PROJECT REPORT
A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM
TEAM ID:PNT2022TMID37085

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Project Overview

Machine learning and deep learning plays an important role in computer technology and artificial intelligence. With the use of deep learning and machine learning, human effort can be reduced in recognizing, learning, predictions and many more areas. This article presents recognizing the handwritten digits (0 to 9) from the famous MNIST dataset, comparing classifiers like KNN, PSVM, NN and convolution neural network on basis of performance, accuracy, time, sensitivity, positive productivity, and specificity with using different parameters with the classifiers. To make machines more intelligent, the developers are diving into machine learning and deep learning techniques. A human learns to perform a task by practicing and repeating it again and again so that it memorizes how to perform the tasks. Then the neurons in his brain automatically trigger and they can quickly perform the task they have learned. Deep learning is also very similar to this. It uses different types of neural network architectures for different types of problems.

1.2 Purpose

The handwritten digit recognition is the ability of computers to recognize human handwritten digits. It is a hard task for the machine because handwritten digits are not perfect and can be made withmany different flavors. The handwritten digit recognition is the solution to this problem which uses the image of a digit and recognizes the digit present in the image.

2. LITERATURE SURVEY

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 analysis- by-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.

2.1 References

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.

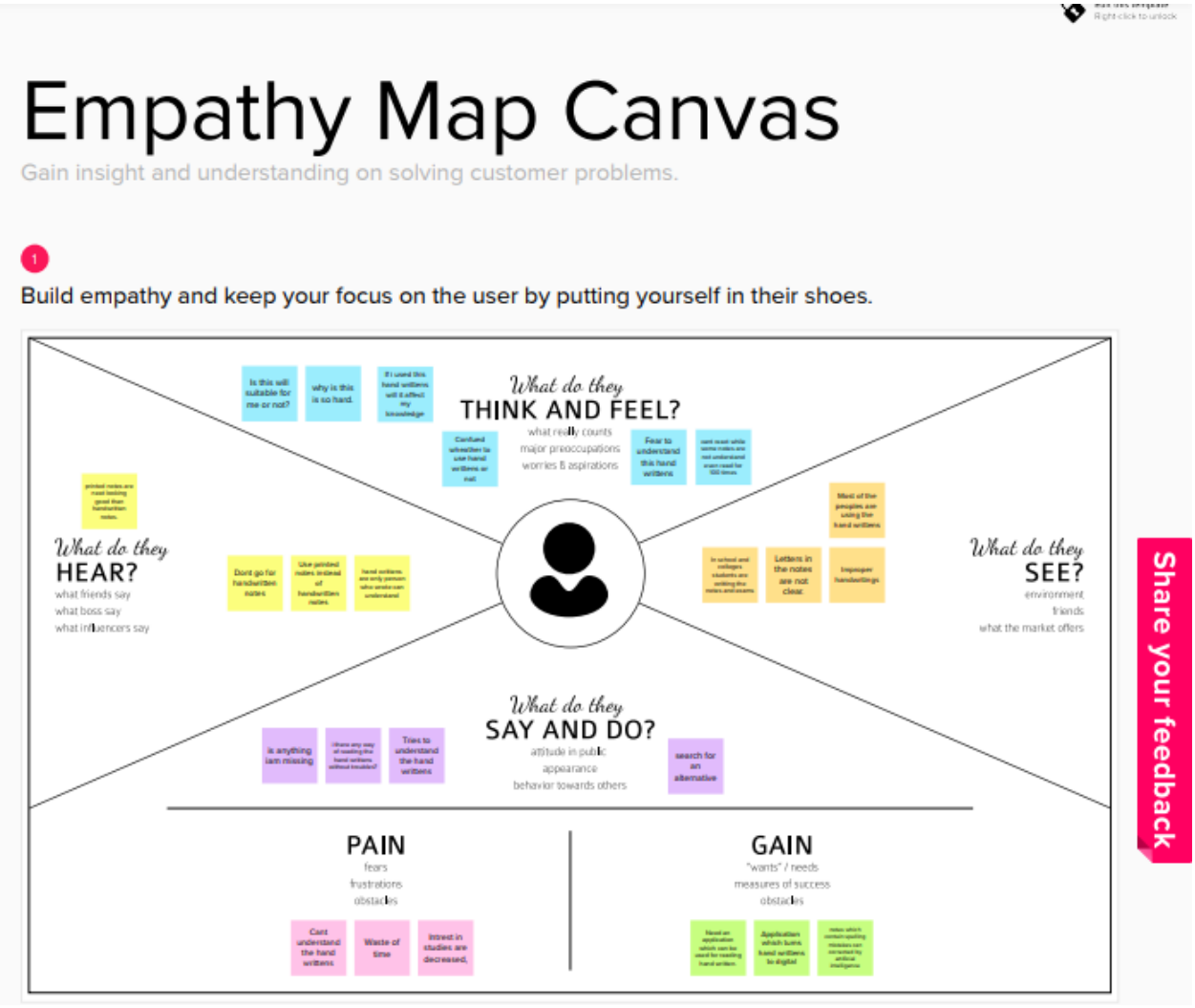
2.2 Problem Statement Definition

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

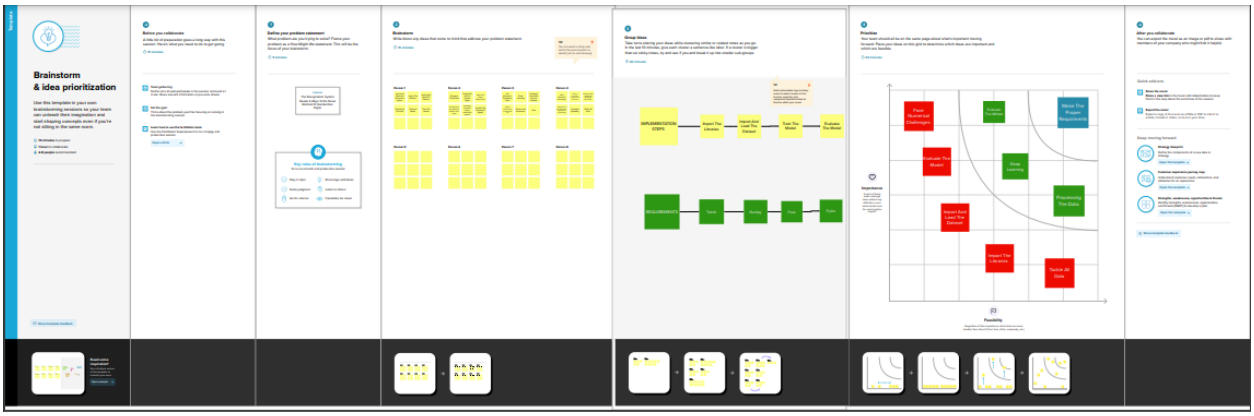
An empathy map is **a collaborative tool teams can use to gain a deeper insight into their customers**. Much like a user persona, an empathy map can represent a group of users, such as a customer segment. The empathy map was originally created by Dave Gray and has gained much popularity within the agile community.



3.2 Ideation and Brainstorming

Ideation is **the process of forming ideas from conception to implementation, most often in a business setting**. Ideation is expressed via graphical, written, or verbal methods, and arises from past or present knowledge, influences, opinions, experiences, and personal convictions.

Ideation is often closely related to the practice of brainstorming, a specific technique that is utilized to generate new ideas. A principal difference between ideation and brainstorming is that **ideation is commonly more thought of as being an individual pursuit, while brainstorming is almost always a group activity**.



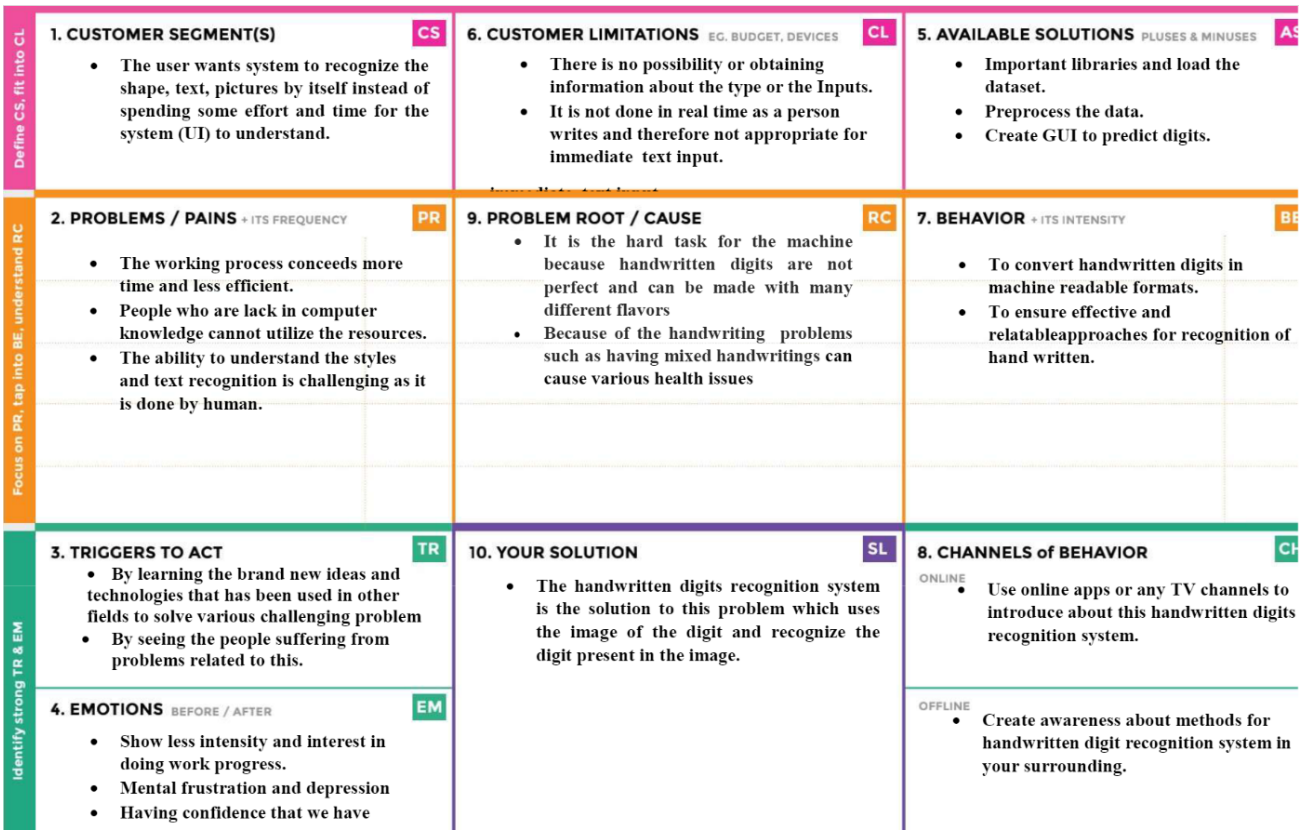
3.3 Proposed Solution

Proposed Solution Template: Project team shall fill the following information in proposed solution template

S.No.	Parameter	Description
1	Problem Statement (Problem to be solved)	● In normal method the time taken for the process will be very high ● The handwritten digit recognition is the capability of computer applications to recognize the human written digits
2	Idea / Solution description	● This idea will able to recognize the hand written and printed image to words easily ● It is the capability of the computer to identity and ● understand handwritten digits or character automatically.
3	Novelty / Uniqueness	● One of the most arduous and captivating domains under image processing in handwritten characters recognition. ● The main objectives of this work is to ensure effective and reliable approach for recognition of handwritten digits and make banking operation easier
4	Social Impact / Customer Satisfaction	● Recently handwritten digit recognition becomes vital scope and it is appealing many researcher because of its using in variety of machine learning and computer vision applications ● 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 a writing style.
5	Business Model (Revenue Model)	● It can be developed with minimum cost and provide high effective process at less time
6	Scalability of the Solution	● This system can be developed further using advanced sensor and can be upgraded wisely and can be stored in large scale sensor medium . ● It can be further used to recognize Arabic digits are more challenging than English pattern

3.4 Problem Solution Fit:

The Problem-Solution Fit simply means that **you have found a problem with your customer and that the solution you have realized for it actually solves the customer's problem**.



4. REQUIREMENT ANALYSIS

4.1 Functional Requirements

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Pre-processing	The role of the pre-processing step is it performs various tasks on the input image. It basically upgrades the image by making it reasonable for segmentation.
FR-2	Segmentation	In this step an edge detection technique is being used for segmentation of dataset images.
FR-3	Feature	Extraction In the feature extraction stage redundancy from the data is removed.
FR-4	Classification and Recognition	feature vectors are taken as an individual input to each of the classifiers

4.2 Non-Functional Requirements

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

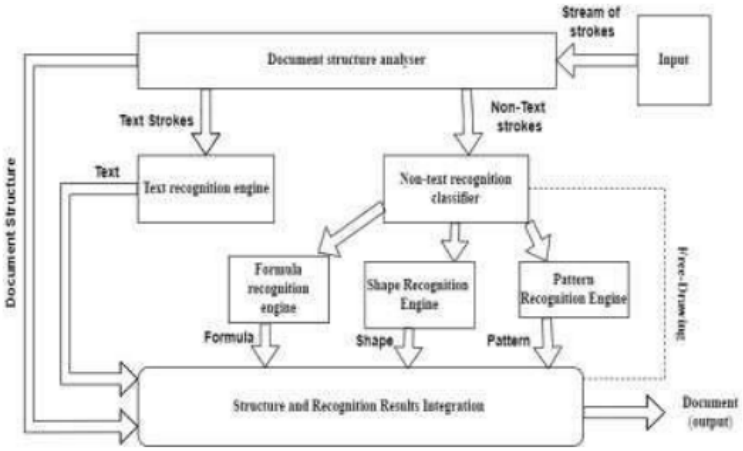
FR No.	Non-Functional Requirement	Description
NFR-1	Usability	The usability of this Handwritten digit recognition system is to identify and understand hand written digits or characters automatically.
NFR-2	Security	The security will be high because since the handwritings has been recognized one cannot upload copy of others document
NFR-3	Reliability	The MNIST data set is widly used for this recognition process and it has 70000 handwritten digits.since it is reliable
NFR-4	Performance	The performance of this web application is high because we use Artificial neural networks to train these images and build a deep learning model.
NFR-5	Availability	Since it is web application one can use it easily and the availability is good ,can be used in laptop, mobile, desktop etc
NFR-6	Scalability	Even though the count of handwritings increased it wont be slow because we are using MNIST data set as it used for recognition process and it has 70000 handwritten digits, so it is very scalable.

5.PROJECT DESIGN

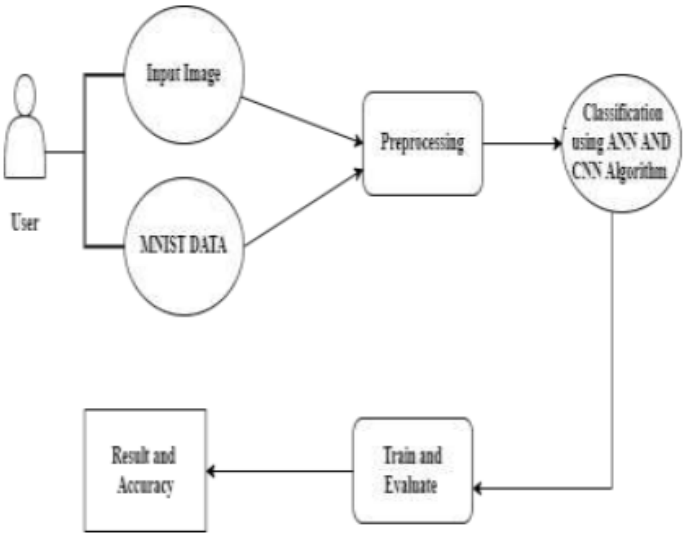
5.1. Data Flow Diagrams

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

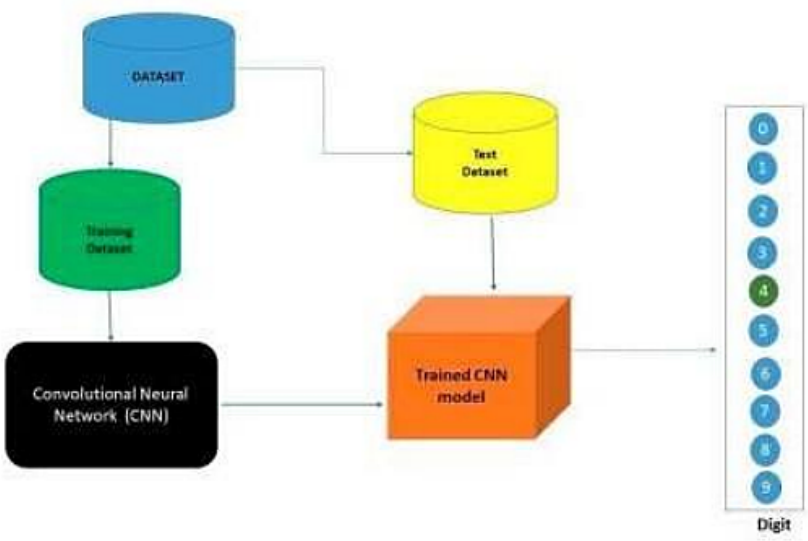
Example: DFD Level 0 (Industry Standard)



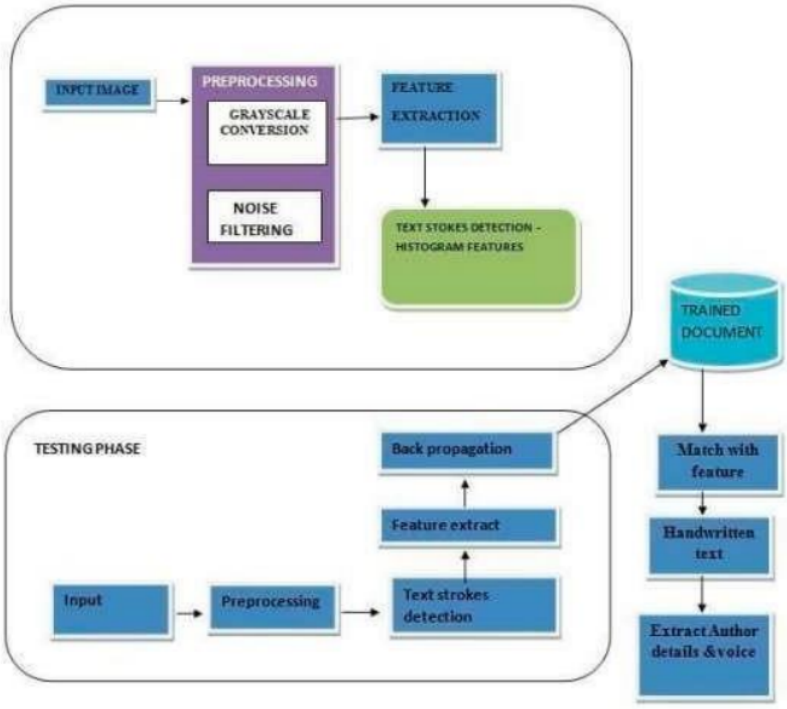
Example: (Simplified) FLOW



5.2 Solution & Technical Architecture



SOLUTION ARCHITECTURE



TECHNICAL ARCHITECTURE

5.3 User Stories

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

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Home	USN-1	As a user, I can viewthe guide andawareness to use this application.	I can view the awareness to use this application and its limitations.	Low	Sprint-1
		USN-2	As a user, I'm allowed to view the guided video to use the interface of this application.	I can gain knowledge to use this application by a practical method.	Low	Sprint-1
		USN-3	As a user, I can readthe instructions to usethis application.	I can read instructions alsoto use it in a user-friendly method.	Low	Sprint-2
	Recognize	USN-4	As a user, In this prediction page I get tochoosethe image.	I can choose the image from ourlocal system andpredict the output.	High	Sprint-2
	Predict	USN-6	As a user, I'm Allowed to upload andchoose the image to be uploaded	I can upload and choose the imagefrom the systemstorage and also in any virtual storage.	Medium	Sprint-3

		USN-7	As a user, I will train and testthe input to getthe maximum accuracy of output.	I can able to train and test the application until it gets maximum accuracy of theresult.	High	Sprint-4
		USN-8	As a user, I can access the MNIST dataset	I can accessthe MNIST data set to produce the accurate result.	Medium	Sprint-3
Customer (Web user)	Home	USN-9	As a user, I can view the guideto use the webapp.	I can view the awareness of this application and its limitations.	Low	Sprint-1

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data Collection	USN-1	As a user, I need to collect the data withdifferent handwriting to train the model	5	High	Vignesh M Hariharan D Haripradh ap P Murugan M Monish Kumar M
Sprint-1	Importing libraries	USN-2	As a user,I have to implement necessary libraries in python packages.	4	Low	Vignesh M Hariharan D Haripradh ap P Murugan M Monish Kumar M
Sprint-1	Data preprocessing	USN-3	As a user, I can load the dataset, handlet he missing values, scaleand split the data.	10	Medium	Vignesh M Hariharan D Haripradh ap P Murugan M Monish Kumar M
Sprint-2	Model building	USN-4	A s a user, I will get an application with ML model which provides high accuracy of recognized handwritten digit.	6	High	Vignesh M Hariharan D Haripradh ap P Murugan M Monish Kumar M

Sprint-2	Add the CNN layers	USN-5	Add input convolutional layer, max-pooling layer,flatten, hidden and outputlayers to the model.	5	High	Vignesh M Hariharan D Haripradhap P Murugan M Monish Kumar M
Sprint-2	Compile the model	USN-6	As a user, compile the model fortraineddataset.	2	Medium	Vignesh M Hariharan D Haripradhap P Murugan M Monish Kumar M
Sprint-2	Train and test themodel	USN-7	As a user, train and test themodel for thedata set collected and data are validated.	5	High	Vignesh M Hariharan D Haripradhap P Murugan M Monish Kumar M
Sprint-2	Save the model	USN-8	As a user, the compiled data are saved and integrated with an android application or web application.	2	Low	Vignesh M Hariharan D Haripradhap P Murugan M Monish Kumar M
Sprint-3	Building UI application	USN-9	As a user upload the input image thatcontains handwritten digits.	10	Medium	Vignesh M Hariharan D Haripradhap P Murugan M Monish Kumar M
Sprint-3		USN-10	As a user, I can provide the fundamentaldetails about the usage of application to customer.	4	Low	Vignesh M Hariharan D Haripradhap P Murugan M Monish Kumar M

		USN-11	As a user, I can see the predicted or recognized digits in the application.	5	Medium	Vignesh M Hariharan D Haripradhap P Murugan M Monish Kumar M
Sprint-4	Train the model on IBM	USN-12	As a usertrain the modelin IBM cloudand integrate the results.	10	High	Vignesh M Hariharan D Haripradhap P Murugan M Monish Kumar M
	Cloud Deployment	USN-13	As a user, I canaccess the web application and make the use of the productfrom anywhere.	10	High	Vignesh M Hariharan D Haripradhap P Murugan M Monish Kumar M

6.2 Sprint Delivery Schedule

Sprint	Total StoryPoints	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as onPlanned End Date)	Sprint ReleaseDate(Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 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

7. TESTING

7.1 Test Cases

Test case ID	Feature Type	Component	Test Scenario	Steps To Execute	Test Data	Expected Result	Actual Result	Status	BUG ID	Executed By
HP_TC_001	UI	Home Page	Verify UI elements in the Home Page	1) Open the page 2) Check if all the UI elements are displayed	127.0.0.1:8000	The Home page must be displayed properly	Working as expected	PASS		Vignesh M Hariharan D Haripradhap P Murugan M Monish
HP_TC_002	UI	Home Page	Check if the UI elements are displayed properly in different screen sizes	1) Open the page in a specific device 2) Check if all the UI elements are displayed properly 3) Repeat the above steps with different device sizes	-- Screen Sizes -- 2560 x 1801 1440 x 970 1024 x 840 768 x 630 320 x 630	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	BUG_HP_001	Vignesh M Hariharan D Haripradhap P Murugan M Monish Kumar M
HP_TC_003	Functional	Home Page	Check if user can upload their file	1) Open the page 2) Click on select button 3) Select the input image	Sample 1.png	The input image should be uploaded to the application successfully	Working as expected	PASS		Vignesh M Hariharan D Haripradhap P Murugan M Monish
HP_TC_004	Functional	Home Page	Check if user cannot upload unsupported files	1) Open the page 2) Click on select button 3) Select a random input file	installer.exe	The application should not allow user to select a non image file	User is able to upload any file	FAIL	BUG_HP_002	Vignesh M Hariharan D Haripradhap P Murugan M Monish
HP_TC_005	Functional	Home Page	Check if the page redirects to the result page once the input is given	1) Open the page 2) Click on select button 3) Select the input image 4) Check if the page redirects	Sample 1.png	The page should redirect to the results page	Working as expected	PASS		Vignesh M Hariharan D Haripradhap P Murugan M Monish
BE_TC_001	Functional	Backend	Check if all the routes are working properly	1) Go to Home Page 2) Upload the input image 3) Check the reults page	Sample 1.png	All the routes should properly work	Working as expected	PASS		Vignesh M Hariharan D Haripradhap P Murugan M Monish
M_TC_001	Functional	Model	Check if the model can handle various image sizes	1) Open the page in a specific device 2) Upload the input image 3) Repeat the above steps with different input image	Sample 1.png Sample 1 XS.png Sample 1 XL.png	The model should rescale the image and predict the results	Working as expected	PASS		Vignesh M Hariharan D Haripradhap P Murugan M Monish

7.2 User Acceptance Testing

Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issuesof the [ProductName] project at the time of the release to User Acceptance Testing (UAT).

Defect Analysis

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

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
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
Totals	6	1	4	3	14

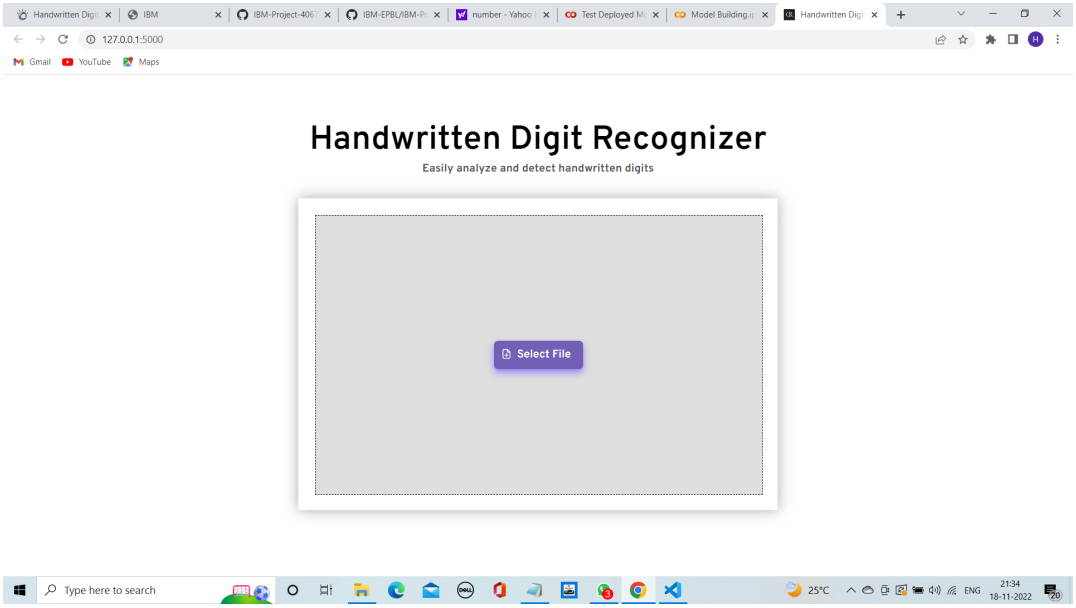
Test Case Analysis

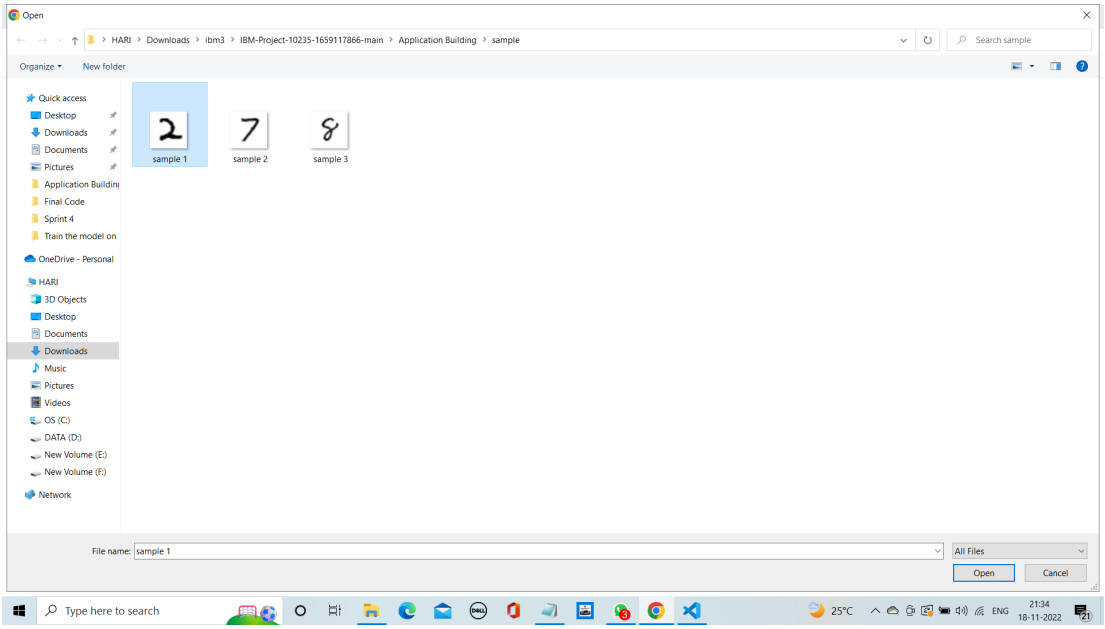
This report shows the number of test cases that have passed, failed,and untested

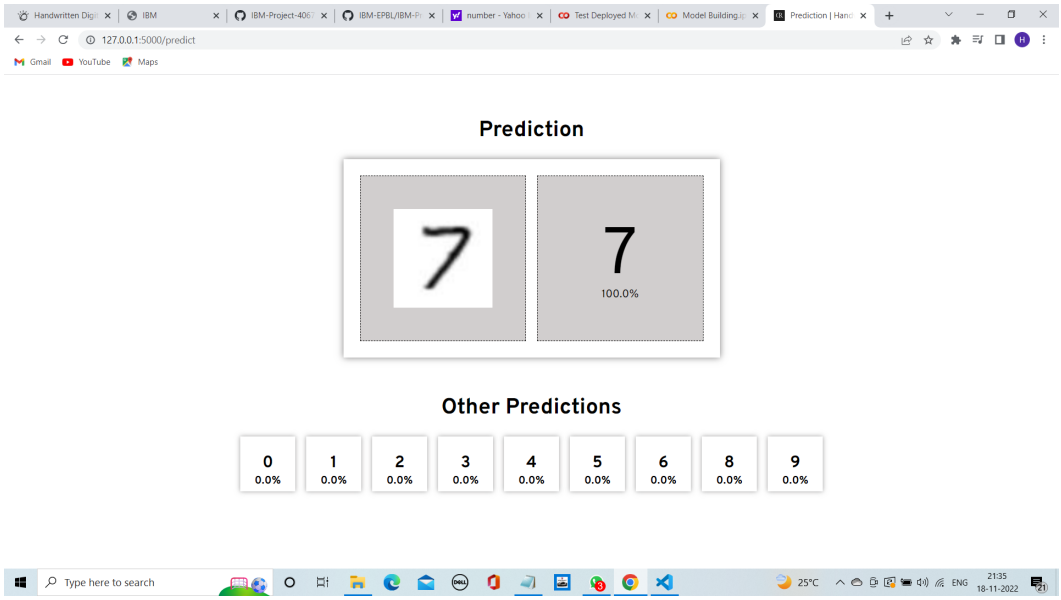
Section	Total Cases	Not Tested	Fail	Pass
Client Application	10	0	3	7
Security	2	0	0	2
Performance	3	0	0	2
Exception Reporting	2	0	0	2

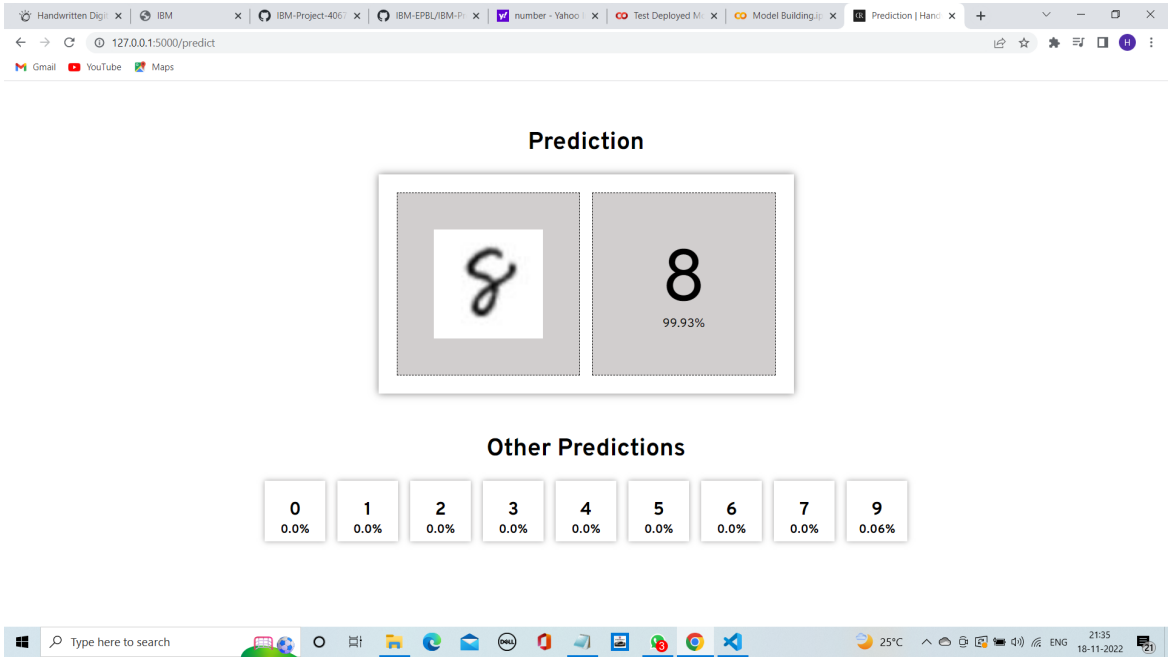
8. RESULTS

OUTPUT SCREENSHOTS






127.0.0.1:5000/predict
Prediction
7
100.0%
Other Predictions
0 0.0%
1 0.0%
2 0.0%
3 0.0%
4 0.0%
5 0.0%
6 0.0%
8 0.0%
9 0.0%
Type here to search


127.0.0.1:5000/predict
Prediction
8
99.93%
Other Predictions
0 0.0%
1 0.0%
2 0.0%
3 0.0%
4 0.0%
5 0.0%
6 0.0%
7 0.0%
9 0.06%
Type here to search

9. ADVANTAGES

In our proposed model, the data augmentation technique serves as the essential build module. Our model takes the advantage of applying data augmentation to modify the original limited dataset (MNIST dataset) with a view towards possessing big data characteristics. In this sense, extracting extra information from the original dataset enables us to obtain training data with improved quality and size, which helps in preventing overfitting in our neural network.

Inspired by the fact that a large number of epochs may result in overfitting, and a small number of epochs usually leads to underfitting, our model adopts the early stopping technique to determine the optimal number of training epochs.

In order to correctly initialize the dataset to be fit for utilizing in our proposed CNN model, data preparation is conducted as an essential first step of our proposed model. By applying the data preparation process, we can determine whether there are any redundant variables or irrelevant variables to the target variable.

To simulate the real-world natural influences that can affect image quality, which in turn influences the classification accuracy of the image, additive white Gaussian noise with $\sigma = 0.5$ was added to the MNIST dataset. Then, the noisy MNIST dataset is used to evaluate the classification performance of our proposed algorithm

10.CONCLUSION

The performance of CNN for handwritten recognition performed significantly. The proposed method obtained 98% accuracy and is able to identify real-world images as well; the loss percentage in both training and evaluation is less than 0.1, which is negligible. The only challenging part is the noise present in the real-world image, which needs to look after. The learning rate of the model is much dependent on the number of dense neurons and the cross-validation measure.

With the usage of batch normalization, we can speed up the training, reduce training and testing time, in addition to lowering the sensitivity initialization. In order to avoid overfitting and underfitting, an early stopping technique has used to determine the optimal number of training epochs.

11. FUTURE SCOPE

The proposed system takes 28x28 pixel sized images as input. The same system with further modifications and improvements in the dataset and the model can be used to build Handwritten Character Recognition System which recognizes human handwritten characters and predicts the output.

We believe that our proposed model can further be applied to other datasets. In contrast, as a future work, we find that it is worth taking further actions to improve our model performance in terms of how to perfectly learn and extract the local features in the hidden layers, and how to enhance the recognition ability in the fully connected layers to avoid mislabeling problems.

12. APPENDIX

Source Code


```
{
 "cells": [
  {
   "cell_type": "markdown",
   "id": "5c5236a8",
   "metadata": {},
   "source": [
    "Import  packages"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 1,
   "id": "e869d9d2",
   "metadata": {},
   "outputs": [],
   "source": [
    "import numpy as np\n",
    "import pandas as pd\n",
    "import matplotlib.pyplot as plt\n",
    "from keras.utils import np_utils\n",
    "from tensorflow.keras.datasets import mnist\n",
    "from tensorflow.keras.models import Sequential\n",
    "from tensorflow.keras.layers import Conv2D, Dense, Flatten\n",
    "from tensorflow.keras.optimizers import Adam\n",
    "from tensorflow.keras.models import load_model\n",
    "from PIL import Image, ImageOps"
   ]
  },
```

```
{
 "cell_type": "markdown",
 "id": "fcc79e8e",
 "metadata": {},
 "source": [
  "Load the data"
 ]
},
{
 "cell_type": "code",
 "execution_count": 2,
 "id": "e977ee08",
 "metadata": {},
 "outputs": [],
 "source": [
  "(X_train, y_train), (X_test, y_test) = mnist.load_data()"
 ]
},
{
 "cell_type": "markdown",
 "id": "d8aa2993",
 "metadata": {},
 "source": [
  "Data analysis"
 ]
},
{
 "cell_type": "code",
 "execution_count": 3,
 "id": "03ab1283",
 "metadata": {},
 "outputs": [
```

```
  {
   "name": "stdout",
   "output_type": "stream",
   "text": [
    "(60000, 28, 28)\n",
    "(10000, 28, 28)\n"
   ]
  }
 ],
 "source": [
  "print(X_train.shape)\n",
  "print(X_test.shape)"
 ]
},
{
 "cell_type": "code",
 "execution_count": 4,
 "id": "5b4e57f4",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "text/plain": [
     "array([[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,\n",
     "         0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,\n",
     "         0,   0],\n",


     "       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   3,\n",
     "        18,  18,  18, 126, 136, 175,  26, 166, 255, 247, 127,   0,   0,\n",
     "         0,   0],\n",
     "       [  0,   0,   0,   0,   0,   0,   0,   0,  30,  36,  94, 154, 170,\n",
```

```
"      [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,\n",
"         0,   0,  16,  93, 252, 253, 187,   0,   0,   0,   0,   0,   0,\n",
"         0,   0],\n",
"      [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,\n",
"         0,   0,   0,   0, 249, 253, 249,  64,   0,   0,   0,   0,   0,\n",
"         0,   0],\n",
"      [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,\n",
"         0,  46, 130, 183, 253, 253, 207,   2,   0,   0,   0,   0,   0,\n",
"         0,   0],\n",
"      [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,  39,\n",
"       148, 229, 253, 253, 253, 250, 182,   0,   0,   0,   0,   0,   0,\n",
"         0,   0],\n",
"      [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,  24, 114, 221,\n",
"       253, 253, 253, 253, 201,  78,   0,   0,   0,   0,   0,   0,   0,\n",
"         0,   0],\n",
"      [  0,   0,   0,   0,   0,   0,   0,   0,  23,  66, 213, 253, 253,\n",
"       253, 253, 198,  81,   2,   0,   0,   0,   0,   0,   0,   0,   0,\n",
"         0,   0],\n",
"      [  0,   0,   0,   0,   0,   0,  18, 171, 219, 253, 253, 253, 253,\n",
"       195,  80,   9,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,\n",
"         0,   0],\n",
"      [  0,   0,   0,   0,  55, 172, 226, 253, 253, 253, 253, 244, 133,\n",
"        11,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,\n",
"         0,   0],\n",
"      [  0,   0,   0,   0, 136, 253, 253, 253, 212, 135, 132,  16,   0,\n",
"         0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,\n",
"         0,   0],\n",
"      [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,\n",
```

```
      "          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,\n",


    },
    "execution_count": 4,
    "metadata": {},
    "output_type": "execute_result"
   }
  ],
  "source": [
   "X_train[0]"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 5,
  "id": "4db1edb5",
  "metadata": {},
  "outputs": [
   {
    "data": {
     "text/plain": [
      "5"
     ]
    },
    "execution_count": 5,
    "metadata": {},
    "output_type": "execute_result"
   }
  ],
  "source": [
   "y_train[0]"
```

]
},
{
"cell_type": "code",
"execution_count": 6,
"id": "64da4472",
"metadata": {},
"outputs": [
{
"data": {
"text/plain": [
"<matplotlib.image.AxesImage at 0x250a85a2a30>"
]
},
"execution_count": 6,
"metadata": {},
"output_type": "execute_result"
},
{
"data": {
"image/png":
"iVBORw0KGgoAAAANSUhEUgAAAPsAAAD4CAYAAAAq5pAIAAAAOXRFW
HRTb2Z0d2FyZQBNYXRwbG90bGliIHZlcnNpb24zLjMuNCwgaHR0cHM6Ly9tY
XRwbG90bGliLm9yZy8QVMy6AAAACXBIWXMAAAsTAAALEwEAmpwYAAA
OX0lEQVR4nO3dbYxc5XnG8euKbUwxJvHGseMQFxzjFAg0Jl0ZkBFQoVCCIgGK
CLGiiFBapwlOQutKUFoVWtHKrRIiSimSS6m4iWQgPAHmsSyECRqcFmoARO
HN+MS4+0aswIDIfZ6fffDjqsFdp5dZs68eO//T1rNzLnnzLk1cPmcmeeceRwRAjD5fa
DTDQBoD8IOJEHYgSQIO5AEYQeSmNrOjR3i6XGoZrk/Aqv9Fb2ht7PFatqbDb
PkfS9ZKmSPrXiFhVev6hmqGTfVYzmwRQsDE21K01fBhve4qkGyV9TtTLxkpbZPr7
R1wPQWs18Zl8i6fmI2BoReyXdJem8atoCULVmwn6kpF+Nery9ZJc+M8p9NNN0
7mtgcgGY0E/axvgR4z7m3EbE6InojoneapjexOQNXH/X445J2NNcOgFZ
pJuyPSlpke4e4HtQyR9SdK6atoCULWGh94iYp/t5ZL+Tez9SdK6atoCULWGh94iYp/tFZJ+rJGhtzUR8XRlnQGoVFPj7BH

```
": {
      "needs_background": "light"
     },
     "output_type": "display_data"
    }
   ],
   "source": [
    "plt.imshow(X_train[0])"
   ]
  },
  {
   "cell_type": "markdown",
   "id": "42ec1c47",
   "metadata": {},
   "source": [
    "Data Pre-processing"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 7,
   "id": "8453610d",
   "metadata": {},
   "outputs": [],
   "source": [
    "X_train = X_train.reshape(60000, 28, 28, 1).astype('float32')\n",
    "X_test = X_test.reshape(10000, 28, 28, 1).astype('float32')"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 8,
```

```
 "id": "7abb41b0",
 "metadata": {},
 "outputs": [],
 "source": [
  "number_of_classes = 10\n",
  "Y_train = np_utils.to_categorical(y_train, number_of_classes)\n",
  "Y_test = np_utils.to_categorical(y_test, number_of_classes)"
 ]
},
{
 "cell_type": "code",
 "execution_count": 9,
 "id": "fd70ac31",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "text/plain": [
     "array([0., 0., 0., 0., 0., 1., 0., 0., 0., 0.], dtype=float32)"
    ]
   },
   "execution_count": 9,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "Y_train[0]"
 ]
},
{
 "cell_type": "markdown",
```

```
  "id": "161265a6",
  "metadata": {},
  "source": [
   "Create model"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 10,
  "id": "90bf6dd3",
  "metadata": {},
  "outputs": [],
  "source": [
   "model = Sequential()\n",
   "model.add(Conv2D(64, (3, 3), input_shape=(28, 28, 1), activation=\"relu\"))\n",
   "model.add(Conv2D(32, (3, 3), activation=\"relu\"))\n",
   "model.add(Flatten())\n",
   "model.add(Dense(number_of_classes, activation=\"softmax\"))"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 11,
  "id": "1cf72280",
  "metadata": {},
  "outputs": [],
  "source": [
   "model.compile(loss='categorical_crossentropy', optimizer=\"Adam\",
metrics=[\"accuracy\"])"
  ]
 },
 {
```

```
  "cell_type": "markdown",
  "id": "273dd04a",
  "metadata": {},
  "source": [
   "Train the model"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 12,
  "id": "294095fe",
  "metadata": {},
  "outputs": [
   {
    "name": "stdout",
    "output_type": "stream",
    "text": [
     "Epoch 1/5\n",
     "1875/1875 [==============================] - 92s 49ms/step - loss:
0.2377 - accuracy: 0.9523 - val_loss: 0.0928 - val_accuracy: 0.9719\n",
     "Epoch 2/5\n",
     "1875/1875 [==============================] - 100s 53ms/step - loss:
0.0656 - accuracy: 0.9803 - val_loss: 0.0910 - val_accuracy: 0.9729\n",
     "Epoch 3/5\n",
     "1875/1875 [==============================] - 105s 56ms/step - loss:
0.0458 - accuracy: 0.9857 - val_loss: 0.1193 - val_accuracy: 0.9684\n",
     "Epoch 4/5\n",
     "1875/1875 [==============================] - 109s 58ms/step - loss:
0.0368 - accuracy: 0.9883 - val_loss: 0.1116 - val_accuracy: 0.9766\n",
     "Epoch 5/5\n",
     "1875/1875 [==============================] - 106s 56ms/step - loss:
0.0274 - accuracy: 0.9914 - val_loss: 0.1173 - val_accuracy: 0.9748\n"
```

```
     ]
    },
    {
     "data": {
      "text/plain": [
       "<keras.callbacks.History at 0x250a7b7ddf0>"
      ]
     },
     "execution_count": 12,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "model.fit(X_train, Y_train, batch_size=32, epochs=5,
validation_data=(X_test,Y_test))"
   ]
  },
  {
   "cell_type": "markdown",
   "id": "9b14c399",
   "metadata": {},
   "source": [
    "Test the model"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 13,
   "id": "452c47bc",
   "metadata": {},
   "outputs": [
```

```
   {
    "name": "stdout",
    "output_type": "stream",
    "text": [
     "Metrics (Test Loss & Test Accuracy): \n",
     "[0.11725354194641113, 0.9747999906539917]\n"
    ]
   }
  ],
  "source": [
   "metrics = model.evaluate(X_test, Y_test, verbose=0)\n",
   "print(\"Metrics (Test Loss & Test Accuracy): \")\n",
   "print(metrics)"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 14,
  "id": "ba63597f",
  "metadata": {},
  "outputs": [
   {
    "name": "stdout",
    "output_type": "stream",
    "text": [
     "1/1 [==============================] - 0s 82ms/step\n",
     "[[2.0987019e-13 2.4276264e-14 1.3246451e-09 1.8783747e-05 1.2520702e-19\n",
     "  1.8974715e-15 1.2059106e-19 9.9998116e-01 5.5190859e-11 3.0066661e-12]\n",
     " [3.6490474e-11 8.6607568e-11 9.9999988e-01 1.4327577e-09 6.2738566e-15\n",
```

```
      "  1.4905276e-18 3.5467995e-09 3.2217497e-17 7.2324944e-08 2.6815476e-20]\n",
      " [5.2345695e-07 9.8446769e-01 5.0688081e-04 2.0625328e-09 7.7538867e-03\n",
      "  8.2934766e-06 2.1169055e-06 2.6437547e-07 7.2603868e-03 2.1643283e-10]\n",
      " [1.0000000e+00 2.3432516e-15 1.6870000e-10 1.4166539e-14 4.9352419e-14\n",
      "  8.6952261e-13 4.7428284e-10 3.0202582e-16 1.2158017e-12 3.0176420e-10]]\n"
     ]
    }
   ],
   "source": [
    "prediction = model.predict(X_test[:4])\n",
    "print(prediction)"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 16,
   "id": "4dc08f2c",
   "metadata": {},
   "outputs": [
    {
     "name": "stdout",
     "output_type": "stream",
     "text": [
      "[7 2 1 0]\n",
      "[[0. 0. 0. 0. 0. 0. 0. 1. 0. 0.]\n",
      " [0. 0. 1. 0. 0. 0. 0. 0. 0. 0.]\n",
      " [0. 1. 0. 0. 0. 0. 0. 0. 0. 0.]\n",
```

```
    " [1. 0. 0. 0. 0. 0. 0. 0. 0. 0.]]\n"
   ]
  }
 ],
 "source": [
  "print(np.argmax(prediction, axis=1))\n",
  "print(Y_test[:4])"
 ]
},
{
 "cell_type": "markdown",
 "id": "97233b2d",
 "metadata": {},
 "source": [
  "Save the model"
 ]
},
{
 "cell_type": "code",
 "execution_count": 17,
 "id": "2eb83236",
 "metadata": {},
 "outputs": [],
 "source": [
  "model.save(\"model.h5\")"
 ]
},
{
 "cell_type": "markdown",
 "id": "6a940125",
 "metadata": {},
 "source": [
```

```
  "Test the saved model"
 ]
},
{
 "cell_type": "code",
 "execution_count": 18,
 "id": "267154d5",
 "metadata": {},
 "outputs": [],
 "source": [
  "model=load_model(\"model.h5\")"
 ]
},
{
 "cell_type": "code",
 "execution_count": 20,
 "id": "8308baec",
 "metadata": {},
 "outputs": [
  {
   "name": "stdout",
   "output_type": "stream",
   "text": [
    "1/1 [==============================] - 0s 140ms/step\n",
    "0    2\n",
    "Name: Label, dtype: int64\n"
   ]
  }
 ],
 "source": [
  "img = Image.open(\"sample.png\").convert(\"L\")\n",
  "img = img.resize((28, 28))\n",
```

```
   "img2arr = np.array(img)\n",
   "img2arr = img2arr.reshape(1, 28, 28, 1)\n",
   "results  = model.predict(img2arr)\n",
   "results = np.argmax(results,axis = 1)\n",
   "results = pd.Series(results,name=\"Label\")\n",
   "print(results)"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": null,
  "id": "0d65d64e",
  "metadata": {},
  "outputs": [],
  "source": []
 }
],
"metadata": {
 "kernelspec": {
  "display_name": "Python 3",
  "language": "python",
  "name": "python3"
 },
 "language_info": {
  "codemirror_mode": {
   "name": "ipython",
   "version": 3
  },
  "file_extension": ".py",
  "mimetype": "text/x-python",
  "name": "python",
  "nbconvert_exporter": "python",
```

```
   "pygments_lexer": "ipython3",
   "version": "3.8.8"
  }
 },
 "nbformat": 4,
 "nbformat_minor": 5
}
```

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()
```

requirements

absl-py==1.2.0
asttokens==2.0.5
astunparse==1.6.3
backcall==0.2.0
cachetools==5.2.0
certifi==2022.6.15
charset-normalizer==2.1.0
click==8.1.3
colorama==0.4.5
cycler==0.11.0
debugpy==1.6.2
decorator==5.1.1
entrypoints==0.4
executing==0.9.1
Flask==2.2.1
flatbuffers==1.12
fonttools==4.34.4
gast==0.4.0
google-auth==2.9.1
google-auth-oauthlib==0.4.6
google-pasta==0.2.0
grpcio==1.47.0
h5py==3.7.0
idna==3.3
ipykernel==6.15.1
ipython==8.4.0
itsdangerous==2.1.2
jedi==0.18.1
Jinja2==3.1.2
jupyter-client==7.3.4
jupyter-core==4.11.1

keras==2.9.0
Keras-Preprocessing==1.1.2
kiwisolver==1.4.4
libclang==14.0.6
Markdown==3.4.1
MarkupSafe==2.1.1
matplotlib==3.5.2
matplotlib-inline==0.1.3
nest-asyncio==1.5.5
numpy==1.23.1
oauthlib==3.2.0
opt-einsum==3.3.0
packaging==21.3
pandas==1.4.3
parso==0.8.3
pickleshare==0.7.5
Pillow==9.2.0
prompt-toolkit==3.0.30
protobuf==3.19.4
psutil==5.9.1
pure-eval==0.2.2
pyasn1==0.4.8
pyasn1-modules==0.2.8
Pygments==2.12.0
pyparsing==3.0.9
python-dateutil==2.8.2
pytz==2022.1
pywin32==304
pyzmq==23.2.0
requests==2.28.1
requests-oauthlib==1.3.1
rsa==4.9

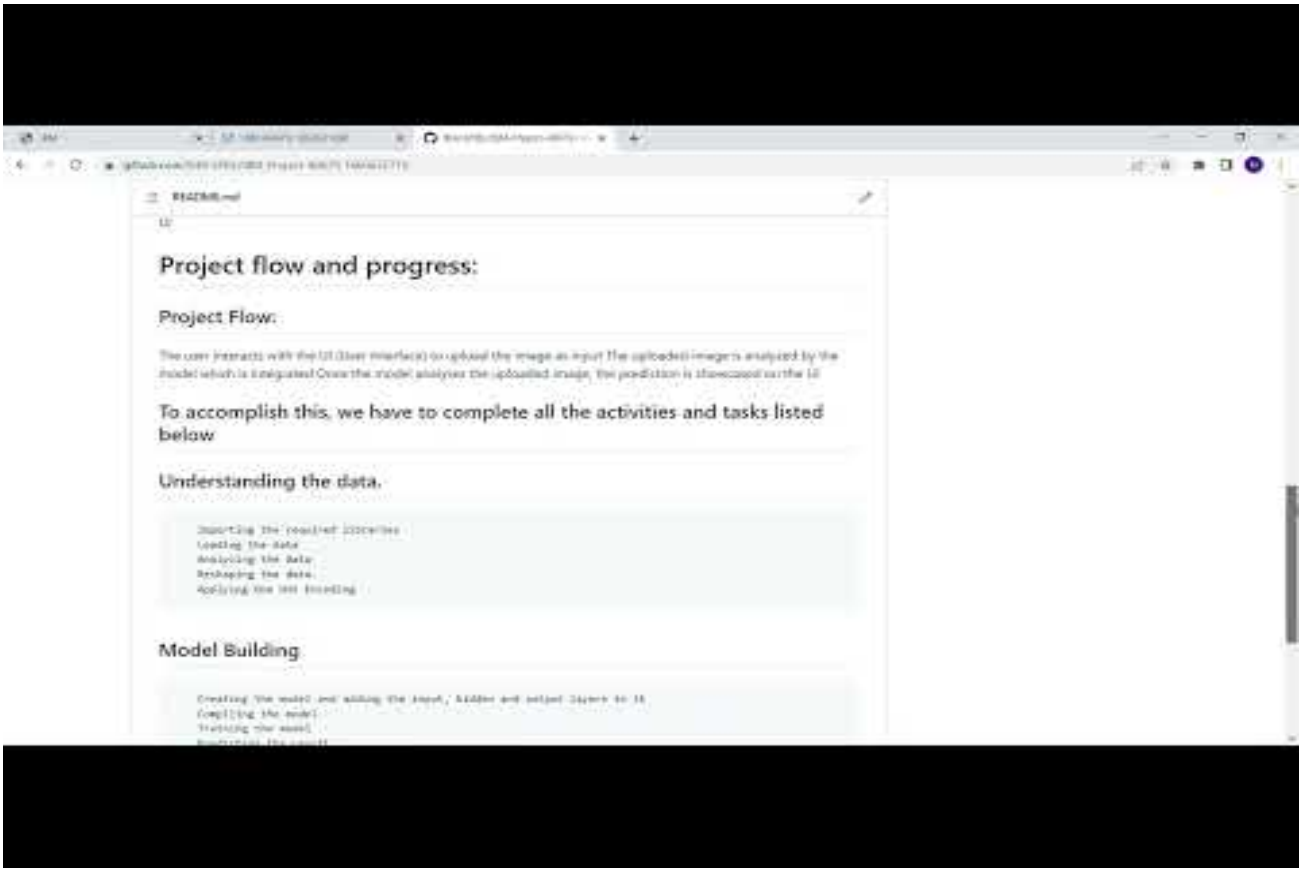
six==1.16.0
stack-data==0.3.0
tensorboard==2.9.1
tensorboard-data-server==0.6.1
tensorboard-plugin-wit==1.8.1
tensorflow==2.9.1
tensorflow-estimator==2.9.0
tensorflow-io-gcs-filesystem==0.26.0
termcolor==1.1.0
tornado==6.2
traitlets==5.3.0
typing_extensions==4.3.0
urllib3==1.26.11
wcwidth==0.2.5
Werkzeug==2.2.1
wrapt==1.14.1

GitHub Link

https://github.com/IBM-EPBL/IBM-Project-40679-1660632718

Project Demo Link

https://youtu.be/5WhpAyNrnTA


README.md
Project flow and progress:
Project Flow:
To accomplish this, we have to complete all the activities and tasks listed below
Understanding the data.
Model Building