A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

A IBM PROJECT REPORT

Submitted by

TEAM ID : PNT22022TMID32105

SHIVASURYAVASAN S (731619205046)

SURYAPRAKASH (731619205053)

SUDHARSON D (731619205051)

BOOPATHI S (731619205005)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

KSR INSTITUTE FOR ENGINEERING AND TECHNOLOGY, TIRUCHENGODE

ANNA UNIVERSITY : CHENNAI 600 025

NOVEMBER 2022

A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

TEAM ID	PNT22022TMID32105
PROJECT NAME	A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM
TEAM MEMBERS	SHIVA SURYA VASAN S SURYAPRAKASH S SUDHARSON D BOOPATHI S

A PROJECT REPORT

CONTENTS

1. INTRODUCTION

PROJECT OVERVIEW

Handwritten digit recognition using MNIST dataset is a major project made with the help of Neural Network. It basically detects the scanned images of handwritten digits. We have taken this a step further where our handwritten digit recognition system not only detects scanned images of handwritten digits but also allows writing digits on the screen with the help of an integrated GUI for recognition.

Approach:

We will approach this project by using a three-layered Neural Network.

The input layer: It distributes the features of our examples to the next layer for calculation of activations of the next layer.

The hidden layer: They are made of hidden units called activations providing nonlinear ties for the network. A number of hidden layers can vary according to our requirements.

The output layer: The nodes here are called output units. It provides us with the final prediction of the Neural Network on the basis of which final predictions can be made.

A neural network is a model inspired by how the brain works. It consists of multiple layers having many activations, this activation resembles neurons of our brain. A neural network tries to learn a set of parameters in a set of data which could help to recognize the underlying relationships. Neural networks can adapt to changing input; so the network generates the best possible result without needing to redesign the output criteria.

Methodology:

We have implemented a Neural Network with 1 hidden layer having 100 activation units (excluding bias units). The data is loaded from a .mat file, features(X) and labels(y) were extracted. Then features are divided by 255 to rescale them into a range of [0, 1] to avoid overflow during computation. Data is split up into 60,000 training and 10,000 testing examples. Feed forward is performed with the training set for calculating the hypothesis and then back propagation is done in order to reduce the error between the layers. The regularization parameter lambda is set to 0.1 to address the problem of over fitting. Optimizer is run for 70 iterations to find the best fit model.

PURPOSE

- The issue of transcribed digit acknowledgment has for some time been an open issue in the field of example order. A few examined have demonstrated that Neural Network has an incredible execution in information arrangement. The fundamental target of this paper is to give effective and solid procedures to acknowledgment of transcribed numerical by looking at different existing arrangement models. This paper thinks about the exhibition of Convolutional Neural Network (CCN). Results demonstrate that CNN classifier beat over Neural Network with critical improved computational effectiveness without relinquishing execution. Handwritten digit recognition can be performed using the Convolutional neural network from Machine Learning.

- Using the MNIST (Modified National Institute of Standards and Technologies) database and compiling with the CNN gives the basic structure of my project development. So, basically to perform the model we need some libraries such as, 'Pandas', Tensor Flow. These are the main structure on which my main project stands. MNIST data contains about 70,000 images of handwritten digits from 0-9. So, it is a class 10 classification model. This dataset is divided into 2 parts i.e. Training and Test dataset. Image representation as 28*28 matrix where each cell contains grayscale pixel value.

2. LITERATURE SURVEY

EXISTING PROBLEM

1. Handwritten Digit Recognition System involves reception and interpretation of handwritten digits by a machine. Due to variation in shape and orientation of handwritten digits, it is difficult for a machine to interpret handwritten digits. Handwritten digit Recognition has a wide area of research due to its vast applications like automatic bank processing, billing and automatic postal service. In this thesis, an Offline Handwritten Digit Recognition, first part is feature extraction from handwritten images and the second one is classification of feature vector into digits.

   We propose descriptors for handwritten digit recognition based on Histogram of Oriented Gradient (HOG) feature. It is one of the widely used feature vector for object detection in computer vision.

   For classification of features, linear Proximal Support Vector Machine Classifier is proposed. This is a binary class classifier which is further converted to a 10-class classifier by means of One against all algorithm. Due to small training time, PSVM classifier is preferable over standard Support Vector Machine (SVM) Classifier. The handwritten images both for training and testing are taken from MNIST database. The performance of the system is measured in terms of Sensitivity, Accuracy, Positive Predictively and Specificity.

2. The handwritten digit recognition problem becomes one of the most notorious problems in machine" "literacy and computer vision operations. numerous machine literacy ways have been employed to break the handwritten number recognition problem. This paper focuses on Neural Network (NN) approaches. The three most" "popular NN approaches are deep neural network (DNN), deep belief network (DBN) and convolutional neural "network (CNN). In this paper, the three NN approaches are compared and estimated in terms of numerous factors" "similar as delicacy and performance. Recognition delicacy rate and performance, still, isn't the only criterion in the evaluation process, but there are intriguing criteria similar as prosecution time. Random and

   standard dataset of handwritten number have been used for conducting the trials. The results show that among the three NN approaches, DNN is the most accurate algorithm; it has98.08 delicacy rate. still, the prosecution time of DNN is similar with the "other two algorithms.

3. Character recognition plays an important role in the modern world. It can solve more complex problems and makes humans' job easier. An example is handwritten character recognition. 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. Two techniques researched in this paper are Pattern Recognition and Artificial Neural Network (ANN). Both techniques are defined and different methods for each technique is also discussed. Bayesian Decision, and Linear Classification or Discrimination is types of methods for Pattern Recognition. Shape recognition, Chinese Character and Handwritten Digit recognition uses Neural Network to recognize them. Neural Network is used to train and identify written digits. After training and testing, the accuracy rate reached 99%. This accuracy rate is very high.

4. Handwriting recognition has gained a lot of attention in the field of pattern recognition and machine learning due to its application in various fields. Optical Character Recognition (OCR) and Handwritten Character Recognition (HCR) has specific domain to apply. Various techniques have been proposed to for character recognition in handwriting recognition system. Even though, sufficient studies and papers describes the techniques for converting textual content from a paper document into machine readable form. In coming days, character recognition system might serve as a key factor to create a paperless environment by digitizing and processing existing paper documents.

5. Handwritten digit recognition has recently been of very interest among the researchers because of the evolution of various Machine Learning, Deep Learning and Computer Vision algorithms. In this report, we compare the results of some of the most widely used Machine Learning Algorithms like CNN- convolution neural networks and with

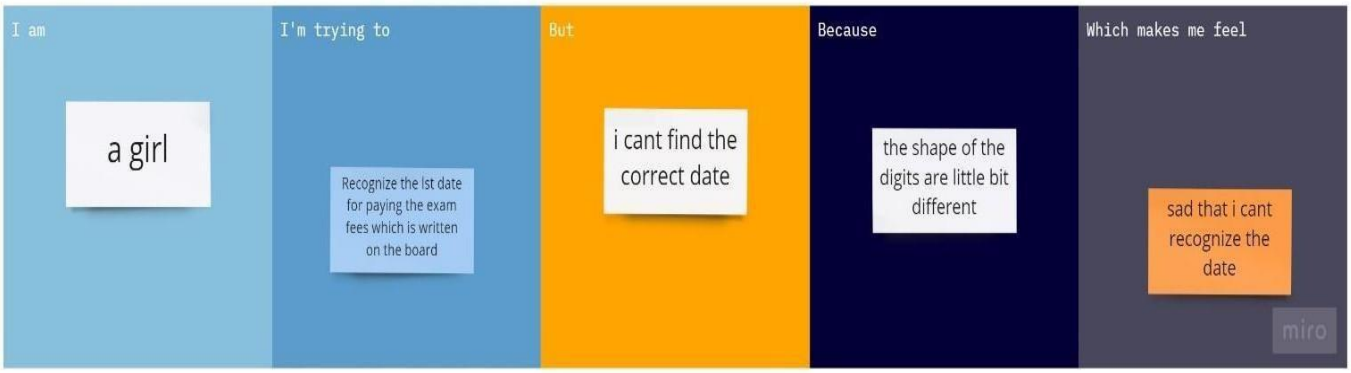
   Deep Learning algorithm like multilayer CNN using Keras with Theano and TensorFlow. MNIST is a dataset which is widely used for handwritten digit recognition. The dataset consists of 60,000 training images and 10,000 test images. The artificial neural networks can all most mimic the human brain and are a key ingredient in image processing field. For example, Convolution Neural networks with back propagation for image processing. The applications where these handwritten digits 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.

REFERENCES

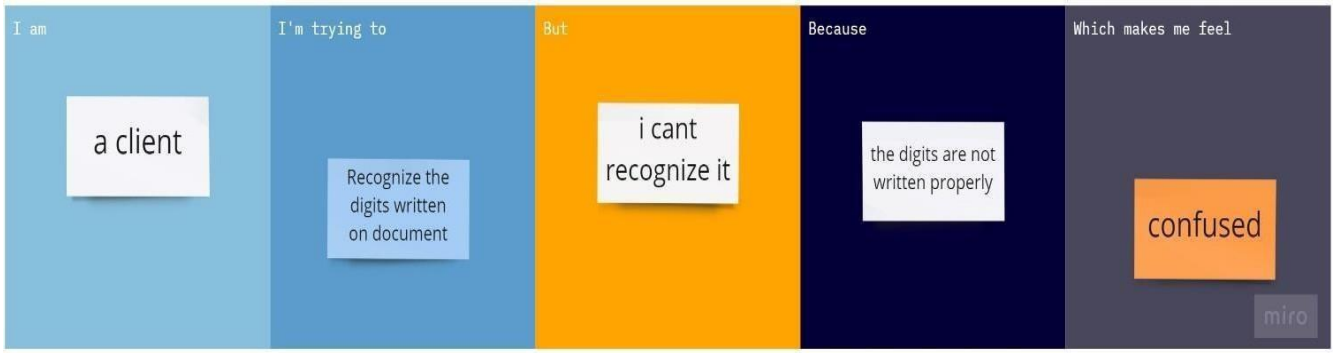
- https://www.ijnrd.org/papers/IJNRD1704024.pdf
- https://www.irjet.net/archives/V9/i6/IRJET-V9I6208.pdf
- https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.228.158&rep=rep1&type=pdf
- http://ijcsit.com/docs/Volume%207/vol7issue1/ijcsit2016070101.pdf
- http://troindia.in/journal/ijcesr/vol6iss6part2/32-36.pdf

PROBLEM STATEMENT DEFINITION

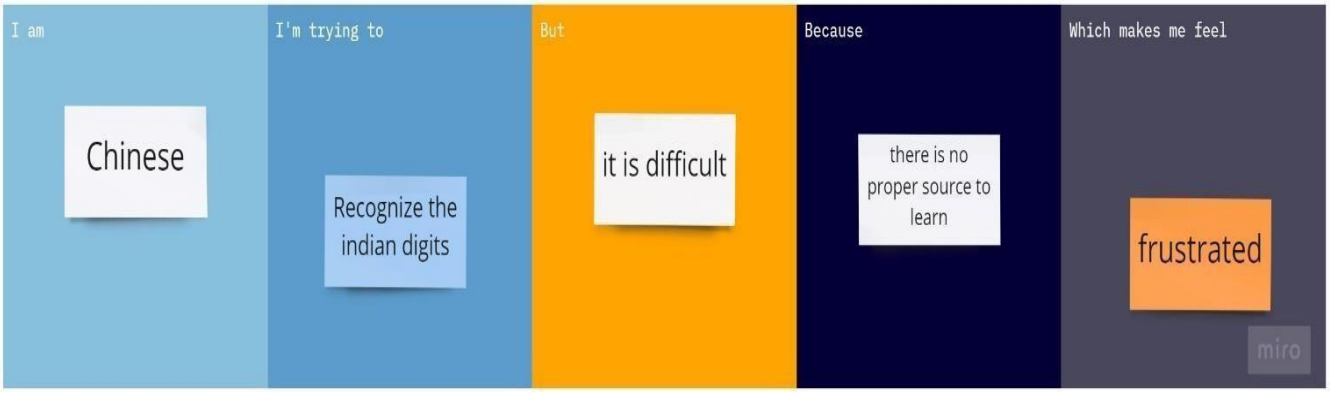
Problem Statement 1



Problem Statement 2



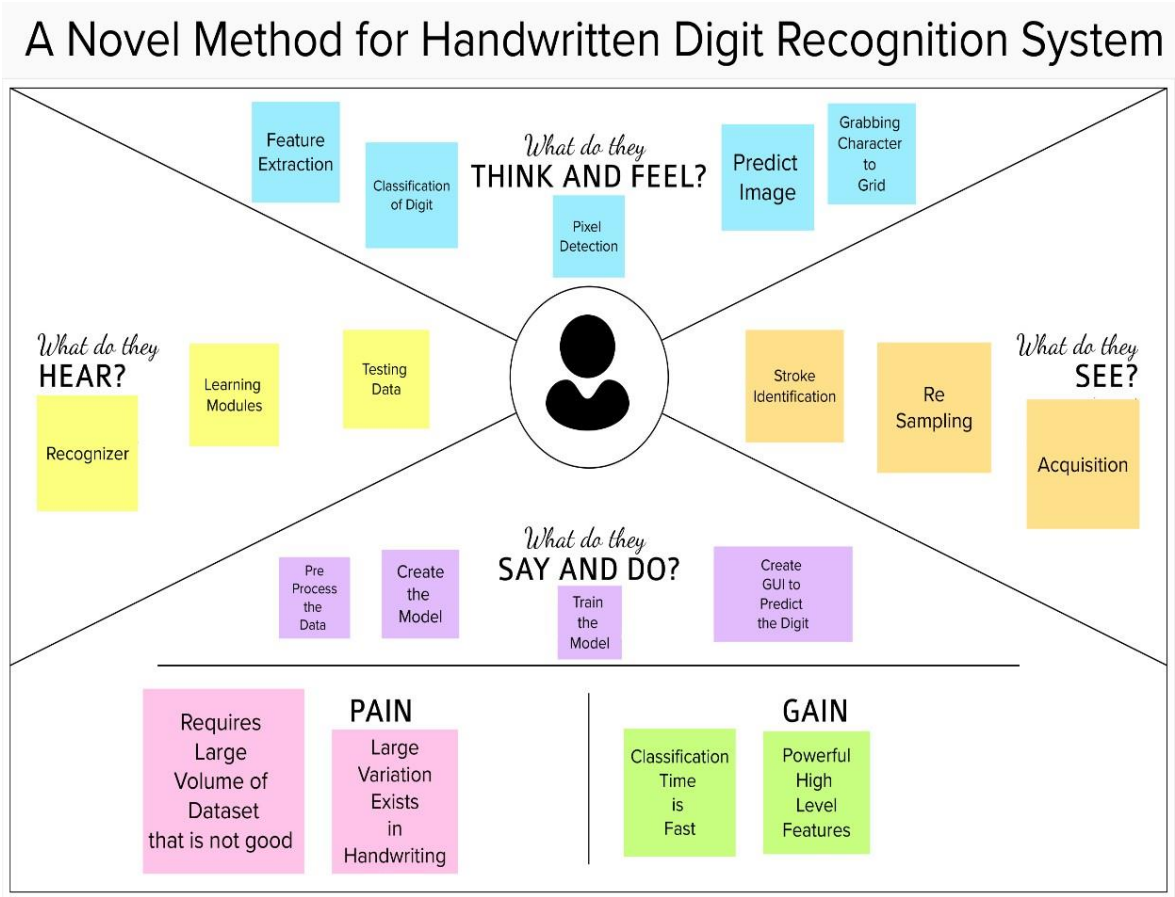
Problem Statement 3



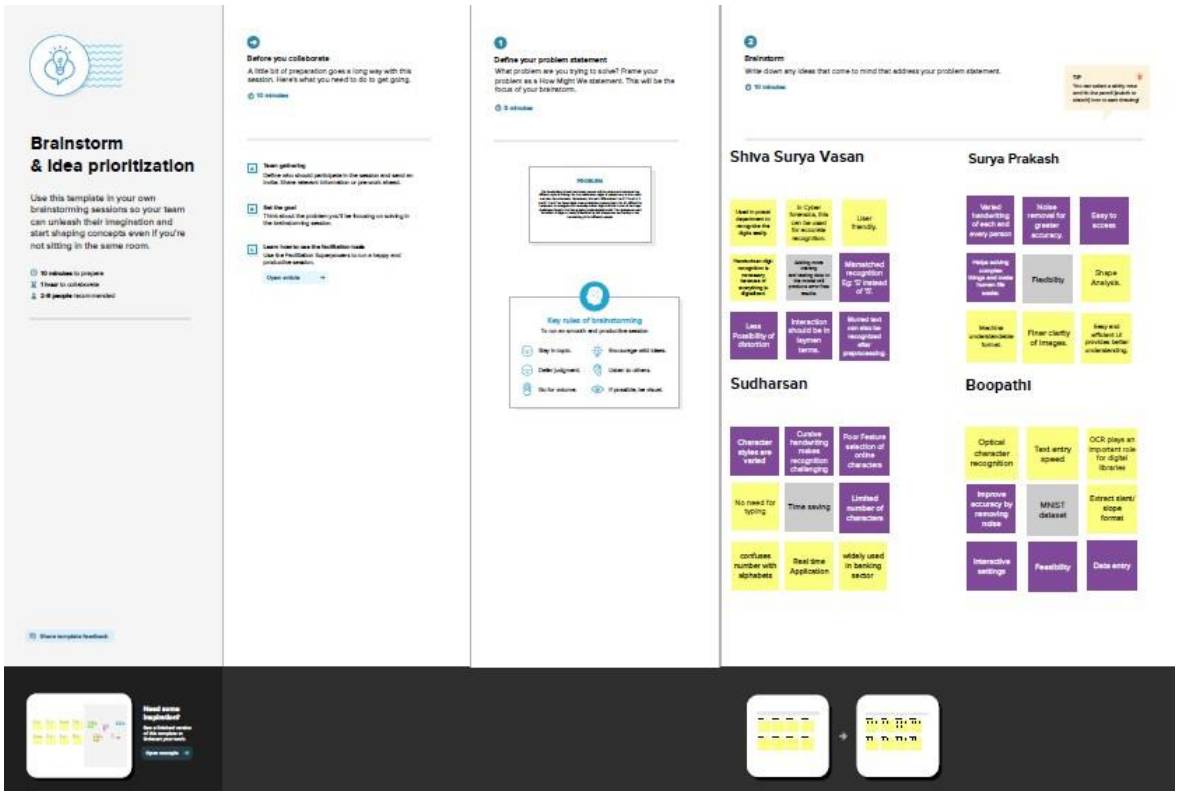
Problem Statement(PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	A girl	Recognize the last date for paying the exam fees which is written on the board	I cant find the correct date	The shape of the digits are little bit different	Sad that I cant recognize the date
PS-2	A client	Recognize the digits written on document	I cant recognize it	The digits are not written properly	confused
PS-3	Chinese	Recognize the Indian digits	It is difficult	There is no proper source to learn	frustrated

3. IDEATION & PROPOSED SOLUTION

EMPATY MAP CANVAS



IDEATION & BRAINSTROMING



PROPOSED SOLUTION

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Computer programmes' ability to detect human-written numbers is known as handwritten digit recognition. Because handwritten figures are not always accurate and can take many various forms and sizes, it is a difficult work for the machine.
2.	Idea / Solution description	Using data from various sources, including images, documents, and touch defences, a computer is able to celebrate the mortal handwritten numbers. It permits users to convert all of their handwritten notes and signatures into text documents in electronic form, using much less physical space than would be needed to store the physical copies of those documents.
3.	Novelty / Uniqueness	Recognize the digits precisely rather than all the characters like OCR.
4.	Social Impact / Customer Satisfaction	The Handwritten Digit Recognizer software was made using artificial intelligence. It approximates the printed word digitally by identifying letters using sophisticated algorithms before producing a digital approximation.
5.	Business Model (Revenue Model)	For efficient traffic control, this technology can be connected with traffic surveillance cameras to read licence plates. Pin-code details can be easily identified and recognised by integrating with the postal system.
6.	Scalability of the Solution	The capacity to recognise numbers in more distracting circumstances. The maximum number of digits that can be recognised is unlimited.

PROBLEM SOLUTION FIT

Project Design Phase-I - Solution Fit Template

ProjectTitle: A Novel Method for Handwritten Digit Recognition System **Team ID:** PNT2022TMID32105

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

- Fintech Industries
- Supply Chain Management
- Medical data Transcriptions
- Scientific and Space Research

2. CUSTOMER CONSTRAINTS CC

- Speed and Accuracy of the system
- Size of the vocabulary
- Spatial layout
- Lack of feedback-based system

3. AVAILABLE SOLUTIONS AS

- Free OCR API
- Human centric data feed

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

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

- To design a system that recognizes a wide range of handwriting scripts
- ML based approach to identify the character quickly and accurately
- Adaptive learning module to learn from its own instances and gets updated

5. PROBLEM ROOT CAUSE RC

- In cases where distinct characters look very similar making it hard for a computer to recognize it accurately.
- Different styles of cursive handwriting is another challenge that requires a support system based on vocabulary

6. BEHAVIOUR BE

- In handwriting recognition (HWR), the module interprets the user's handwritten script into an appropriate digital format s
- Provision for real-time handwritten update in case if the application used by fixed and same users
- Know the market trends and adapt accordingly

Focus on J&P, tap into BE, understand RC

Identify StongTR & EM

7. TRIGGERS TR

- Longer and more in scale, the system understood better
- With its rich vocabulary, it has a support system to autofill the suggestions based on user input

8. EMOTIONS: BEFORE / AFTER EM

- Before: Sometimes character look similar so digit identification process is tedious and time consuming.
- Also, inaccurate sometimes. . After: Using deep learning, identification is faster and relatively more accurate.

9. YOUR SOLUTION SoLN

- Deep learning.
- Intelligent feedback and support system based on neural network making the system more robust

10. CHANNELS of BEHAVIOUR CH

1. ONLINE
   - online handwriting recognition consists of interpreting handwriting represented either by the trajectoryofthepenorbyscanningthescript
2. OFFLINE
   - Offline handwriting recognition consists of interpreting the handwritten scanned document.

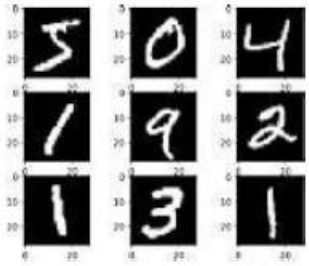
Extract Online and offline CH of BE

4. REQUIREMENT ANALYSIS

Functional Requirements:

Following are the functional requirements of the proposed solution.

ÏR No:	Functional Requirement and description:
FR-1	**Image Data**: Handwritten digit recognition is **the ability of a computer to recognize the human handwritten digits from different sources like images, papers, touch screens, etc., and classify them into 10 predefined classes (0-9).** this has been a topic of boundless-research in the field of deep learning.

FR-2	**Website:** Web hosting **makes the files that comprise a website (code, images, etc.) available forviewing online. Every websiteyou've ever visited is hostedonaserver. The** amount of space allocated on a server to a website depends on the type of hosting. the main types of hosting are shared, dedicated, VPS..
FR-3	**Digit_Classifiei_Model: Use the MNISt database of handwritten digits to train a convolutional network to predict the digit given an image. First obtain the** training and validation data.
FR-4	**MNISt dataset: the MNIST dataset is an acronym that stands for the Modified National Institute ofStandards and Technologydataset. It is a dataset of 60,000 small** square 28×28 pixel grayscale images of handwritten single digits between 0and 9. 

4.2 NON FUNCTIONAL REQUIREMENTS

Non-functional Requirements:

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

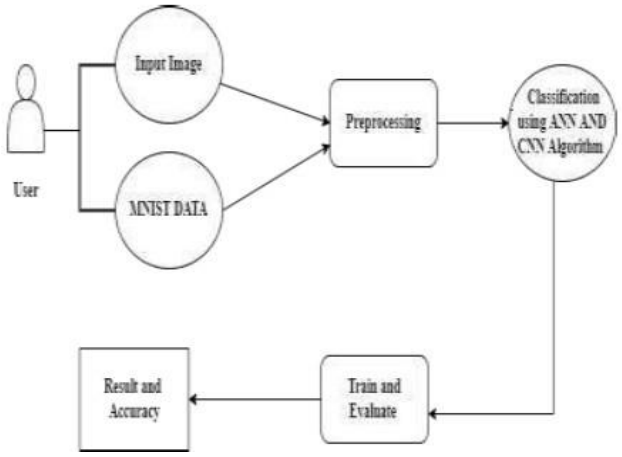
NFRNo.	Non-FunctionalRequirement
NFR-1	**Usability:** Handwritten character recognition is one of the practically important issues in pattern recognition applications. the applications of digit recognition include in **postal mail sorting, bank check processing, form data entry**, etc.
NFR-2	**Reliability:** 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.
NFR-3	**Performance:** the neural network uses **the examples to automatically inferrules 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 lees, Random Joists, etc.**
NFR-4	**Accuracy:** **Optical Character Recognition** (OCR) technology provides **higher than 99% accuracy** with typed characters in high- quality images. However, the diversity in human writing types, spacing differences, and inequalities of handwriting causesless accurate character recognition.

5. PROJECT DESIGN

DataFlowDiagrams:

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

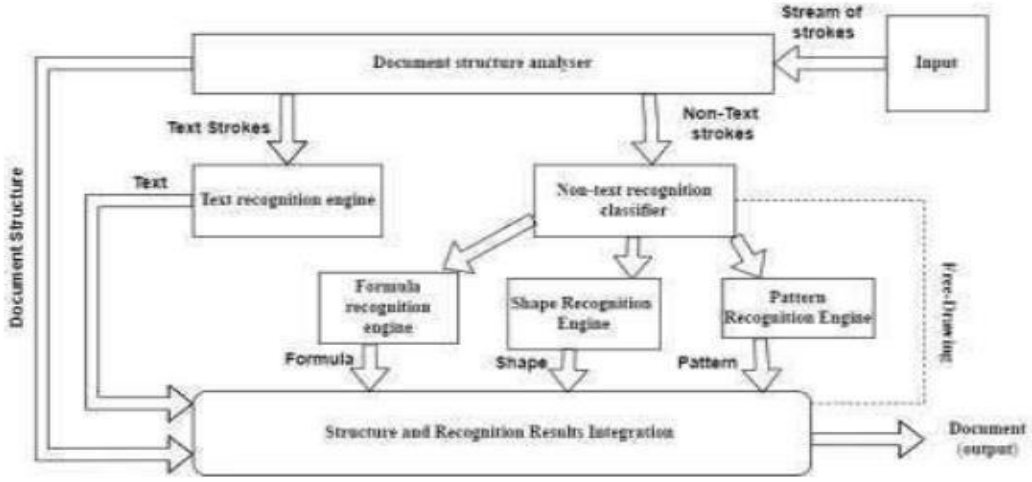
Example:(Simplified)FLOW



User Stories

Use the below template to list all the user stories for theproduct.

Example:DFDLevel0(IndustryStandard)

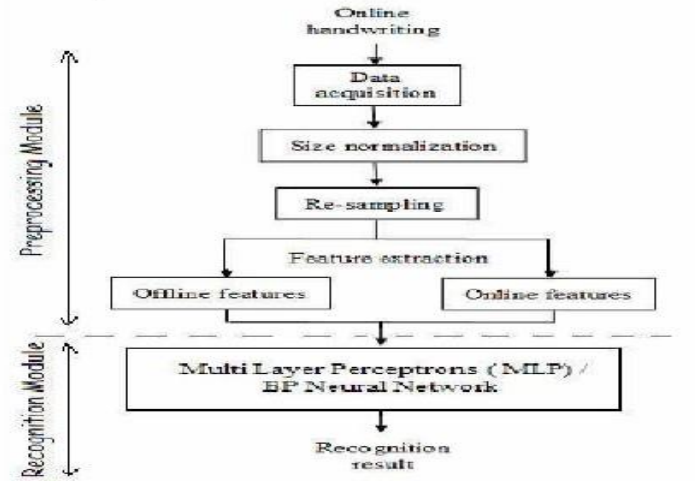


SOLUTION & TECHNICAL ARCHITECTURE

Project Description :

Handwritten digit recognition is one of the important problems in computer vision these days. There is a great interest in this field because of many potential applications, most importantly where a large number of documents must be dealt such as post mail sorting, bank cheque analysis, handwritten form processing etc. So a system should be designed in such a way that it is capable of reading handwritten digits and providing appropriate results. We propose a solution on neural network approaches to recognize handwritten digits.

Technical Architecture :



Working :

Dataset used :

MNIST ("Modified National Institute of Standards and Technology") is the "Hello World" dataset of computer vision. Since its release in 1999, this classic dataset of handwritten images has served as the basis for benchmarking classification algorithms. As new machine learning techniques emerge, MNIST remains a reliable resource for researchers and learners alike.

In this competition, we aim to correctly identify digits from a dataset of tens of thousands of handwritten images. Kaggle has curated a set of tutorial-style kernels which cover everything from regression to neural networks. They hope to encourage us to experiment with different algorithms to learn first-hand what works well and how techniques compare.

Dataset description :

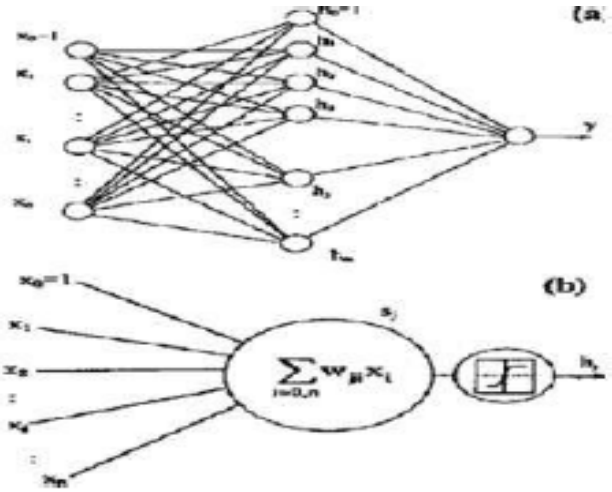
For this competition, we will be using Keras (with TensorFlow as our backend) as the main package to create a simple neural network to predict, as accurately as we can, digits from handwritten images. In particular, we will be calling the Functional Model API of Keras, and creating a 4-layered and 5-layered neural network.

The MNIST Handwritten Digit Recognition Dataset contains 60,000 training and 10,000 testing labeled handwritten digit pictures. Each picture is 28 pixels in height and 28 pixels wide, for a total of 784 (28×28)pixels. Each pixel has a single pixel value associated with it. It indicates how bright or dark that pixel is (larger numbers indicate darker pixel). This pixel value is an integer ranging from 0 to 255.

Procedure :

- Install the latest TensorFlow library.
- Prepare the dataset for the model.
- Develop Single Layer Perceptron model for classifying the handwritten digits. Plot the change in accuracy per epochs.
- Evaluate the model on the testing data.
- Analyze the model summary.
- Add a hidden layer to the model to make it a Multi-Layer Perceptron.
- Add Dropout to prevent overfitting and check its effect on accuracy.
- Increasing the number of Hidden Layer neurons and checking its effect on accuracy.
- Use different optimizers and check its effect on accuracy.
- Increase the hidden layers and check its effect on accuracy.
- Manipulate the batch size and epochs and check its effect on accuracy.

Handwritten digit recognition using MNIST dataset is a major project made with the help of Neural Network. It basically detects the scanned images of



Working :

Neural Networks receive an input and transform it through a series of hidden layers. Each hidden layer is made up of a set of neurons, where each neuron is fully connected to all neurons in the previous layer. Neurons in a single layer function completely independently. The last fully connected layer is called the "output layer".

Tensor flow :

TensorFlow is an open-source machine learning library for research and production. TensorFlow offers APIs for beginners and experts to develop for desktop, mobile, web, and cloud. See the sections below to get started. By scanning the numerical digit and converting it into png format using the python3 command in the terminal we can get text output and sound output.

Feature extraction :

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

Classification :

Convolutional neural network that is very popular for computer vision tasks like image classification, object detection, image segmentation and a lot more. Image classification is one of the most needed techniques in today's era, it is used in various domains like healthcare, business, and a lot more.

Result :

We do not consider our results to be flawless, as with any study or project undertaken in the field of machine learning and image processing. There is always opportunity for improvement in your methods because machine learning is a topic that is continually developing; there will always be a fresh new idea that solves a given problem more effectively. Three models were used to test the application: Multi-Layer Perceptron (MLP), Convolution NeuralNetwork, and (CNN). We obtain a different classifier accuracy with each model, indicating which is superior.

Technical Architecture:

The architectural diagram of the model is as below and the Technology used is shown in table1 & table 2

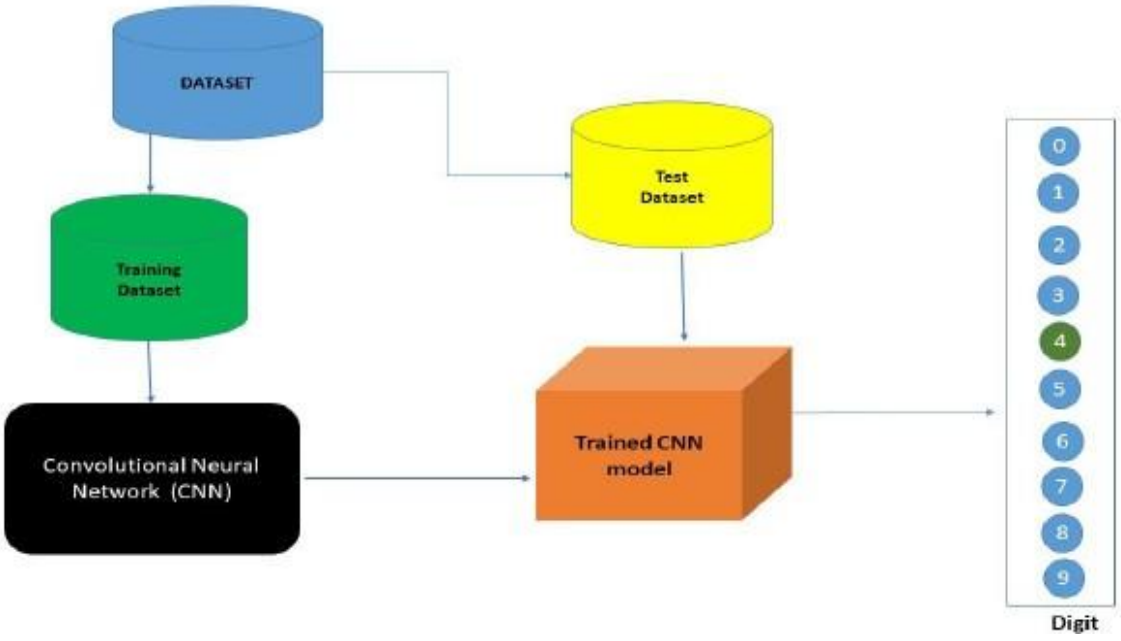


Table-1: Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g., Mobile Application	HTML, CSS, JavaScript / Angular JS / Node Red.
2.	Application Logic-1	Logic for a process in the application	Java / Python
3.	Application Logic-2	Logic for a process in the application	IBM Watson STT service
4.	Application Logic-3	Logic for a process in the application	IBM Watson Assistant
5.	Database	Data Type, Configurations etc.	MySQL, NoSQL, etc.
6.	Cloud Database	Database Service on AI	IBM DB2.
7.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem
8.	External API-1	Purpose of External API used in the application	IBM Weather API, etc.
9.	IoT Model	Purpose of AI Model is for integrating the sensors with a user interface.	IBM AI Platform
10.	Infrastructure (Server / AI)	Application Deployment on Local System / AI Local Server Configuration AI Server Configuration	Local, Kubernetes, etc.

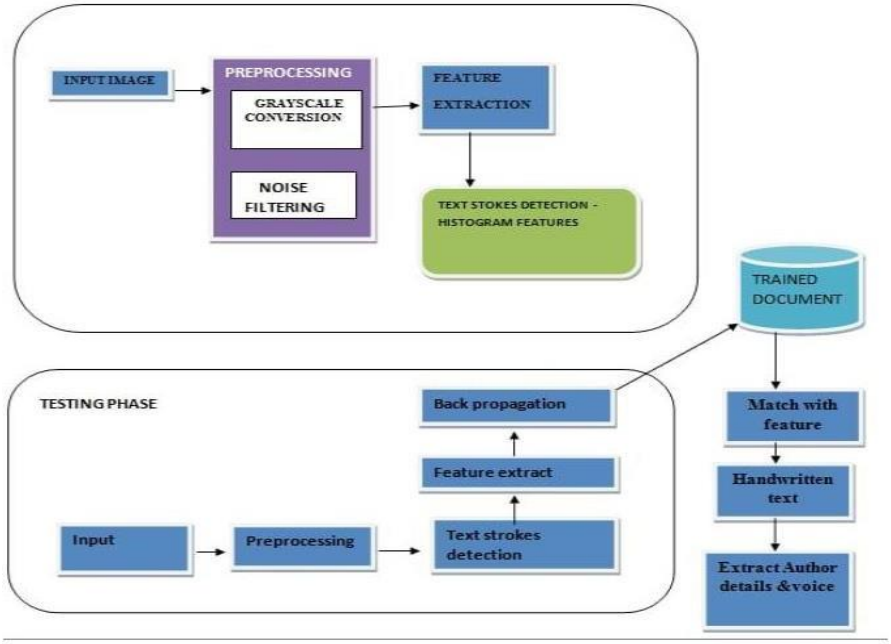
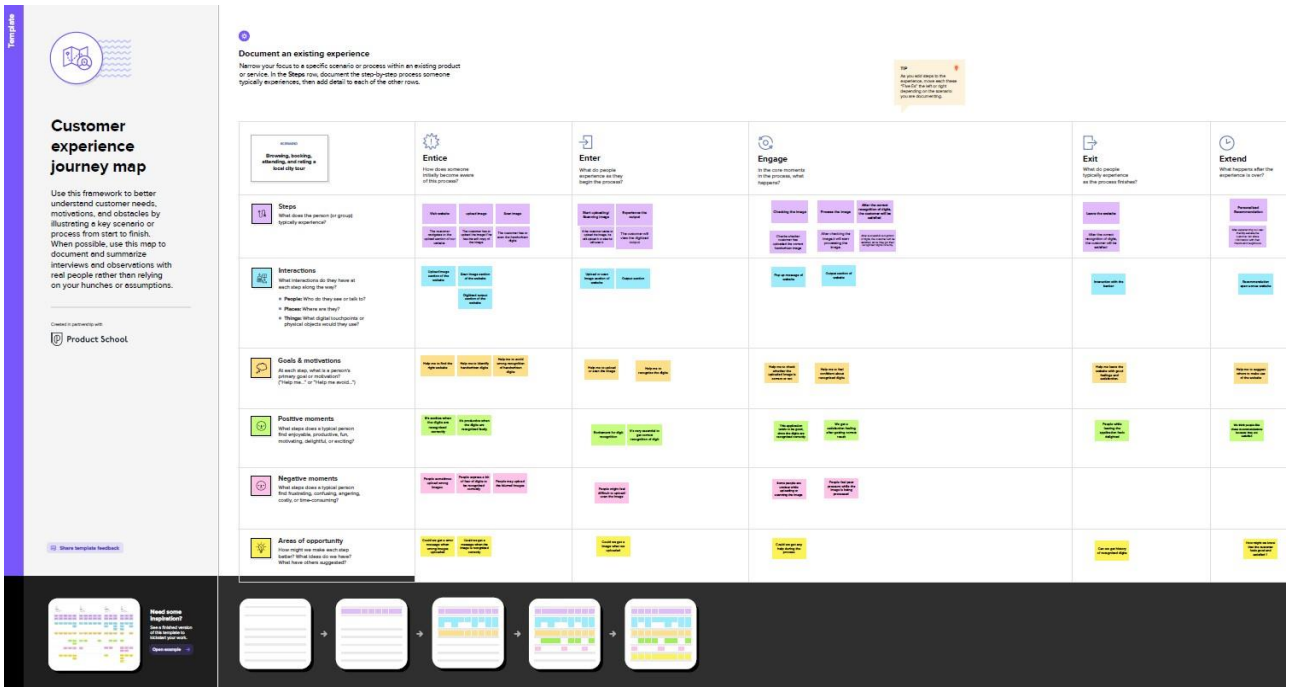


FIG. 1. BLOCK DIAGRAM

USER STORIES



6. PROJECT PLANNING & SCHEDULING

SPRINT PLANNING & ESTIMATION

Milestone and Activity List:

TITLE	DESCRIPTION	DATE
Literature Survey & Information Gathering	Literature survey on the selected project & gathering information by referring the, technical papers, research publications etc.	17 SEPTEMBER 2022
Prepare Empathy Map	Prepare Empathy Map Canvas to capture the user Pains & Gains, Prepare list of problem statements	17 SEPTEMBER 2022
Ideation	List the by organizing the brainstorming session and prioritize the top 3 ideas based on the feasibility & importance.	24 SEPTEMBER 2022
Proposed Solution	Prepare the proposed solution d ocument, which includes the novelty, feasibility of idea, business model, social impact, scalability of solution, etc.	24 SEPTEMBER 2022
Problem Solution Fit	Prepare problem - solution fit document.	8 OCTOBER 2022

Solution Architecture	Prepare solution architecture document.	8 OCTOBER 2022
Customer Journey	Prepare the customer journey maps to understand the user interactions & experiences with the application	21 OCTOBER 2022
Data Flow Diagrams	Draw the data flow diagrams and submit for review.	21 OCTOBER 2022
Technology Architecture	Architecture diagram.	21 OCTOBER 2022
Prepare Milestone & Activity List	Prepare the milestones & activity list of the project.	2 NOVEMBER 2022
Project Development - Delivery of Sprint1, 2, 3 & 4	Develop & submit the developed code by testing it.	IN PROGRESS....

SPRINT DELIVERY SCHEDULE

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data Collection	USN-1	Collect Dataset .	9	High	Shiva Surya Vasan
Sprint-1		USN-2	Reshape the data and apply one hot encoding	8	Medium	Surya Prakash, Sudharson, Boopathi
Sprint-2	Model Building	USN-3	Import the required libraries, add the necessary layers and compile the model	10	High	Shiva Surya Vasan Surya Prakash
Sprint-2		USN-4	Training the image classification model using CNN	7	Medium	Boopathi, Sudharson
Sprint-3	Training and Testing	USN-5	Building Python code and run the application	9	High	Shiva Surya Vasan
Sprint-4	Implementation of the application and deployment on cloud	USN-6	Training the model on IBM cloud.	8	Medium	Surya Prakash, Boopathi

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	10	6 Days	24 Oct 2022	29 Oct 2022	8	29 Oct 2022
Sprint-2	10	6 Days	31 Oct 2022	04 Nov 2022	5	04 Nov 2022
Sprint-3	10	6 Days	07 Nov 2022	11 Nov 2022	7	11 Nov 2022
Sprint-4	10	6 Days	14 Nov 2022	18 Nov 2022	5	18 Nov 2022

Velocity:

$$AV = \frac{sprint\ duration}{velocity}$$

AV = 6/10 = 0.6

Burndown chart:

7. CODING & SOLUTIONING

FEATURE 1

- Would you like to know how to measure the accuracy of your Machine Learning model? Do you wonder if high accuracy automatically means good performance? Are you trying to measure accuracy on multiclass and multilabel problems?

FEATURE 2

- Accuracy is perhaps the best-known Machine Learning model validation method used in classification problems. One reason for its popularity is its relative simplicity. It is easy to understand and easy to implement. Accuracy is a good metric to assess model performance for simple cases.

- However, modeling problems are rarely simple. You need to work with imbalanced datasets or in a multiclass or multilabel setting. A high accuracy might not even be your goal.

- As you solve more complex Machine Learning problems, calculating and using accuracy becomes less obvious and requires extra consideration.

DATABASE SCHEMA (IF APPLICABLE)

- For this reason, it is important to understand what accuracy is, how to calculate it, and what its weaknesses are in different Machine Learning contexts.

- Accuracy = TP + TN / TP + FP + TN + FN.

- Accuracy = Number of Correct Predictions / Total No Of Predictions

8. TESTING

TEST CASE

Import necessary libraries

```
import requests
import numpy as np
from PIL import Image, ImageOps
import matplotlib.pyplot as plt
```

Input pre-processing

```
img = Image.open(f"../sample/sample 1.png").convert("L")
img = ImageOps.invert(img)
img = img.resize((28, 28))
img_arr = np.array(img)
img_arr = img_arr / 255.0
img_arr = img_arr.reshape(28, 28, 1)

img2 = Image.open(f"../sample/sample 2.png").convert("L")
img2 = ImageOps.invert(img2)
img2 = img2.resize((28, 28))
img2_arr = np.array(img2)
img2_arr = img2_arr / 255.0
img2_arr = img2_arr.reshape(28, 28, 1)

img3 = Image.open(f"../sample/sample 3.png").convert("L")
img3 = ImageOps.invert(img3)
img3 = img3.resize((28, 28))
img3_arr = np.array(img3)
img3_arr = img3_arr / 255.0
img3_arr = img3_arr.reshape(28, 28, 1)
```

Display results

```
plt.imshow(plt.imread("../sample/sample 1.png"))
plt.axis('off')
plt.show()
print("Result: ", response_scoring.json()['predictions'][0]['values'][0][1])
```

```
Result:  2
```

```
plt.imshow(plt.imread("../sample/sample 2.png"))
plt.axis('off')
plt.show()
print("Result: ", response_scoring.json()['predictions'][0]['values'][1][1])
```

```
Result:  7
```

USER ACCEPTANCE TESTING

```
#Importing the Keras libraries and packages
from tensorflow.keras.models import load_model
from PIL import Image #used for manipulating image uploaded by the user.
import numpy as np #used for numerrical analysis
model = load_model(r'C:\Users\Kishore Kumar\models\mnistCNN.h5')

for index in range(9):
    img = Image.open("C://Users//Kishore Kumar//data//"+str(index)+'.png').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
    # Predicting the Test set results
    y_pred = model.predict(im2arr) #predicting the results
    print(y_pred)
```

```
1/1 [==============================] - 0s 53ms/step
[[1.0000000e+00 1.0641970e-17 1.7592266e-12 3.6711346e-19 4.4121307e-19
  1.8262852e-14 6.1975113e-12 1.1711813e-14 5.0637394e-10 2.7724546e-12]]
1/1 [==============================] - 0s 17ms/step
[[2.4917088e-10 9.9976009e-01 1.3111265e-06 2.1737113e-08 9.0895455e-08
  9.3907556e-07 2.3508941e-04 5.1348117e-09 2.4486098e-06 4.5918522e-10]]
1/1 [==============================] - 0s 16ms/step
[[1.55988611e-10 1.48565285e-08 9.99990106e-01 3.48837420e-06
  1.79694652e-13 3.87173810e-16 3.41927613e-15 1.10914505e-06
  5.28426381e-06 3.37601336e-11]]
1/1 [==============================] - 0s 17ms/step
[[6.1019656e-09 1.2880452e-10 1.7472412e-05 9.9981397e-01 8.8599359e-13
  6.5538508e-05 1.7593281e-09 1.4509633e-11 3.5314154e-11 1.0301937e-04]]
1/1 [==============================] - 0s 16ms/step
[[1.1062213e-11 8.2784821e-13 1.9456865e-11 1.4016800e-10 9.9999964e-01
  6.9351846e-10 1.0945117e-12 1.2004395e-07 6.6777318e-08 1.5515947e-07]]
1/1 [==============================] - 0s 17ms/step
[[5.61475800e-16 1.24912224e-18 2.68544331e-16 1.61019939e-12
  1.23687994e-17 1.00000000e+00 8.35287810e-19 2.08621216e-18
  6.42477337e-14 7.26153133e-13]]
1/1 [==============================] - 0s 15ms/step
[[1.4059653e-09 4.3012458e-14 6.7409906e-15 1.1206856e-16 3.6507321e-11
  1.9055557e-11 1.0000000e+00 8.6574725e-18 2.1831118e-13 5.7935814e-14]]
1/1 [==============================] - 0s 17ms/step
[[9.10107616e-08 3.60086894e-08 7.62391539e-11 1.01688796e-10
  5.61130022e-08 4.19142268e-08 1.23757765e-14 9.99990225e-01
  2.41333231e-10 9.50253707e-06]]
1/1 [==============================] - 0s 20ms/step
[[2.8380433e-05 1.3553810e-06 1.6209298e-05 1.6655436e-06 1.6096469e-04
  2.3872168e-08 1.9169308e-09 3.6023029e-09 9.9918765e-01 6.0375809e-04]]
```

```
import numpy as np
print(np.argmax(y_pred, axis=1)) #printing our Labels from first 4 images
```

```
[8]
```

9. RESULTS

PERFORMANCE METRICS

Observing The Metrics

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

```
Metrics (Test loss &Test Accuracy) :
[0.09719487279653549, 0.98089998960495]
```

Test The Model

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

```
1/1 [==============================] - 1s 636ms/step
[[5.1234290e-12 2.9132625e-12 4.7156525e-11 1.9892369e-07 1.4272153e-02
  4.1180425e-05 6.8472568e-15 7.2068692e-04 6.3077291e-06 9.8495942e-01]]
```

```
import numpy as np
print(np.argmax(prediction, axis=1)) #printing our Labels from first 4 images
np.argmax(y_test[5000:5001]) #printing the actual labels
```

```
[9]
3
```

10. ADVANTAGES & DISADVANTAGES

❖ ADVANTAGE

- Information of images can be readable with high degree of accuracy. Flatbed scanners are very accurate and may produce reasonably top quality results.
- Processing of DL information is fast. Large quantities of text are often input quickly.
- It takes less time to convert within the Digital form Advanced version can even Recreate.

❖ DISADVANTAGE

- Not 100% accurate, there are likely to be some mistakes made during the method.

11. CONCLUSION

- The proposed handwritten recognition system recognizes the letters with acceptable accuracy. Extraction of small and capital letters is done separately.
- In future the small letters and capital letters can be integrated. There is a Scope for Recognition of digits and Special symbols. The information of the document can be edited more conveniently and can reuse the edited information as and when required.
- The grid infrastructure used in the implementation of Optical Character Recognition system can be efficiently used to speed up the translation of image based documents into structure documents that are currently easy to discover, search and process.
- There is no out of box software for this kind of documents which can read handwriting automatically without human validation and training.

12. FUTURE SCOPE

- Font Independent OCR system could be developed by considering the multiple font style in use. Our approach is very much useful for the font independent case. Because, for font or character size, it finds the string and the strings are parsed to recognize the character. Once character is identified, the corresponding character could be ejected through an efficient editor. Efforts have been taken
- To develop a compatible editor for Tamil and English. There is heavy demand for an OCR system which recognizes cursive scripts and manuscripts like Palm Leaves. This actually avoids keyboard typing.
- The most required application today is Speech recognition. The recognized Printed or Handwritten character could be recorded and through a voice synthesizer speech output could be generated. This would help the blind to send and receive information.

13. APPENDIX

SOURCE CODE

app.py

```
from flask import Flask,render_template,request
from recognizer import recognize

app=Flask(_name_)

@app.route('/')
def main():
return render_template("home.html")


@app.route('/predict',methods=['POST'])
def predict():
if request.method=='POST':
image = request.files.get('photo', '')
best, others, img_name = recognize(image)
return render_template("predict.html", best=best, others=others, img_name=img_name)

if _name=="main_":
app.run()
```

Recognizer.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 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
```

GITHUB

- https://github.com/IBM-EPBL/IBM-Project-9957-1659085393

PROJECT DEMO LINK

- https://drive.google.com/file/d/1cqZ1IsQgE6JChiEjowgI98rUeUXgToff/view?usp=sharing