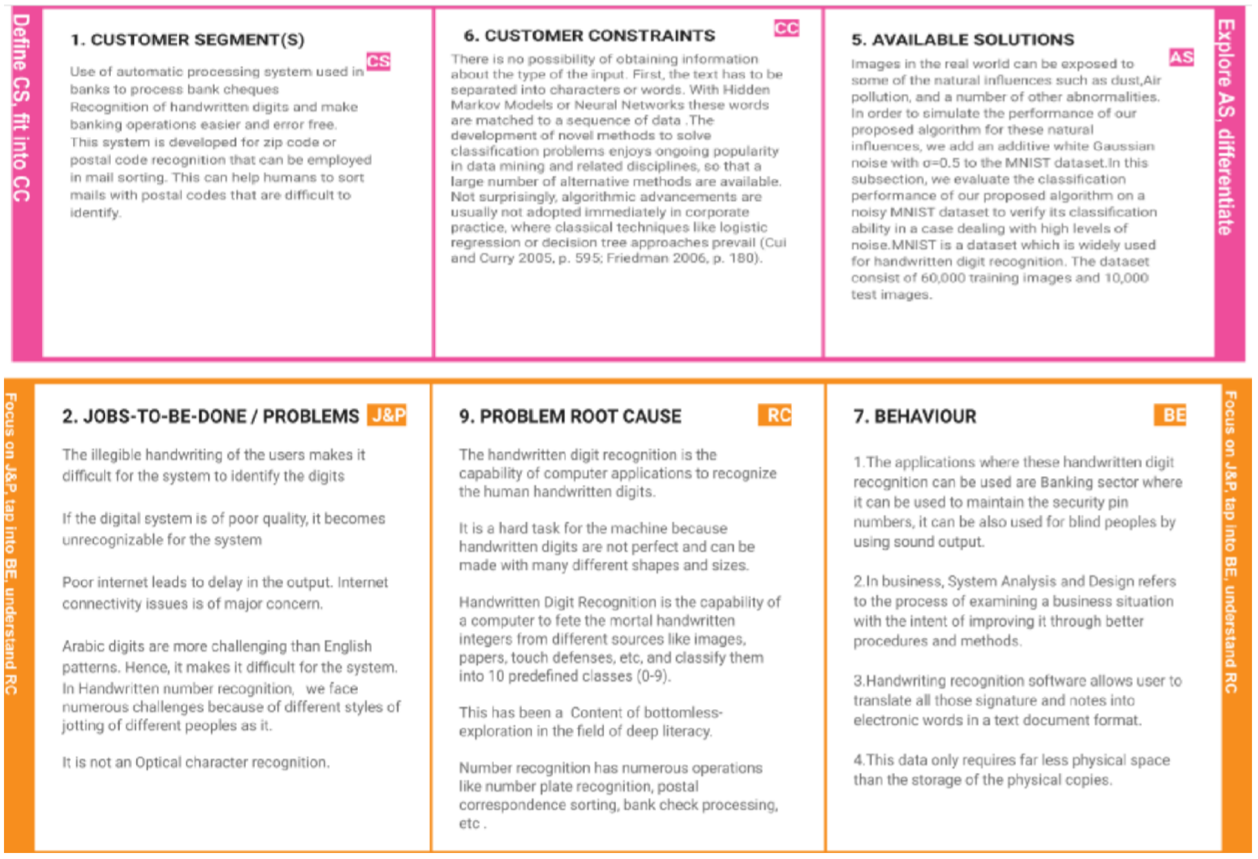
3.3 Proposed Solution

S.NO.	PARAMETER	DESCRIPTION
1	Problem Statement (Problem to be solved)	1. It can solve more complex problems and makes humans' job easier. This is a system widely used in the world to recognize zip code or postal code for mail sorting. There are different techniques that can be used to recognize handwritten characters. 2. The handwritten digit recognition is the capability of computer applications to recognize the human handwritten digits. It is a hard task for the machine because handwritten digits are not perfect and can be made with many different shapes and sizes.
2	Idea / Solution description	1. To design and implement a system using artificial intelligence, image processing and data mining concepts to take input as hand gestures. 2. A handwriting digit recognition system is to convert handwritten digits into machine readable formats. The main objective of this work is to ensure effective and reliable approaches for recognition of handwritten digits and make banking operations easier and error free.
3	Novelty / Uniqueness	There are different techniques that can be used to recognize handwritten characters. Two techniques researched in our project are Pattern Recognition and Artificial Neural Network

		(ANN) .
4	Social Impact / Customer Satisfaction	1. This is a system widely used in the world to recognize zip code or postal code for mail sorting. 2. The main objective of this work is to ensure effective and reliable approaches for recognition of handwritten digits and make banking operations easier and error free.
5	Business Model (Revenue Model)	Business process outsourcing Business smartization enhances automated AI Optical Recognition Protocols (ORPs) for OHR, thereby improving insight into customers and thus enabling them to make better decisions: 1. To promote business growth and innovation strategy. 2. To enhance customer experience and value proposition. 3. To increase through OHR business adaptability, alignment and agility. 4. To create data strategy, thereby influencing growth potential for knowledge intelligence.
6	Scalability of the Solution	1. Datasets of images contain 1,000 handwritten digits with 100 images of each digit that are used to train and test in Neural Network. 2. For image processing, captured images are converted into 16 x 16 pixels and transformed into a gray scale image. 3. Their work used a logistic regression algorithm to get the best probability of scanned images and got approximately

		90% accuracy in the results. 4. This system gave the best recognition accuracy of 94% and the worst recognition accuracy of 64%. 5. The average accuracy of this system is 82.5%. Therefore, this system needs a lot of training epochs to get higher accuracy. 6. The accuracy point was mainly focused in related works

3.4 Problem Solution Fit


Define CS, fit into CC
Explore AS, differentiate
Focus on J&P, tap into BE, understand RC
Focus on J&P, tap into BE, understand RC

1. CUSTOMER SEGMENT(S) CS

Use of automatic processing system used in banks to process bank cheques
Recognition of handwritten digits and make banking operations easier and error free.
This system is developed for zip code or postal code recognition that can be employed in mail sorting. This can help humans to sort mails with postal codes that are difficult to identify.

6. CUSTOMER CONSTRAINTS CC

There is no possibility of obtaining information about the type of the input. First, the text has to be separated into characters or words. With Hidden Markov Models or Neural Networks these words are matched to a sequence of data .The development of novel methods to solve classification problems enjoys ongoing popularity in data mining and related disciplines, so that a large number of alternative methods are available. Not surprisingly, algorithmic advancements are usually not adopted immediately in corporate practice, where classical techniques like logistic regression or decision tree approaches prevail (Cui and Curry 2005, p. 595; Friedman 2006, p. 180).

5. AVAILABLE SOLUTIONS AS

Images in the real world can be exposed to some of the natural influences such as dust,Air pollution, and a number of other abnormalities. In order to simulate the performance of our proposed algorithm for these natural influences, we add an additive white Gaussian noise with σ=0.5 to the MNIST dataset.In this subsection, we evaluate the classification performance of our proposed algorithm on a noisy MNIST dataset to verify its classification ability in a case dealing with high levels of noise.MNIST is a dataset which is widely used for handwritten digit recognition. The dataset consist of 60,000 training images and 10,000 test images.

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

The illegible handwriting of the users makes it difficult for the system to identify the digits

If the digital system is of poor quality, it becomes unrecognizable for the system

Poor internet leads to delay in the output. Internet connectivity issues is of major concern.

Arabic digits are more challenging than English patterns. Hence, it makes it difficult for the system. In Handwritten number recognition, we face numerous challenges because of different styles of jotting of different peoples as it.

It is not an Optical character recognition.

9. PROBLEM ROOT CAUSE RC

The handwritten digit recognition is the capability of computer applications to recognize the human handwritten digits.

It is a hard task for the machine because handwritten digits are not perfect and can be made with many different shapes and sizes.

Handwritten Digit Recognition is the capability of a computer to fete the mortal handwritten integers from different sources like images, papers, touch defenses, etc, and classify them into 10 predefined classes (0-9).

This has been a Content of bottomless-exploration in the field of deep literacy.

Number recognition has numerous operations like number plate recognition, postal correspondence sorting, bank check processing, etc .

7. BEHAVIOUR BE

1.The applications where these handwritten digit recognition can be used are Banking sector where it can be used to maintain the security pin numbers, it can be also used for blind peoples by using sound output.

2.In business, System Analysis and Design refers to the process of examining a business situation with the intent of improving it through better procedures and methods.

3.Handwriting recognition software allows user to translate all those signature and notes into electronic words in a text document format.

4.This data only requires far less physical space than the storage of the physical copies.

3. TRIGGERS

TR

Human effort can be reduced.

Bank can employ less employees who are as efficient as the computerized processing system

.

4. EMOTIONS: BEFORE / AFTER

EM

Handwriting recognition software allows users to translate all those signatures and notes into electronic words in a text document format.

This data only requires far less physical space than the storage of the physical copies.

.

It helps humans ease their jobs and solve more complex problems.

10. YOUR SOLUTION

SL

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.

This exploration provides a comprehensive comparison between different machine literacy and deep literacy algorithms for the purpose of handwritten number recognition.

For this, we've used Support Vector Machine, Multilayer Perceptron, and Convolutional Neural Network.

The comparison between these algorithms is carried out on the base of their delicacy, crimes, and testing- training time corroborated by plots and maps that have been constructed using matplotlib for visualization

8.CHANNELS of BEHAVIOR

CH

8.1 ONLINE

Isolated-characters and word recognizers are the most basic type of classifiers currently used in handwriting recognition.

The development of classifiers is attempted using a variety of techniques ranging from struc-tural and rule-based methods to statistical modeling.

The performance of rule-based methods is limited by the capabilities of the designer to reliably devise the set of rules.

On the other hand, statistical approaches generally require a large amount of data for training.

Such classi-fiers require a fixed number of features in multidimensional feature space.

The problem now is to define a separation boundary between classes in this feature space.

This often leads to complex solution representations and intolerance to noise and consequently large generalization errors.

As a result heavy emphasis is placed on the preprocessing stage prior to classification in order to effectively reduce these problems.

8.2 OFFLINE

In the domain of off-line handwritten character recognition there are two key strategies in current use.

They can be broadly grouped as 'active' and 'passive' character recognition.

At the heart of our approach is the feature extraction routine that divides the image into a quad tree.

Features are then extracted based on the contour representation in the sub-image.

Hierarchy is maintained by dividing each sub-image into deeper quad trees in order to increase the level of detail.

For most datasets a depth of 4 is sufficient. The features from all these levels are then placed in a feature vector.

This feature vector is then presented to the GP classifier.

The same technique can be adopted using quin tree representation.

4. REQUIREMENT ANALYSIS

4.1 Functional requirements

Import the Modified National Institute of Science and Technology Dataset:

- Import dataset file directly to the program from a command that will download the dataset from its website.
- Save the dataset file in the same directory as the program

  -Create the model.

- Take the value of the color is the pixels.
- Put the value of all the pixels in a one-dimensional array.
- Build a Neural Network with a number of nodes in the input layer equal to the number of pixels in the arrays.
- Activate the Neural Network.
- Test the precision of the model in a testing set

  - Recognize handwritten number input.

- Allow user to input a number using a touchscreen.
- Predict in real-time the value of the number written.

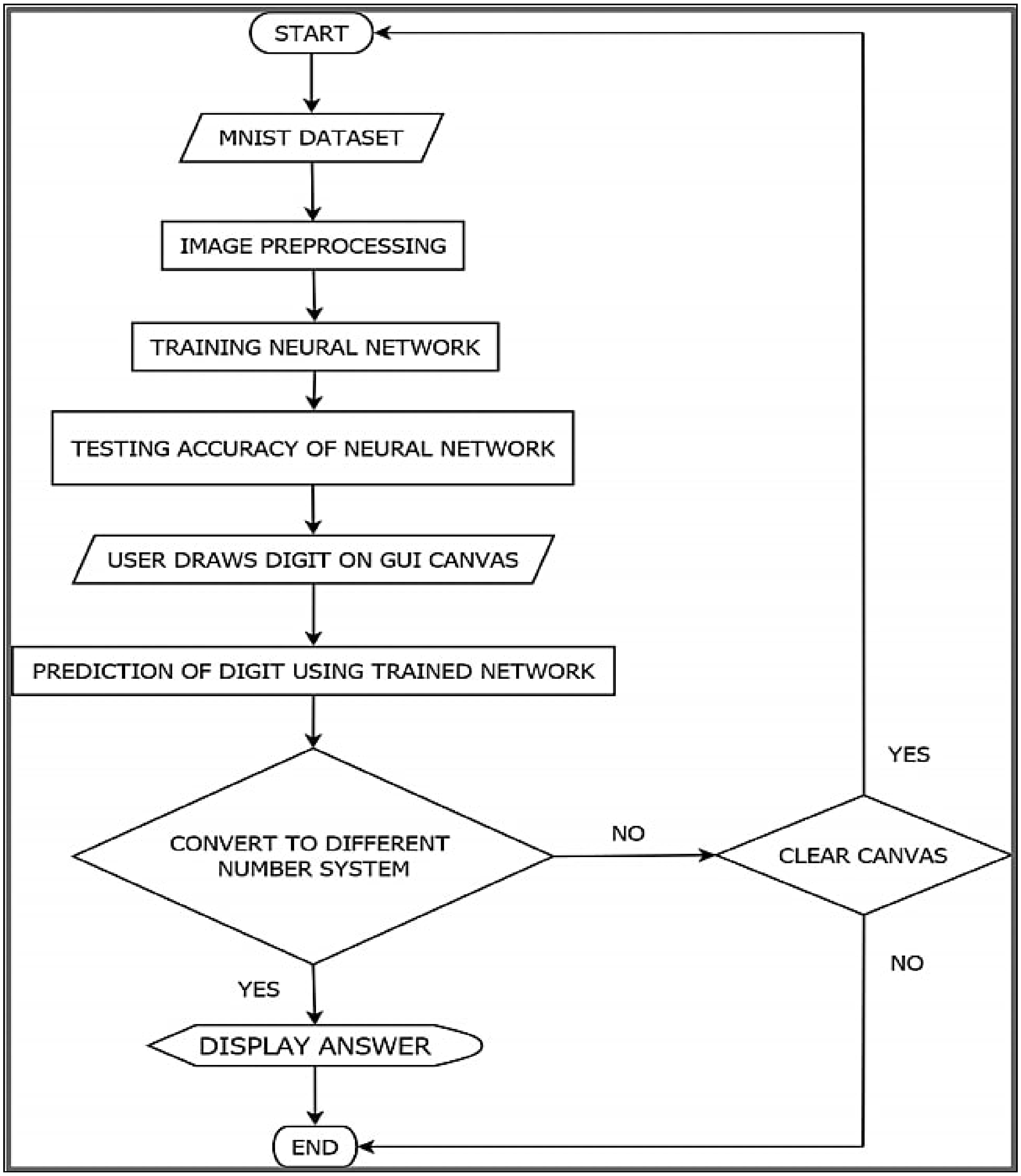
4.2 Non-Functional requirements

Windows Application

- Using Python.
- Using Tensorflow.

5. PROJECT DESIGN

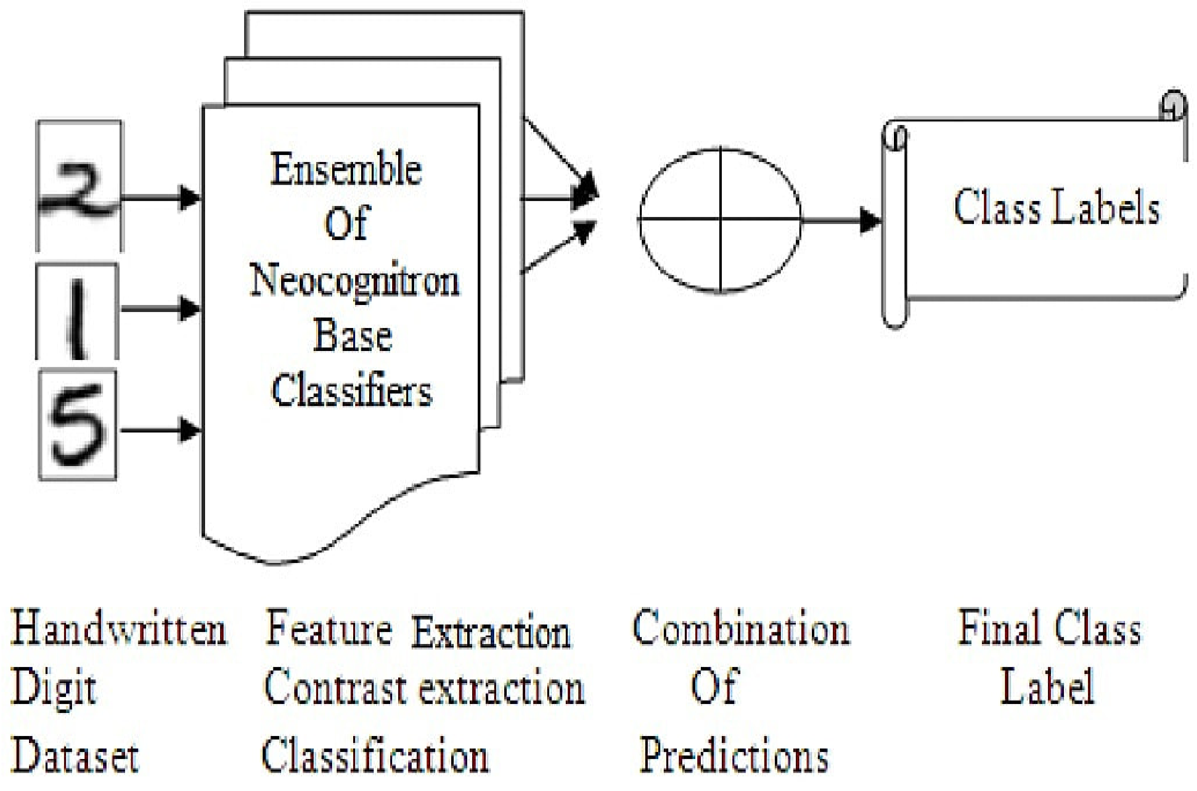
5.1 Data Flow Diagram

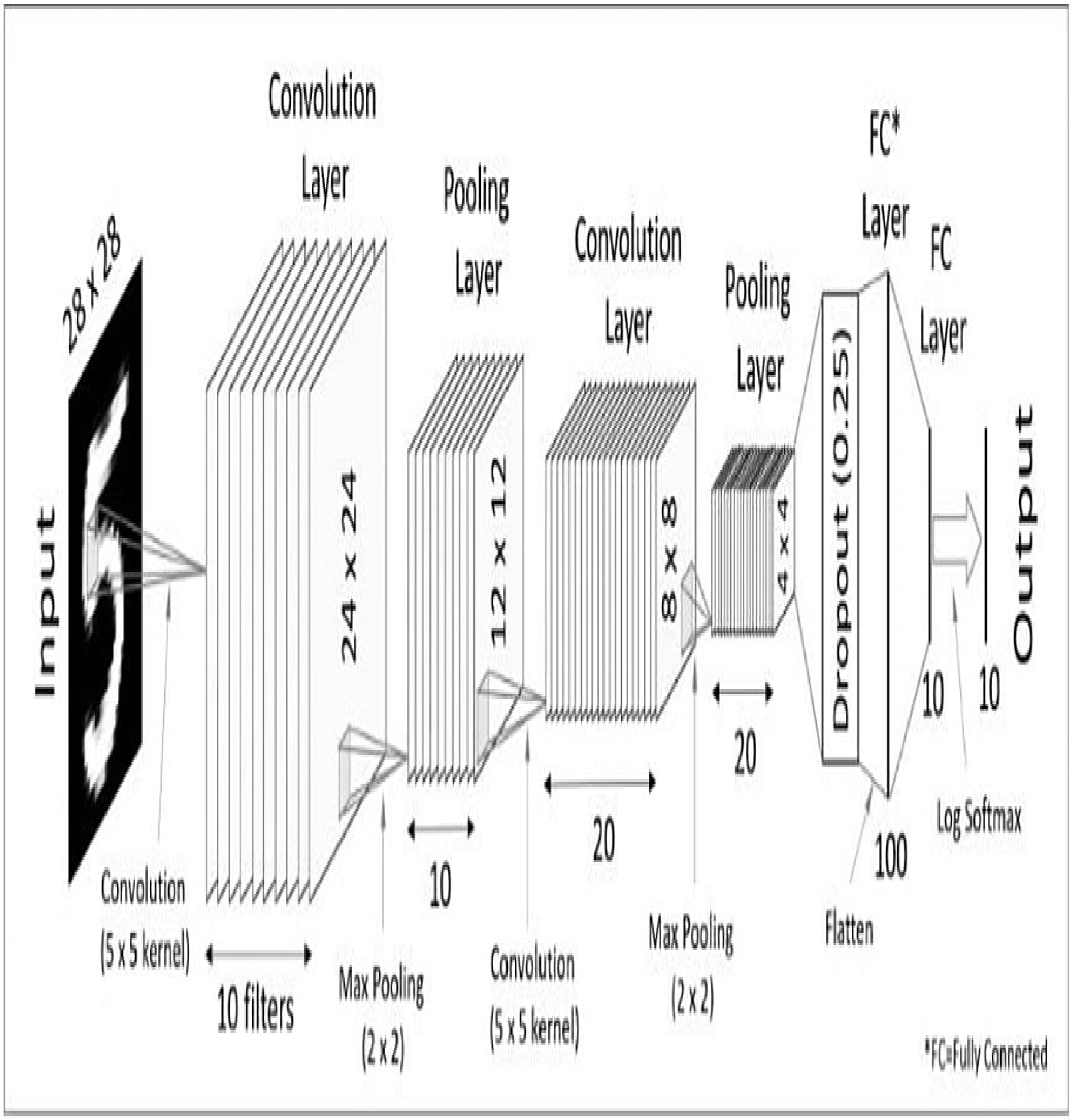

START
MNIST DATASET
IMAGE PREPROCESSING
TRAINING NEURAL NETWORK
TESTING ACCURACY OF NEURAL NETWORK
USER DRAWS DIGIT ON GUI CANVAS
PREDICTION OF DIGIT USING TRAINED NETWORK
CONVERT TO DIFFERENT NUMBER SYSTEM
NO
CLEAR CANVAS
YES
NO
YES
DISPLAY ANSWER
END

5.2 Solution and 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.


Ensemble Of Neocognitron Base Classifiers
Class Labels
Handwritten Digit Dataset
Feature Extraction Contrast extraction Classification
Combination Of Predictions
Final Class Label


Input
28 x 28
Convolution (5 x 5 kernel)
Convolution Layer
24 x 24
10 filters
Max Pooling (2 x 2)
Pooling Layer
12 x 12
10
Convolution (5 x 5 kernel)
Convolution Layer
8 x 8
20
Max Pooling (2 x 2)
Pooling Layer
4 x 4
20
Flatten
Dropout (0.25)
FC* Layer
100
FC Layer
10
Log Softmax
Output
10
*FC=Fully Connected

5.3 User Stories

User Type	Functional Requirements	User Story Number	User Story/Task	Acceptance Criteria	Priority	Release
Customer	Accessing the Application	USN-1	As a user, I should beable to access the application from anywhere and use onany devices.	User can access the application using the browser on any device.	High	Sprint-4
	Uploading Image	USN-2	As a user, I should beable to upload images to predict the digits.	User can upload images	High	Sprint-3
	Viewing the Results	USN-3	As a user, I should beable to view the results.	The result of the prediction is displayed.	High	Sprint-3
	Viewing Other Prediction	USN-4	As a user, Ishould be able to see other close predictions.	The accuracy of othervalues must be displayed.	Medium	Sprint-4
	Usage Instruction	USN-5	As a user, I shouldhave a usage instruction to knowhow to use the application	The usage instruction is displayed on the homepage.	Medium	Sprint-4

6.PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

TITLE	DESCRIPTION	DATE
Prepare Empathy Map	Prepare Empathy Map Canvas to capture the user Pains & Gains, Prepare list of problem statements.	17 SEPTEMBER 2022
Literature Survey & Information Gathering	Literature survey on the selected project & gathering information by referring the, technical papers , research publications etc.	18 SEPTEMBER 2022
Ideation	List the by organizing the brainstorming session and prioritize the top 3 ideas based on the feasibility & importance.	18 SEPTEMBER 2022
Proposed Solution	Creation of proposed solution document, which includes the novelty, feasibility of idea, business model, social impact, scalability of solution, etc.	27 SEPTEMBER 2022
Problem Solution Fit	Creation of problem solution fit document.	02 OCTOBER 2022
Solution Architecture	Creation of solution architecture document.	09 OCTOBER 2022
Customer Journey	Prepare the customer journey maps to understand the user interactions & experiences with the application.	30 OCTOBER 2022
Data Flow Diagrams	Draw the data flow diagrams and submit for review.	30 OCTOBER 2022
Technology Architecture	Prepare the technology architecture diagram.	30 OCTOBER 2022
Requirement Analysis	Prepare the requirement analysis for the solution.	30 OCTOBER 2022

6.2 Sprint Delivery Schedule

Project Tracker, Velocity & Burndown Chart: (4 Marks)

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	02 Nov 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

Velocity:

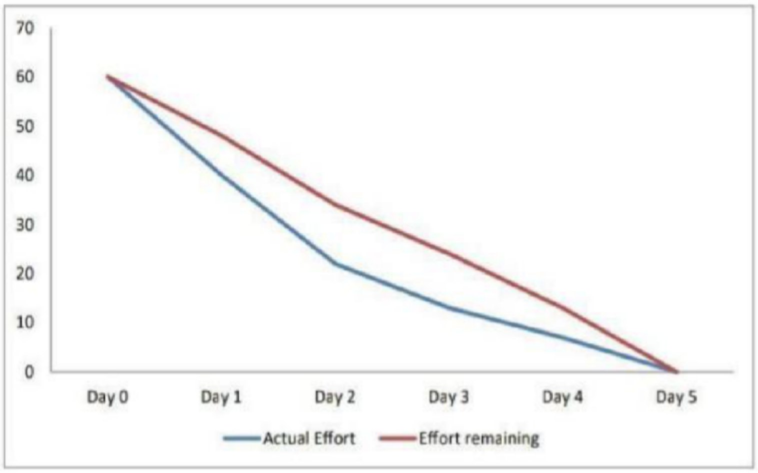
Imagine we have a 6-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

AV=sprint duration/Velocity

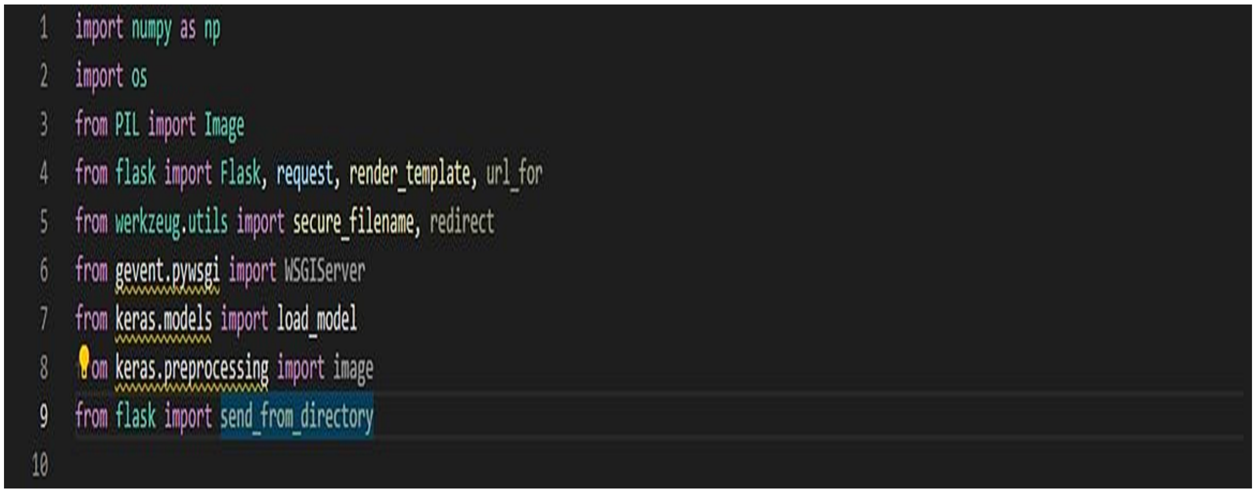
=20/6=3.33

Burndown Chart:

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

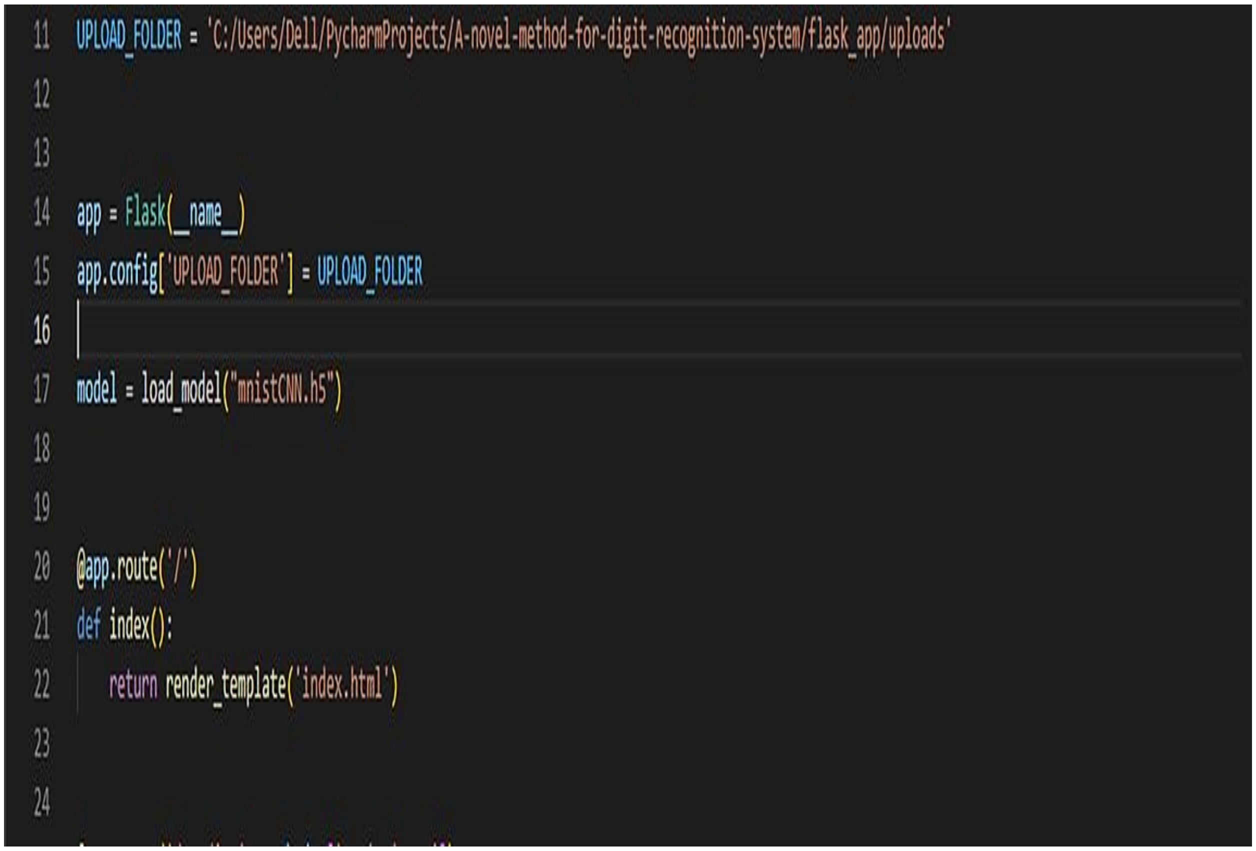


7. CODING & SOLUTIONING



```python
import numpy as np
import os
from PIL import Image
from flask import Flask, request, render_template, url_for
from werkzeug.utils import secure_filename, redirect
from gevent.pywsgi import WSGIServer
from keras.models import load_model
om keras.preprocessing import image
from flask import send_from_directory

```



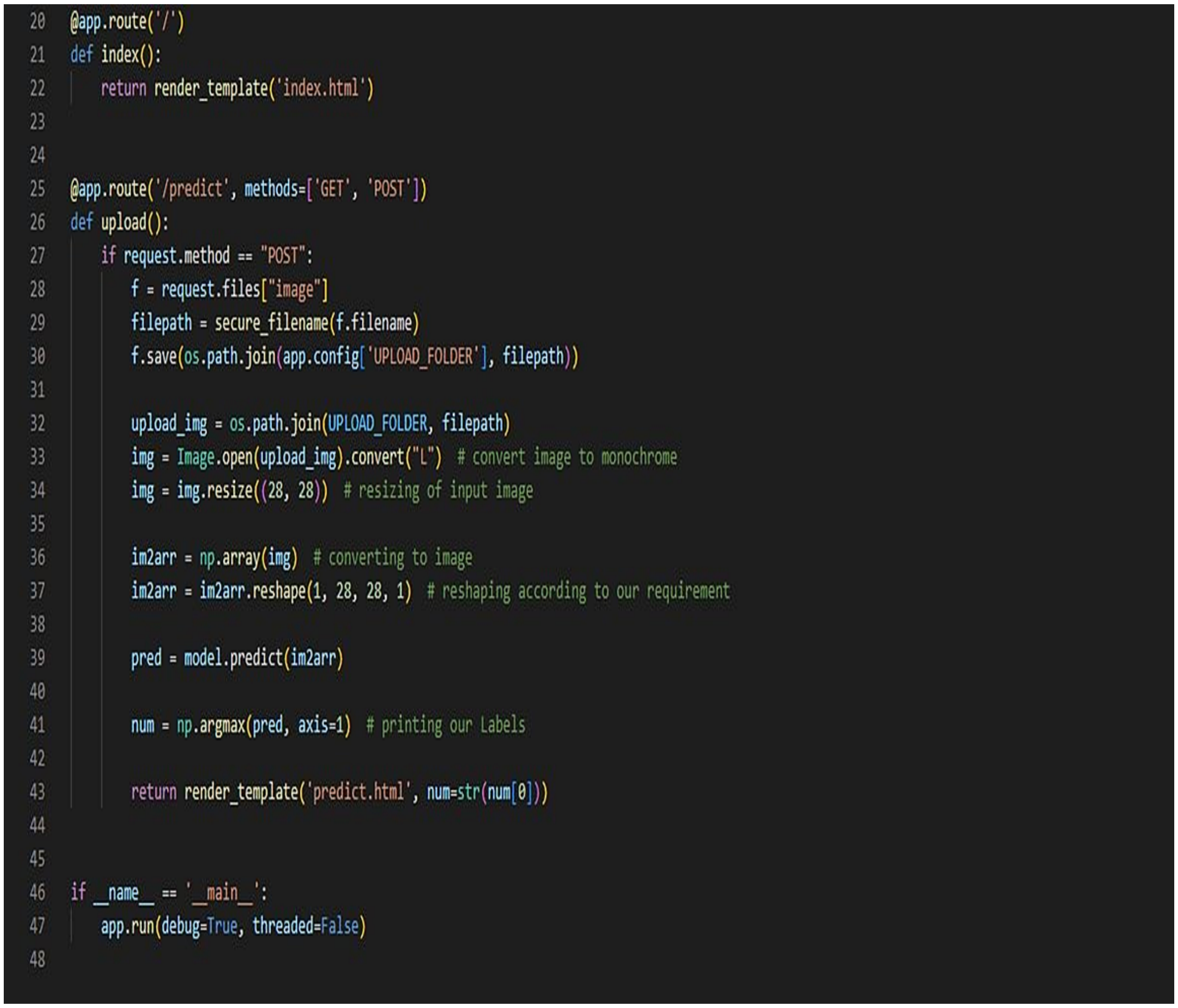
```python
UPLOAD_FOLDER = 'C:/Users/Dell/PycharmProjects/A-novel-method-for-digit-recognition-system/flask_app/uploads'


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

model = load_model("mnistCNN.h5")


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


```



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


@app.route('/predict', methods=['GET', 'POST'])
def upload():
    if request.method == "POST":
        f = request.files["image"]
        filepath = secure_filename(f.filename)
        f.save(os.path.join(app.config['UPLOAD_FOLDER'], filepath))

        upload_img = os.path.join(UPLOAD_FOLDER, filepath)
        img = Image.open(upload_img).convert("L")  # convert image to monochrome
        img = img.resize((28, 28))  # resizing of input image

        im2arr = np.array(img)  # converting to image
        im2arr = im2arr.reshape(1, 28, 28, 1)  # reshaping according to our requirement

        pred = model.predict(im2arr)

        num = np.argmax(pred, axis=1)  # printing our Labels

        return render_template('predict.html', num=str(num[0]))


if __name__ == '__main__':
    app.run(debug=True, threaded=False)

```

8. TESTING

8.1 Test Cases

Test case ID	Feature Type	Component	Test Scenario	Expected Result	Actual Result	Status
HP_TC_001	UI	Home Page	Verify UI elements inthe Home Page	The Home page must be displayed properly	Working as expected	FAIL
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 allowuser to select anon image file	User is ableto 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 variousimage 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 inputimage	The model shouldpredict the number in the complex image	The model failsto identify the digit since themodel is not built to handlesuch data	FAIL
RP_TC_001	UI	Result Page	Verify UI elements inthe Result Page	The Result page mustbe displayed properly	Working as expected	PASS
RP_TC_0 02	UI	Result Page	Check if the input image isdisplayed properly	The input image should be displayed properly	The size of theinput image exceeds the display container	FAIL
RP_TC_0 03	UI	Result Page	Check if the result is displayed properly	The result shouldbe displayed properly	Working as expected	PASS

8.2 User Acceptance Testing

DEFECT ANALYSIS

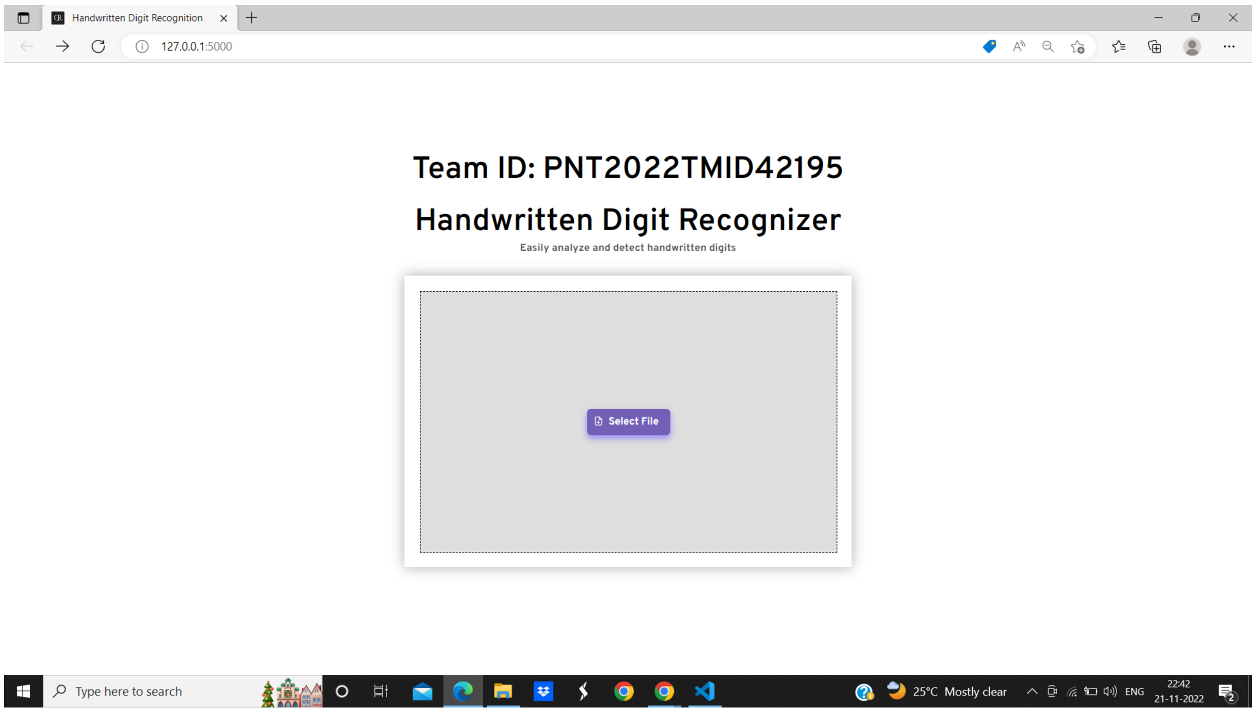
Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Total
By Design	1	0	1	0	2
Duplicate	0	0	0	0	0
External	0	0	2	0	2
Fixed	4	1	0	1	6
Not Reproduced	0	0	0	1	1
Skipped	0	0	0	1	1
Won’t Fix	1	0	1	0	2
Total	6	1	4	3	14

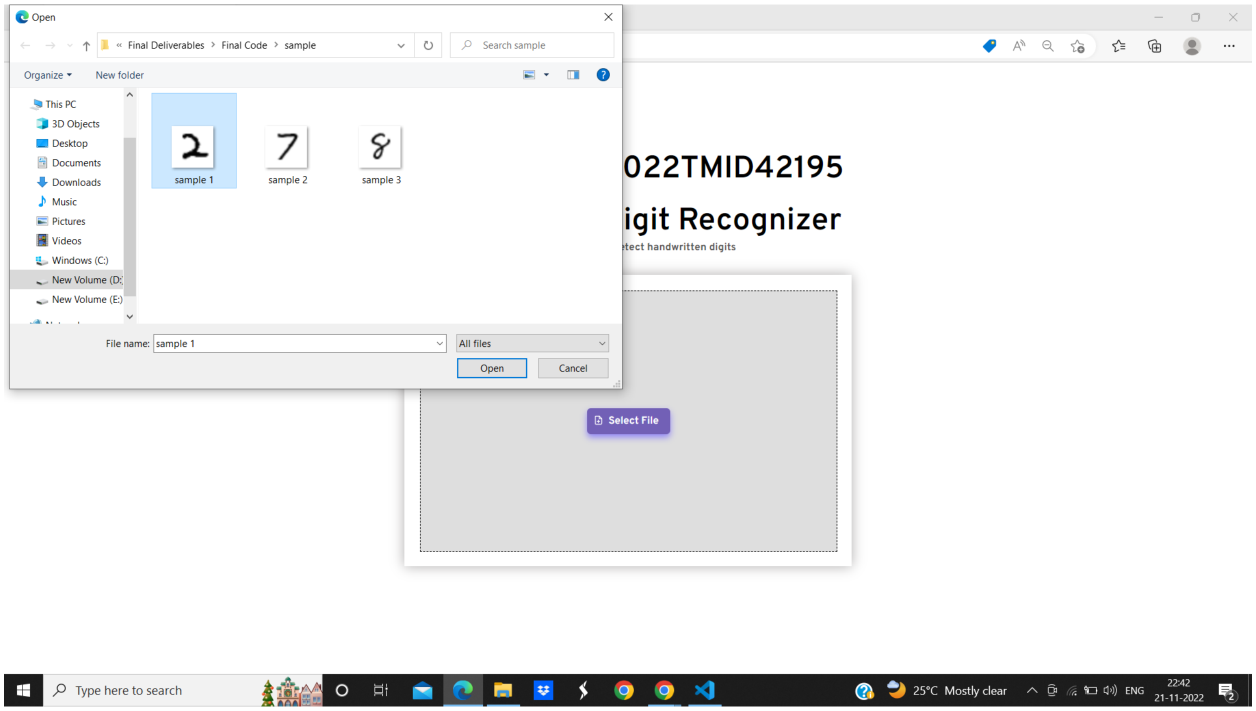
TEST CASE ANALYSIS

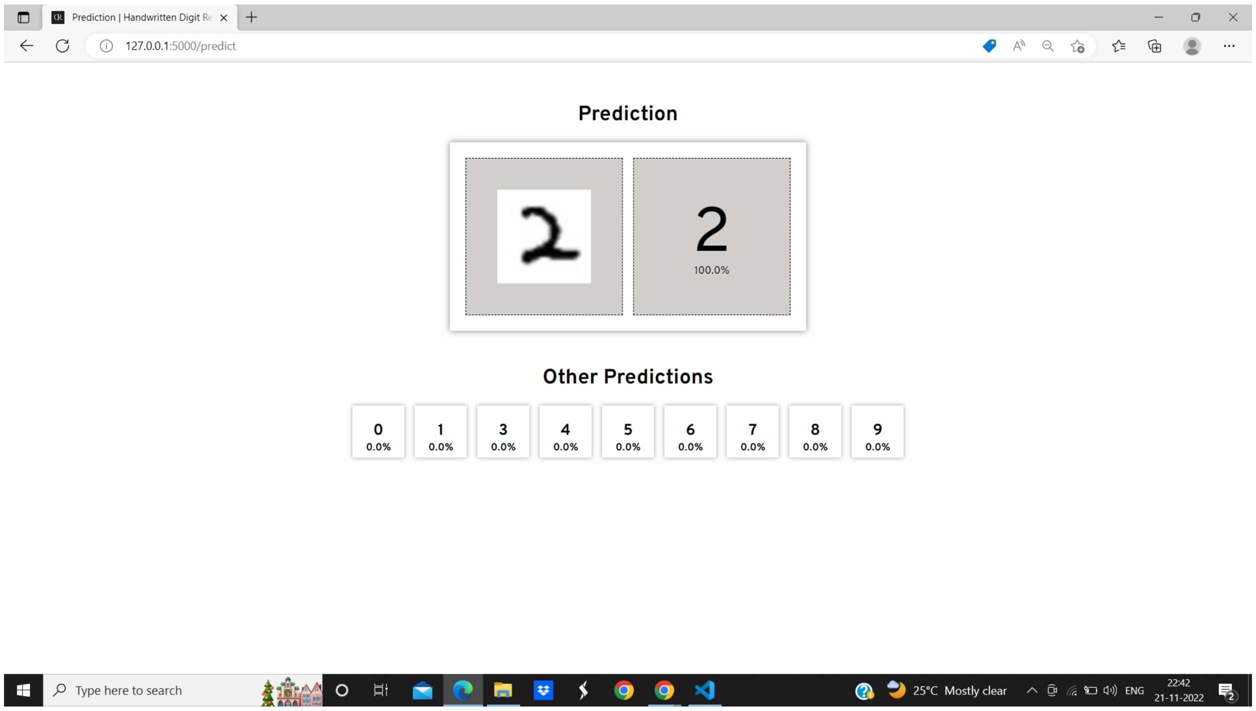
Section	Total Cases	Not Tested	Fail	Pass
Client Application	10	0	3	7
Security	2	0	1	1
Performance	3	0	1	2
Exception Reporting	2	0	0	2

9. RESULTS

9.1 Performance Metrics


Handwritten Digit Recognition
127.0.0.1:5000
Team ID: PNT2022TMID42195
Handwritten Digit Recognizer
Easily analyze and detect handwritten digits
Select File
Type here to search
25°C Mostly clear
22:42
21-11-2022





Prediction | Handwritten Digit Re

127.0.0.1:5000/predict

Prediction

2

100.0%

Other Predictions

0	1	3	4	5	6	7	8	9
0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%

10. ADVANTAGES & DISADVANTAGES

Advantages:

- 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.
- The generative models can perform recognition driven segmentation.
- The method involves a relatively small number of parameters and hence training is relatively easy and fast.
- 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:

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

11. CONCLUSION

The performance of CNN for handwritten recognition performed significantly. The proposed method obtained around 98% accuracy and is able to identify real-world images as well; the loss percentage obtained in both training and evaluation is almost negligible. The only difficult part is the noise present in the input canvas image, which needs to be taken care of. The learning rate of the model is much dependent on the number of dense neurons and the cross-validation measure.

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

Source Code:

13.1 Importing Libraries:

```
import cv2
import numpy as np
from keras.datasets import mnist
from keras.layers import Dense, Flatten
from keras.layers.convolutional import Conv2D
from keras.models import Sequential
from keras.utils import to_categorical
import matplotlib.pyplot as plt
```

13.2 Downloading the MNIST data:

```
(X_train, y_train), (X_test, y_test) = mnist.load_data()
```

13.3 Checking the gray scale picture from MNIST dataset:

```
plt.imshow(X_train[0], cmap="gray")
plt.show()
print (y_train[0])
```

13.4 Test and Train the data:

```
print ("Shape of X_train: {}".format(X_train.shape))
print ("Shape of y_train: {}".format(y_train.shape))
print ("Shape of X_test: {}".format(X_test.shape))
print ("Shape of y_test: {}".format(y_test.shape))
```

13.5 Reshape the data:

```
X_train = X_train.reshape(60000, 28, 28, 1)
X_test = X_test.reshape(10000, 28, 28, 1)
```

13.6 Updating the Test and Train values:

```
y_train = to_categorical(y_train)
y_test = to_categorical(y_test)
```

13.7 Adding the Hidden layers to the dataset:

```
model = Sequential()
layer_1 = Conv2D(32, kernel_size=3, activation="relu" , input_shape=(28, 28,1))
layer_2 = Conv2D(64, kernel_size=3, activation="relu")
layer_3 = Flatten()
layer_4 = Dense(10, activation="softmax")

## Add the layers to the model
model.add(layer_1)
model.add(layer_2)
model.add(layer_3)
model.add(layer_4)
```

13.8 Compiling the model:

```
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
```

13.9 Fit the model or Validate the model:

```
model.fit(X_train, y_train, validation_data=(X_test, y_test), epochs=3)
```

13.10 Predict the Training data:

```
example = X_train[364]
prediction = model.predict(example.reshape(1, 28, 28, 1))## First output
print ("Prediction (Softmax) from the neural network:\n\n {}".format(prediction))##
Second output
hard_maxed_prediction = np.zeros(prediction.shape)
hard_maxed_prediction[0][np.argmax(prediction)] = 1
print("\n\nHard-maxed form of the prediction: \n\n
{}".format(hard_maxed_prediction))## Third output
print("\n\n--------- Prediction --------- \n\n")
plt.imshow(example.reshape(28, 28), cmap="gray")
plt.show()
print("\n\nFinal Output: {}".format(np.argmax(prediction)))
```

13.11 Test the Real time image:

```
image = cv2.imread('/content/test_image.jpg')
grey = cv2.cvtColor(image.copy(), cv2.COLOR_BGR2GRAY)
ret, thresh = cv2.threshold(grey.copy(), 75, 255, cv2.THRESH_BINARY_INV)
```

```
contours, _ = cv2.findContours(thresh, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIM
PLE)
preprocessed_digits = []
for c in contours:
x,y,w,h = cv2.boundingRect(c)

# Creating a rectangle around the digit in the original image (for displaying the digits
fetched via contours)
cv2.rectangle(image, (x,y), (x+w, y+h), color=(0, 255, 0), thickness=2)

# Cropping out the digit from the image corresponding to the current contours in the for
loop
digit = thresh[y:y+h, x:x+w]

# Resizing that digit to (18, 18)
resized_digit = cv2.resize(digit, (18,18))

# Padding the digit with 5 pixels of black color (zeros) in each side
to finally produce the image of (28, 28)
padded_digit = np.pad(resized_digit, ((5,5),(5,5)), "constant", consta
nt_values=0)

# Adding the preprocessed digit to the list of preprocessed digits
preprocessed_digits.append(padded_digit)
print("\n\n\n----------------Contoured Image--------------------")
plt.imshow(image, cmap="gray")
plt.show()
inp = np.array(preprocessed_digits)
```

FLASK

```
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 random_name_generator(n: int) -> str:
	"""
	Generates a random file name.

	Args:
		n (int): Length the of the file name.

	Returns:
		str: The file name.
	"""
	return ''.join(random.choices(string.ascii_uppercase + string.digits, k=n))

def recognize(image: bytes) -> tuple:
	"""
	Predicts the digit in the image.

	Args:
		image (bytes): The image data.

	Returns:
		tuple: The best prediction, other predictions and file name
	"""

	model=load_model(Path("./model/model.h5"))
```

```
img = Image.open(image).convert("L")

# Generate a random name to save the image file.
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}"))

# Convert the Image to Grayscale, Invert it and Resize to get better prediction.
img = ImageOps.grayscale(img)
img = ImageOps.invert(img)
img = img.resize((28, 28))

# Convert the image to an array and reshape the data to make prediction.
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]

# Get all the predictions and it's respective accuracy.
pred = list(map(lambda x: round(x*100, 2), results[0]))

values = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
others = list(zip(values, pred))

# Get the value with the highest accuracy
best = others.pop(best)

return best, others, img_name
```

HTML CODE

index.html

```
<head>
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Handwritten Digit Recognition</title>
                        <link rel="icon" type="image/svg" sizes="32x32"
href="{{url_for('static',filename='images/icon.svg')}}" />
    <link rel="stylesheet" href="{{url_for('static',filename='css/main.css')}}" />
    <script src="https://unpkg.com/feather-icons"></script>
    <script defer src="{{url_for('static',filename='js/script.js')}}"></script>
  </head>
  <body>
    <div class="container">
      <div class="heading">
        <h3 class="heading__main"> Team ID: PNT2022TMID42195 </h3>
        <br>
        <h1 class="heading__main"> Handwritten Digit Recognizer</h1>
        <h2 class="heading__sub">Easily analyze and detect handwritten digits</h2>
      </div>
      <div class="upload-container">
        <div class="form-wrapper">
                        <form class="upload" action="/predict" method="post"
enctype="multipart/form-data">
              <label id="label" for="upload-image"><i data-feather="file-plus"></i>Select
File</label>
            <input type="file" name="photo" id="upload-image" hidden />
            <button type="submit" id="up_btn"></button>
          </form>
          <img id="loading" src="{{url_for('static',filename='images/loading.gif')}}">
        </div>
      </div>
    </div>
  </body>
</html>
```

predict.html

```
<html>
        <head>
                <title>Prediction | Handwritten Digit Recognition</title>
                <link rel="stylesheet" href="{{url_for('static',filename='css/predict.css')}}" />
                <link            rel="icon"            type="image/svg"            sizes="32x32"
href="{{url_for('static',filename='images/icon.svg')}}" />
                <meta name="viewport" content="width=device-width, initial-scale=1.0" />
        </head>
        <body>
                <div class="container">
                        <h1>Prediction</h1>
                        <div class="result-wrapper">
                                <div class="input-image-container">
                                        <img
src="{{url_for('static',filename='data/')}}{{img_name}}" />
                                </div>
                                <div class="result-container">
                                        <div class="value">{{best.0}}</div>
                                        <div class="accuracy">{{best.1}}%</div>
                                </div>
                        </div>
                        <h1>Other Predictions</h1>
                        <div class="other_predictions">
                                {% for x in others %}
                                <div class="value">
                                        <h2>{{x.0}}</h2>
                                        <div class="accuracy">{{x.1}}%</div>
                                </div>
                                {% endfor %}
                        </div>
                </div>
        </body>
</html>
```

GITHUB LINK

https://github.com/IBM-EPBL/IBM-Project-1006-1658334398

PROJECT DEMO LINK

https://www.youtube.com/watch?v=PPtdodUUPek