A Novel Method for Handwritten Digit Recognition System

Team ID: PNT2022TMID12900

Bachelor of Engineering

Electronics and Communication Engineering

PSG College Of Technology

Coimbatore – 641004

Faculty Evaluator: Dr. Manoharan

Faculty Mentor: Dr. Srivatsun

Team Members:

Shanchith Kumar P

Prassanna S

Sivaram S

Pavish Gowtham S

1. INTRODUCTION

1.1 Introduction

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, using classifiers like 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 deeplearning techniques. A human learns to perform a task by practicing and repeating it again and againso that it memorizes how to perform the tasks. Then the neurons in his brain automatically triggerand 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.

For example – object recognition, image and sound classification, object detection, image segmentation, etc.

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 with many different flavors. The handwritten digit recognition is the solution to this problem which usesthe image of a digit and recognizes the digit present in the image.

1.1.1 Digit Recognition System:

Digit recognition system is the working of a machine to train itself or recognizing the digits from different sources like emails, bank cheque, papers, images, etc. and in different real-world scenariosfor online handwriting recognition on computer tablets or system, recognize number plates of , numeric entries in forms filled up by hand and so on.

1.2 Purpose:

Automating these tasks removes the need for human effort which is error prone in performing these kinds of tedious works and improves speed as well as efficiency.

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.

1. Handwritten Digit Recognition Using Various Machine Learning Algorithms and Models

SSRN 2020

Pranit Patil

Handwritten digit recognition is a technique or technology for automatically recognizing and detecting handwritten digital data through different Machine Learning models. In this paper we use various Machine Learning algorithms to enhance the productiveness of technique and reduce the complexity using various models. Machine Learning is an application of Artificial Intelligence that learns from previous experience and improves automatically through experience. We illustrate various Machine learning algorithms such as Support Vector Machine, Convolutional Neural Network, Quantum Computing, K-Nearest Neighbour Algorithm, Deep Learning used in Recognition technique.

2. Recognition of Handwritten Digit using Convolutional Neural Network in Python with TensorFlow and Comparison of Performance for Various Hidden Layers

IEEE 2019

Fathma Siddique, Shadman Sakib, Md. Abu Bakr Siddique

In recent times, with the increase of Artificial Neural Network (ANN), deep learning has brought a dramatic twist in the field of machine learning by making it more artificially intelligent. Deep learning is remarkably used in vast ranges of fields because of its diverse range of applications such as surveillance, health, medicine, sports, robotics, drones, etc. In deep learning, convolutional Neural Network (CNN) is at the center of spectacular advances that mixes Artificial Neural Network (ANN) and up to date deep learning strategies. It has been used broadly in pattern recognition, sentence classification, speech recognition, face recognition, text categorization, document analysis, scene, and handwritten digit recognition. The goal of this paper is to observe the variation of accuracies of CNN to classify handwritten digits using various numbers of hidden layers and epochs and to make the comparison between the accuracies. For this performance evaluation of CNN, we performed our experiment using Modified National Institute of Standards and Technology (MNIST) dataset. Further, the network is trained using stochastic gradient descent and the backpropagation algorithm.

3. Handwritten Digit Recognition System Based on Convolutional Neural Network

IEEE 2020

Jinze Li; Gongbo Sun; Leiye Yi; Qian Cao; Fusen Liang; Yu Sun

Image recognition is widely used in the field of computer vision today. As a kind of image recognition, digit recognition is widely used. Today, the online recognition technology in digit recognition is relatively mature while the offline recognition technology is not. This paper mainly introduces an offline recognition system for handwritten digits based on convolutional neural networks. The system uses the MINST dataset as a training sample and pre-processes the picture with the Opencv toolkit. Then it uses LeNet-5 in the convolutional neural network to extract the handwritten digit image features, repeatedly convolution pooling, and pull the result into a one-dimensional vector. And finally find the highest probability point to determine the result to achieve handwritten digit recognition with the Softmax regression model. The application of this system can greatly reduce labor costs and improve work efficiency, which is of great significance in many fields.

4. A Comparative Study on Handwriting Digit Recognition Using Neural Networks

IEEE 2017

Mahmoud M. Abu Ghosh; Ashraf Y. Maghari

The handwritten digit recognition problem becomes one of the most famous problems in machine learning and computer vision applications. Many machine learning techniques have been employed to solve the handwritten digit recognition problem. This paper focuses on Neural Network (NN) approaches. The most three famous NN approaches are deep neural network (DNN), deep belief network (DBN) and convolutional neural network (CNN). The handwritten digit recognition problem becomes one of the most famous problems in machine learning and computer vision applications. Many machine learning techniques have been employed to solve the handwritten digit recognition problem. This paper focuses on Neural Network (NN) approaches. The most three famous NN approaches are deep neural network (DNN), deep belief network (DBN) and convolutional neural network (CNN).

5. A Robust Model for Handwritten Digit Recognition using Machine and Deep Learning Technique

IEEE 2021

Ayush Kumar Agrawal; A.K. Shrivas; Vineet kumar Awasthi

In the era of research, pattern recognition is one of the most famous and widely used area in the field of research work. There are various types of patterns are available for the researches like: audio, video, handwritten digit images and handwritten characters images etc. In this paper, we concentrate in the field of handwritten digit recognition for classification of patterns. We have used famous handwritten digit datasets named as MNIST, which is collection of 70000 images. In this research work, we have suggested CNN as deep learning technique on keras for MNIST handwritten

6. Handwritten Digit Recognition using Adaptive Neuro-Fuzzy System and Ranked Features

IEEE 2019

Savita Ahlawat; Rahul Rishi

This paper investigates Adaptive Neuro-Fuzzy Inference System (ANFIS) for recognition of handwritten digits. First, an efficient feature extraction module based on five feature extraction techniques has been performed. Second, an optimal feature selection method for feature ranking and feature reduction has been proposed. Third, a classification based on ANFIS has been done.

The Experiments has been performed on standard handwritten digit dataset to evaluate the performance of the proposed system. Simulation result revels the proposed system has low testing and checking error with high recognition accuracy.

7. Segmentation and Recognition Strategy of Handwritten Connected Digits Based on the Oriented Sliding Window

IEEE 2013

Abdeljalil Gattal; Youcef Chibani

In this paper, a system to recognize handwritten digit strings, which constitutes a difficult task because of overlapping and/or joining of adjacent digits. To resolve this problem, we use a segmentation-recognition of handwritten connected digits based on the oriented sliding window. The proposed approach allows separating adjacent digits according the connection configuration by finding at the same time the interconnection points between adjacent digits and the cutting path. The segmentation-recognition using the global decision module allows the rejection or acceptance of the processed image. Experimental results conducted on the handwritten digit database NIST SD19 show the effective use of the sliding window for segmentation-recognition.

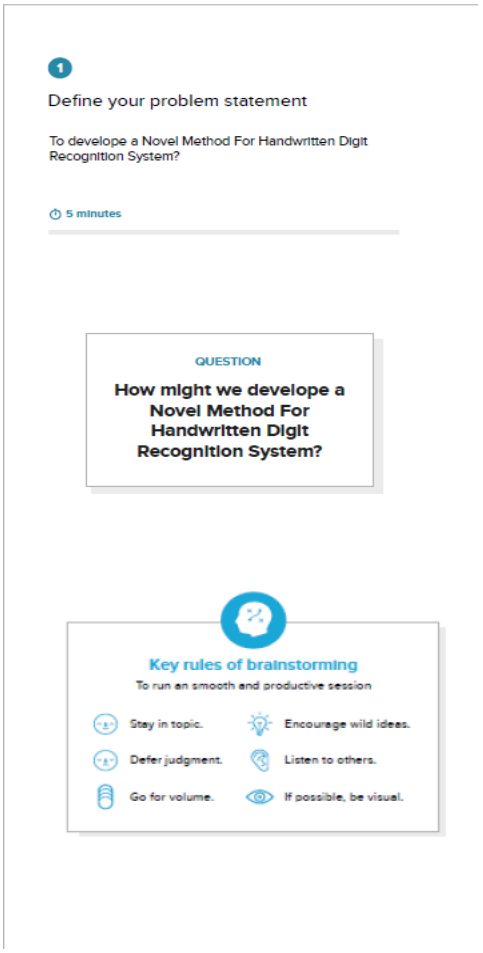
2.1 Problem Statement:

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 AND PROPOSED SOLUTION

3.1. Empathy Map Canvas

3.2. Ideation and Brainstorming

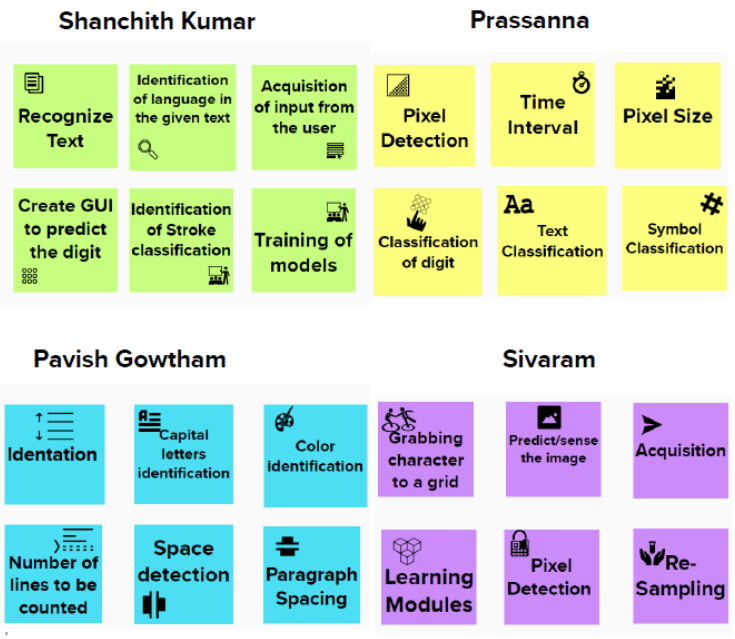


2

Brainstorm

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

10 minutes



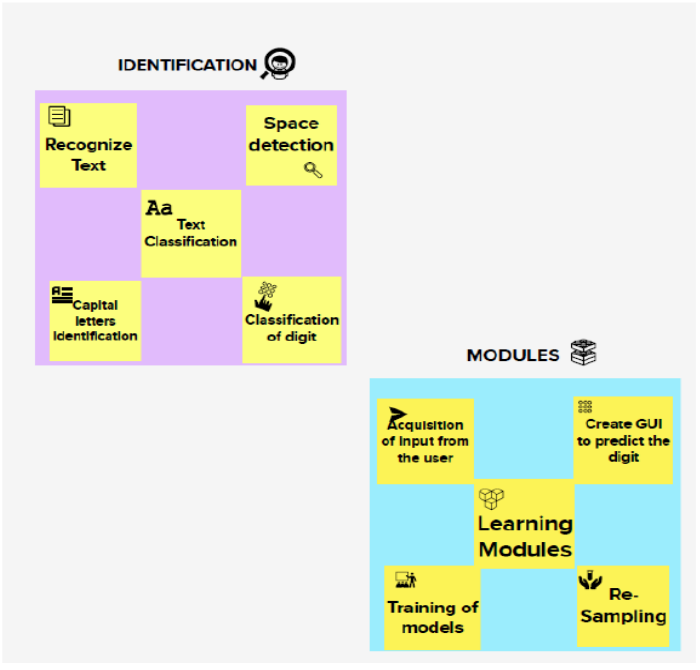
3

Group Ideas

Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10 minutes, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

TIP
You can use the **Voting session** tool above to focus on the strongest ideas.

20 minutes



3.3. Proposed Solution

S.NO	Parameter	Description
1.	Problem Statement (Problem to be solved)	**Statement**-The handwritten digit recognition is the capability of computer applications to recognize the human handwritten digits. **Description**: 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. It is the capability of a computer to fete the mortal handwritten integers from different sources like images, papers, touch defences.

		2. It allows user to translate all those signature and notes into electronic words in a text document format and this data only requires far less physical space than the storage of the physical copies.
3.	Novelty / Uniqueness	Accurately recognize the digits rather than recognizing all the characters like OCR.
4.	Social Impact / Customer Satisfaction	1. Artificial Intelligence developed the app called Handwritten digit Recognizer. 2. It converts the written word into digital approximations and utilizes complex algorithms to identify characters before churning out a digital approximation.
5.	Business Model (Revenue Model)	• This system can be integrated with traffic surveillance cameras to recognize the vehicle's number plates for effective traffic management. • Can be integrated with Postal system to identify and recognize the pin-code details easily.
6.	Scalability of the Solution	• Ability to recognise digits in more noisy environments. • There is no limit in the number of digits it can be recognized.

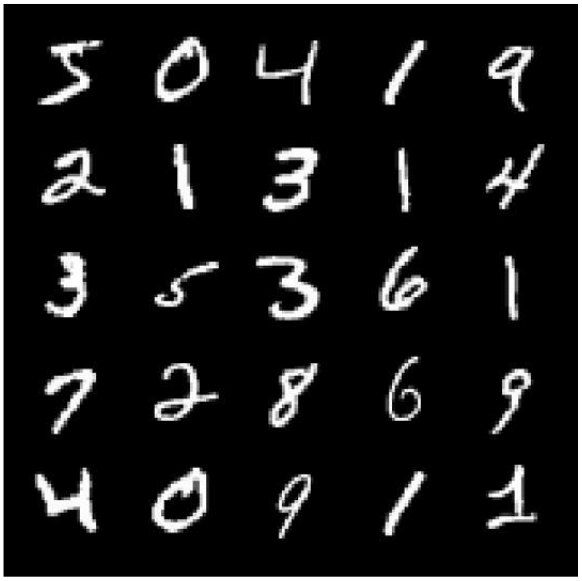
3.4. Problem Solution Fit

MNIST Dataset Description

Handwriting recognition is one of the compelling research works going on because every individual in this world has their own style of writing. It is the capability of the computer to identify and understand handwritten digits or characters automatically. Because of the progress in the field of science and technology, everything is being digitalized to reduce human effort. Hence, there comes a need for handwritten digit recognition in many real-time applications. MNIST data set is widely used for this recognition process and it has 70000 handwritten digits. We use Artificial neural networks to train these images and build a deep learning model. Web application is created where the user can upload an image of a handwritten digit. this image is analysed by the model and the detected result is returned on to UI

The MNIST Handwritten Digit Recognition Dataset contains 60,000 training and 10,000 testing labelled 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 indicates 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.

• Analyse the model summary.

• Add hidden layer to the model to make it Multi-Layer Perceptron.

• Add Dropout to prevent overfitting and check its effect on accuracy.

• Increasing the number of Hidden Layer neuron and check its effect on accuracy.

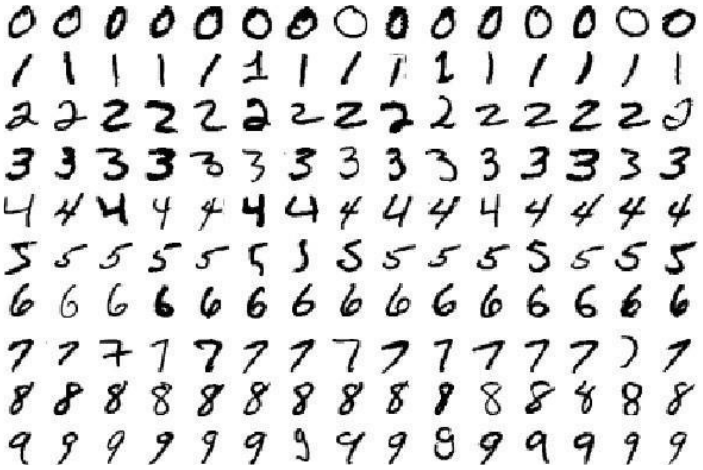
• Use different optimizers and check its effect on accuracy.

• Increase the hidden layers and check its effect on accuracy.

• Manipulate the batch size and epochs and check its effect on accuracy.

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.

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.

4. REQUIREMENT ANALYSIS

4.1. Functional Requirements

S.NO	Functional Requirement (Epic)	Sub Requirement (Sub-task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIN
FR-2	User Confirmation	Confirmation via meeting Confirmation via mail
FR-3	User Requirements	Knowledge about inputting the data Teaching on using the ML Model
FR-4	User Infrastructure	A system to support ML data modelling A Suitable GPU and CPU
FR-5	User Network	Network infrastructure to connect the wind mill to the Control station
FR-6	User Cost	User has to spend only for attaining the software, additional components are not required

4.2. Non-Functional Requirements

S.NO	Non-Functional Requirement	Description
NFR-1	Usability	It can be used for various Wind mill and trained for specific models.
NFR-2	Security	The data can be stored in a secure cloud.
NFR-3	Reliability	The predicted will be accurate and can be relied upon.
NFR-4	Performance	The performance of the model depends on the computer it runs on the accuracy of the data it is fed with.
NFR-5	Availability	It is available as a software package.
NFR-6	Scalability	It can be scaled up and interconnected with other Wind Mills to create a connected Wind Form.

5.PROJECT DESIGN

5.1. Data Flow Diagram

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.

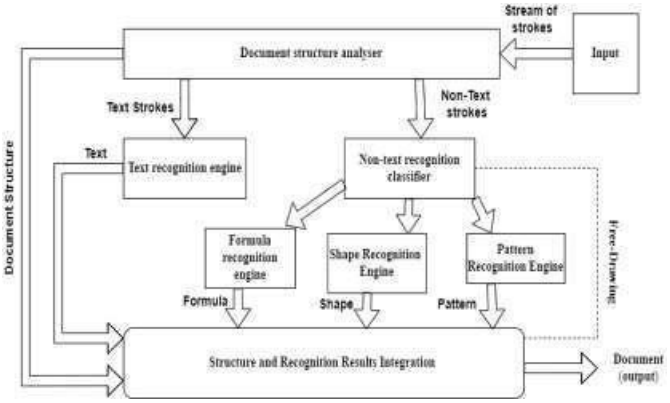


Figure: **DFD Level 0 (Industry Standard)**

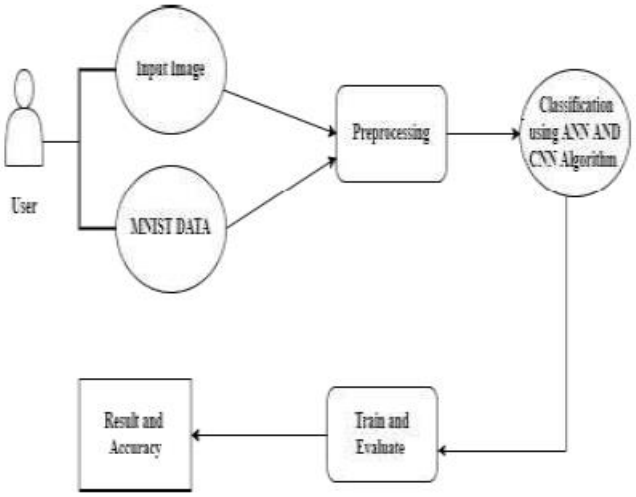


Figure: **Simplified diagram**

5.2. Solution and Technical Architecture

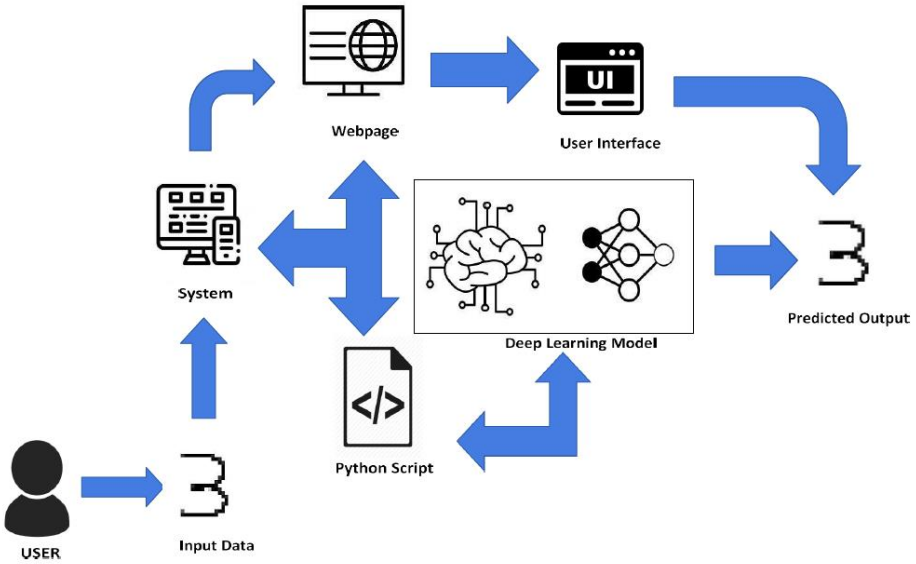


Table-1: Components & Technologies:

S.no	Components	Description	Technology
1	User Interface	API	HTML, CSS, JavaScript /Angular Js / React Js etc.
2	Application Logic-1	Data Pre-processing	Java / Python
3	Application Logic-2	Data Input	IBM Watson STT service
4	Database	Previous Year data	MySQL, NoSQL, etc.
5	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc.
6	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem
7	Machine Learning Model	Purpose of Machine Learning Model	Handwritten Digit recognition prediction Model, etc.
8	Infrastructure (Server /Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration	Local, Cloud Foundry, Kubernetes, etc.

Table-2: Application Characteristics:

S.no	Characteristics	Description	Technology
1	Open-Source Frameworks	List the open-source frameworks used	FLASK
2	Security Implementations	List all the security / access controls implemented, use of firewalls etc.	SHA-256, Encryptions, IAM Controls, OWASP etc.
3	Scalable Architecture	Justify the scalability of architecture (3 – tier, Microservices)	Cloud
4	Availability	Justify the availability of application (e.g. use of load	Distributed cloud service

		balancers, distributed servers etc.)	
5	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc.	SDN

5.3. USERS 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 view the guide andawareness to use this application.	I can view the awarenessto use this application and its limitations.	Low	Sprint-1
		USN-2	As a user, I'm allowed to view the guidedvideo 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 read the instructions to usethis application.	I can read instructions also to use it in a user-friendly method.	Low	Sprint-2
	Recognize	USN-4	As a user, In this prediction page I get to choose the image.	I can choose the image from our local system and predict 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 choosethe image from the system storage and alsoin any virtual storage.	Medium	Sprint-3
		USN-7	As a user, I will train and test the input toget the maximum accuracy of output.	I can able to train and test the application untilit gets maximum accuracy of the result.	High	Sprint-4
		USN-8	As a user, I can access the MNIST data set	I can access the MNISTdata set to produce the accurate result.	Medium	Sprint-3
Customer (Web user)	Home	USN-9	As a user, I can view the guide to use theweb app.	I can view the awareness of this application and itslimitations.	Low	Sprint-1
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 view the guide andawareness to use this application.	I can view the awarenessto use this application and its limitations.	Low	Sprint-1

		USN-2	As a user, I'm allowed to view the guidedvideo 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 read the instructions to usethis application.	I can read instructions also to use it in a user-friendly method.	Low	Sprint-2
	Recognize	USN-10	As a user, I can use the web applicationvirtually anywhere.	I can use the application portably anywhere.	High	Sprint-1
		USN-11	As it is an open source, can use it costfreely.	I can use it without any payment to be paid for itto access.	Medium	Sprint-2
		USN-12	As it is a web application, it is installationfree	I can use it without the installation of the application or any software.	Medium	Sprint-4
	Predict	USN-13	As a user, I'm Allowed to upload andchoose the image to be uploaded	I can upload and choosethe image from the system storage and alsoin any virtual storage.	Medium	Sprint-3

6.PROJECT PLANNING AND SCHEDULING

6.1. Sprint Planning and Estimation

Sprint	Milestone
Sprint 1	1. User Registers into the application through entering Email Id and Password for confirmation. 2. User Receives a confirmation mail for their registered Email. 3. User can also register to the application through Mobile number. 4. User logs in into the website using Email Id and password
Sprint 2	1. User can access the dashboard. 2. User enters the required details of weather conditions to get the desired turbine power output based on our model's prediction.
Sprint 3	1. Application stores the predictions, that can be used for future analysis. 2. The data stored has to be maintained securely
Sprint 4	1. Administrator should properly maintain the website and update it whenever required.

6.2. Sprint Delivery Schedule

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	**20**	6 Days	24 Oct 2022	29 Oct 2022	20	27 Oct 2022
Sprint-2	**20**	6 Days	31 Oct 2022	05 Nov 2022	20	04 Nov 2022
Sprint-3	**20**	6 Days	07 Nov 2022	12 Nov 2022	20	09 Nov 2022
Sprint-4	**20**	6 Days	14 Nov 2022	19 Nov 2022	20	16 Nov 2022

6.3. Reports from JIRA

7. CODING AND SOLUTION

7.1. Feature 1

Machine Learning Model Code:

```
import numpy as np #used for numerical analysis
import tensorflow #open source used for both ML and DL for
computation
import matplotlib.pyplot as plt
import os
import torch
from tensorflow import keras
from tensorflow.keras.datasets import mnist #mnist dataser
```

Importing the required Libraries.

```
(x_train , y_train),(x_test , y_test) = mnist.load_data() #x
contains the images and y has its label, like if its the
image of 1 then its label will be 1.
x_train.shape , y_train.shape , x_test.shape , y_test.shape
plt.imshow(x_train[5] , cmap = "binary")
```

Creating the test and train variables.

```
# Pre Process the images

#Normalizing the imaeges to [0,1] range
x_train = x_train.astype(np.float32)/255
x_test = x_test.astype(np.float32)/255

#expand the dimensions of the images to (28,28,1)[we use 1
here bcoz the image we take is binary]
x_train = np.expand_dims(x_train,-1)
x_test = np.expand_dims(x_test,-1)
x_train.shape , x_test.shape
```

Data Pre-Processing Stage.

```
y_train = keras.utils.to_categorical(y_train)
y_test = keras.utils.to_categorical(y_test)
```

One Hot Encoding.

```
model = Sequential() #INITIALIZE

model.add(Conv2D(32, (3,3), input_shape = (28,28,1) ,
activation = "relu"))#(3,3) is the kernal size
model.add(MaxPool2D((2,2))) # (2,2) isthe pool size

model.add(Conv2D(64, (3,3) , activation = "relu"))#(3,3) is
the kernal size
model.add(MaxPool2D((2,2))) # (2,2) isthe pool size

model.add(Flatten())

model.add(Dropout(0.25))

model.add(Dense(10, activation = "softmax")
```

Building the CNN model.

```
#callbacks

#EarlyStopping

es = EarlyStopping(monitor = "val_acc", min_delta= 0.01,
patience = 5 , verbose= 1)

 #Model Checpoint

mc = ModelCheckpoint('./bestmodel.h5' , monitor = "val_acc",
verbose = 1, save_best_only = True)

cb = [es,mc]
```

This is done to prevent the model from overfitting and underfitting. EarlyStopping does that and callbacks acts as a checkpoint where the training has to be stopped.

```
#save the model

model.save("digitreg.h5")
model_path = "/content/drive/MyDrive/digitreg.h5"
model_s = keras.models.load_model( "/content/digitreg.h5")
```

Model built is saved in digitreg.h5. Which will further be used to integrate with the website.

```
#model evalutaion/ accuracy

score = model_s.evaluate(x_test, y_test)
print(f"the model accuracy is {score[1]} ")
print(f"the model loss is {score[0]} ")
```

Provides the model accuracy and Loss value while training.

7.2. Feature 2

Deploying the Model in IBM Cloud:

```
!pip install watson-machine-learning-client
```

Here the required library of IBM Watson Machine Learning is getting installed.

```
from ibm_watson_machine_learning import APIClient
wml_credentials = {
    "url": "https://us-south.ml.cloud.ibm.com",
    "apikey": "ydsVJmqjGOQz-DPlKpo72hizlL3prSNkuQ1Jj2bjPoFR"
}
client = APIClient(wml_credentials)
client
```

```
<ibm_watson_machine_learning.client.APIClient at
0x7fd65c11c670>
```

```
def guid_space_name(client,Newspace):
    space = client.spaces.get_details()
    return(next(item for item in space['resources'] if
item['entity']['name']==Newspace)['metadata']['id'])
```

```
space_uid = guid_space_name(client, 'Newspace')
print("SPACE UID "+space_uid)
```

```
SPACE UID 936dc63b-8437-4183-a109-b29cf7928a6e
```

```
client.set.default_space(space_uid)
```

```
'SUCCESS'
```

Using the unique API key generated in IBM Cloud and mentioning our server location. Using the API credentials a new space is created in IBM Watson. The space has its unique Space id.

```
client.software_specifications.list(200)
software_space_uid=client.software_specifications.get_uid_by
_name('tensorflow_rt22.1-py3.9')
software_space_uid
```

```
'acd9c798-6974-5d2f-a657-ce06e986df4d'
```

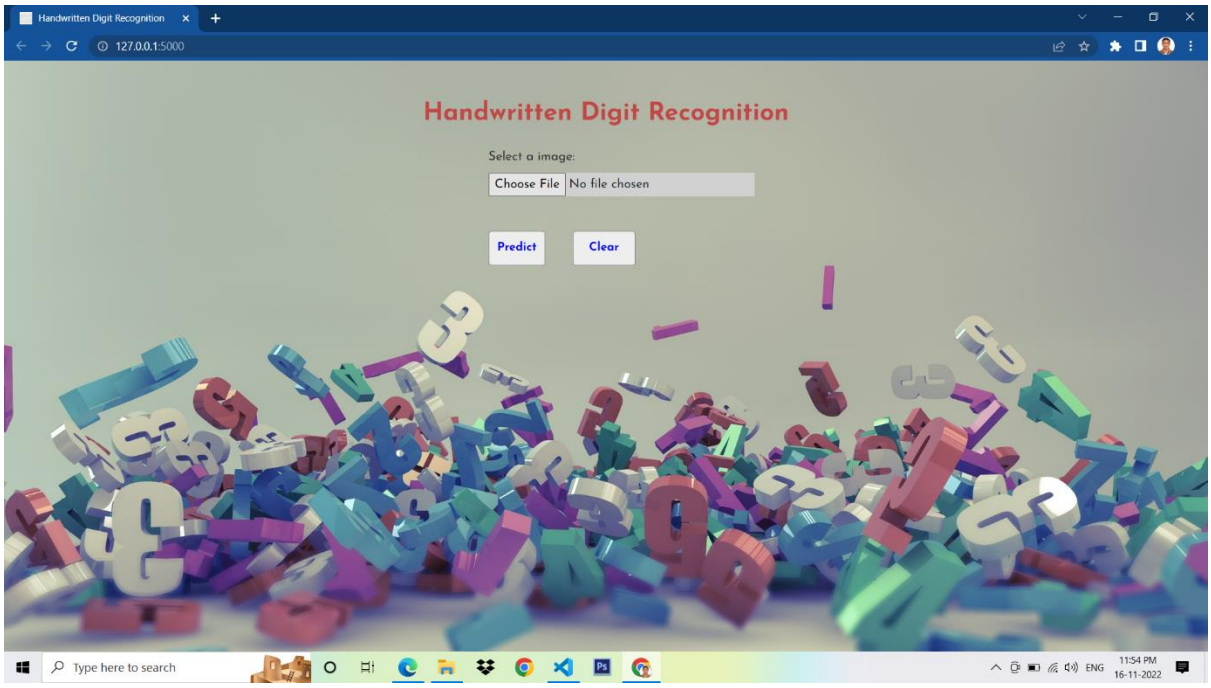
Downloading the required ML model. Looking for the version that is being supported by IBM and downloading the correct version. Creating a new deployment space for the model.

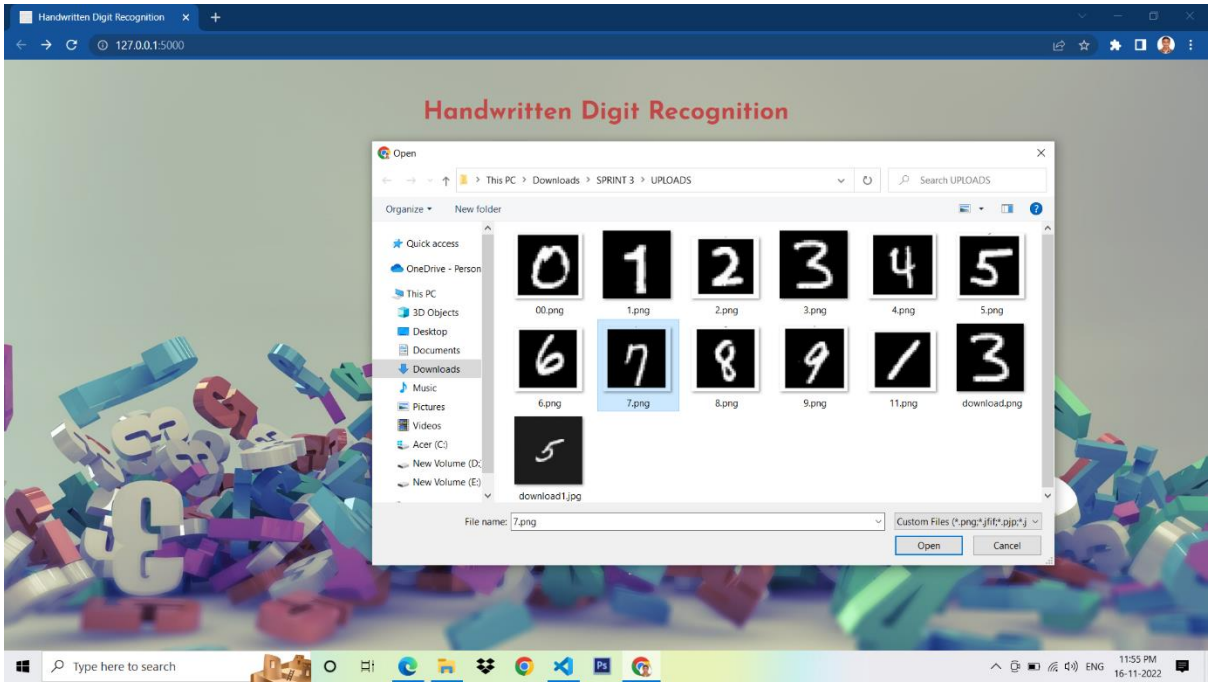
```
model_details=client.repository.store_model(model='handwritt
en-digit-recognition-model.tgz',meta_props={
    client.repository.ModelMetaNames.NAME:"CNN Model
Building",
    client.repository.ModelMetaNames.TYPE:'tensorflow_2.7,
    client.repository.ModelMetaNames.SOFTWARE_SPEC_UID:softw
are_space_uid
})
model_id=client.repository.get_model_id(model_details)
model_id
client.repository.download(model_id,
'Handwrittendigitreg.tar.gb')
```

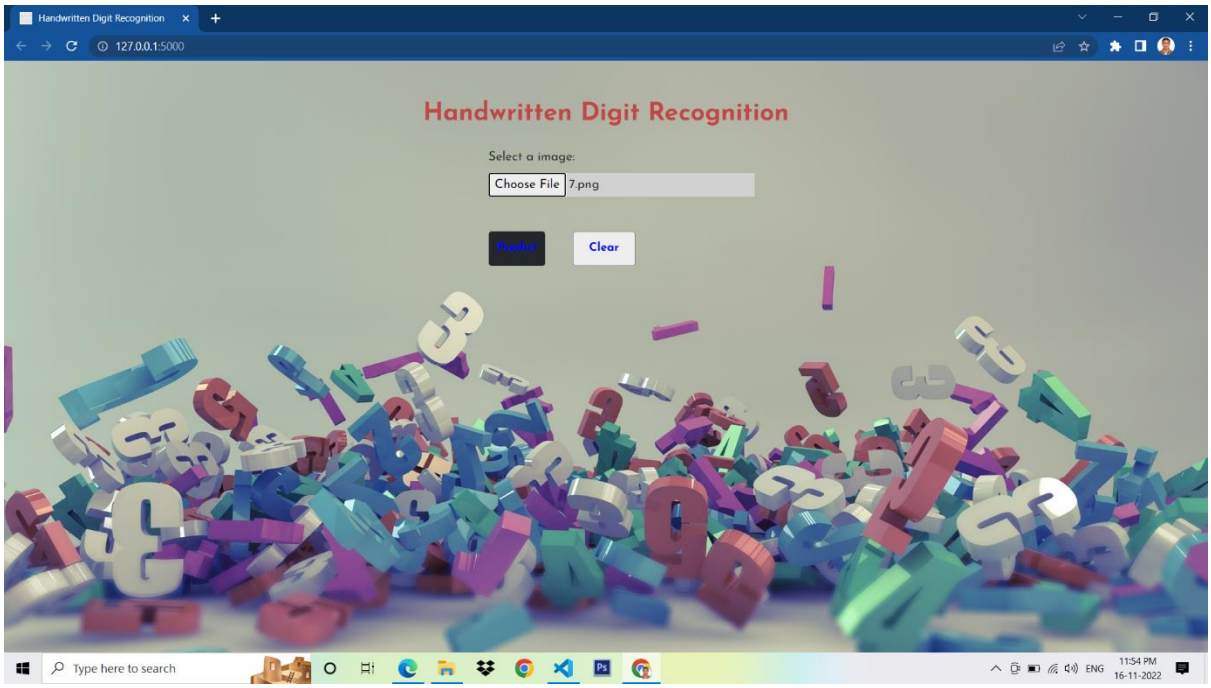
To set up the model requirements and link it to the deployment space. Saving the model to the space by mentioning the attributes of the model.

Webpage Design

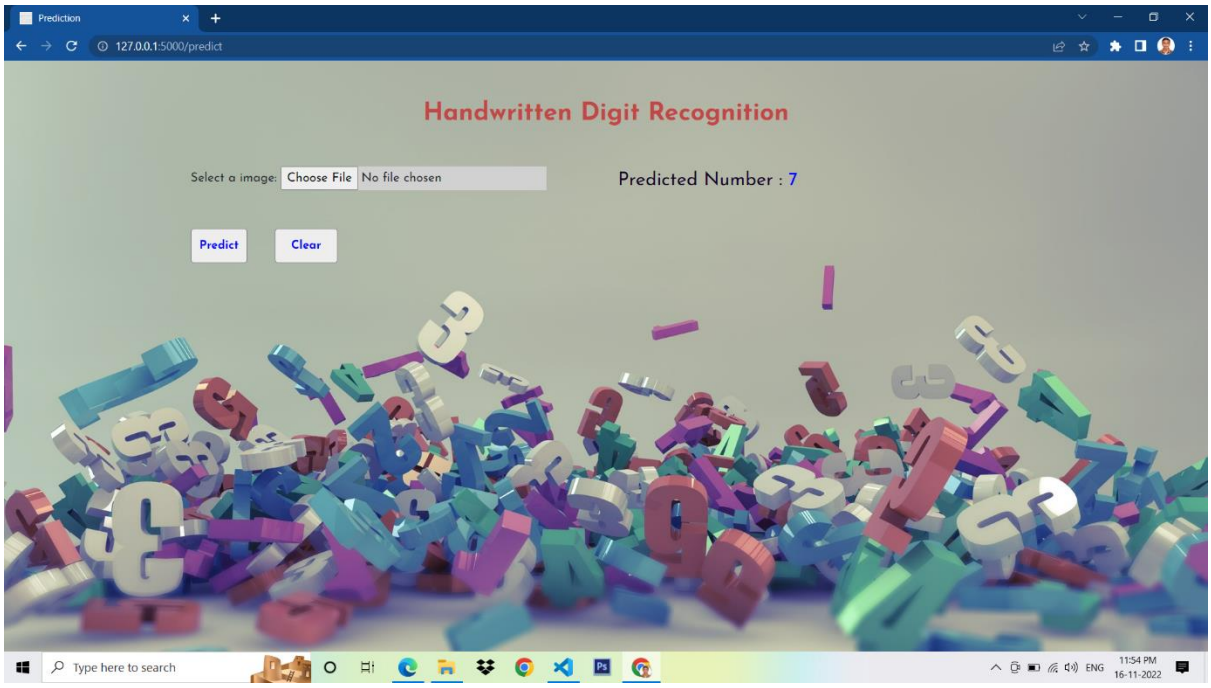
Home Page:




Handwritten Digit Recognition
127.0.0.1:5000
Handwritten Digit Recognition
Open
This PC > Downloads > SPRINT 3 > UPLOADS
Search UPLOADS
Organize
New folder
Quick access
OneDrive - Person
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
Acer (C:)
New Volume (D:)
New Volume (E:)
00.png
1.png
2.png
3.png
4.png
5.png
6.png
7.png
8.png
9.png
11.png
download.png
download1.jpg
File name: 7.png
Custom Files (*.png;*.jfif;*.pjp;*.j
Open
Cancel
Type here to search
ENG
11:55 PM
16-11-2022

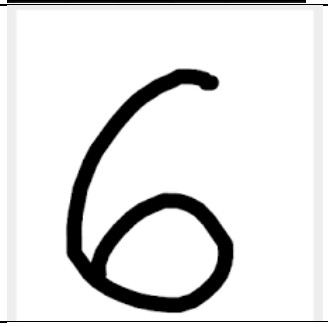
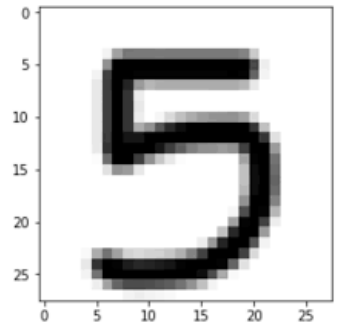

Handwritten Digit Recognition
127.0.0.1:5000
Handwritten Digit Recognition
Select a image:
Choose File
7.png
Predict
Clear
Type here to search
ENG
11:54 PM
16-11-2022

Output Page:


Prediction
127.0.0.1:5000/predict
Handwritten Digit Recognition
Select a image:
Choose File
No file chosen
Predicted Number : 7
Predict
Clear
Type here to search
ENG
11:54 PM
16-11-2022

8. TESTING

8.1. Test Cases

S.NO	INPUT IMAGE	PREDICTED VALUE
1		3
2		6
3		5

8.2. User Acceptance Testing

The project has been tested extensively with a number of users. The users found the interface very easy to use. The Web pages were colourful and attractive. There was no unnecessary details in the web page. It was clean and simple that any new user could master it. The data input format was also simple. The user need not enter any unit. User could simply upload the image. The prediction time is fairly low at an average time of 3 seconds. This delay primarily varies depending on the internet connectivity. The model has been hosted in IBM cloud. Thus, with the API available, the model can be accessed remotely from any system provided IBM access key is given. The model predicts the power output close to the actual power generated. The users are satisfied

with the predicted output power. Various inputs have been given by the users to test the consistency of the model. The model proved itself and all the users accepted the model as a reliable and convenient.

9. RESULTS

Model Accuracy

```
313/313 [==============================] - 4s 11ms/step - loss: 0.0397 - accuracy: 0.9926
the model accuracy is 0.9926000237464905
the model loss is 0.03968239203095436
```

We can infer that the model is neither overfitted nor underfitted. The model accuracy is 0.99 which is almost accurate and even the loss is minimized.

10. CONCLUSION

This project HANDWRITTEN DIGIT RECOGNITION deals with identifying the digits. The main purpose of this project is to build an automatic handwritten digit recognition method for the recognition of handwritten digit strings.

In this project, CNN (Convolutional Neural Networks) architecture is used to achieve high performance on the digit string recognition problem.

11. FUTURE SCOPE

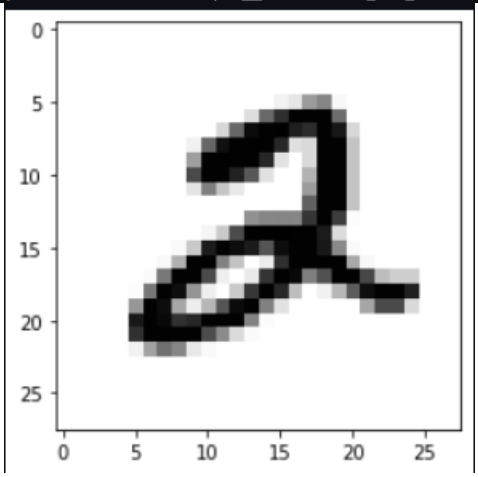
The proposed system takes 28x28 pixel sized images as input. The same system with further modifications and improvements in the dataset and the model can be used to build Handwritten Character Recognition System which recognizes human handwritten characters and predicts the output.

12.APPENDIX

12.1. SOURCE CODE

12.1.1. ML Model Code

```
import numpy as np #used for numerical analysis
import tensorflow #open source used for both ML and DL for
computation
import matplotlib.pyplot as plt
import os
import torch
from tensorflow import keras
from tensorflow.keras.datasets import mnist #mnist dataser
from tensorflow.keras.models import Sequential #it is a
plain stack of Layers
from keras.callbacks import EarlyStopping, ModelCheckpoint
from tensorflow.keras import layers #A Layer consists of a
tensor - in tensor - out computation function
from tensorflow.keras.layers import Dense , Flatten
# Faltten - used fot flattening the input or change the
dimension
# Dense - Dense Layer is the regular deeply connected in
drive.mount('/content/drive')
(x_train , y_train),(x_test , y_test) = mnist.load_data() #x
contains the images and y has its label, like if its the
image of 1 then its label will be 1.
x_train.shape , y_train.shape , x_test.shape , y_test.shape
plt.imshow(x_train[5] , cmap = "binary")
```



```
# Pre Process the images

#Normalizing the imaeges to [0,1] range
x_train = x_train.astype(np.float32)/255
x_test = x_test.astype(np.float32)/255

#expand the dimensions of the images to (28,28,1)[we use 1
here bcoz the image we take is binary]
x_train = np.expand_dims(x_train,-1)
x_test = np.expand_dims(x_test,-1)
x_train.shape , x_test.shape
```

```
y_train = keras.utils.to_categorical(y_train)
y_test = keras.utils.to_categorical(y_test)
```

```
model = Sequential() #INITIALIZE

model.add(Conv2D(32, (3,3), input_shape = (28,28,1) ,
activation = "relu"))#(3,3) is the kernal size
model.add(MaxPool2D((2,2))) # (2,2) isthe pool size

model.add(Conv2D(64, (3,3) , activation = "relu"))#(3,3) is
the kernal size
model.add(MaxPool2D((2,2))) # (2,2) isthe pool size

model.add(Flatten())

model.add(Dropout(0.25))

model.add(Dense(10, activation = "softmax")
```

```
input_shape = (28,28,1)
model.build(input_shape)
model.summary()
```

```
Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_4 (Conv2D)           (None, 26, 26, 32)        320

 max_pooling2d_4 (MaxPooling  (None, 13, 13, 32)       0
 2D)

 conv2d_5 (Conv2D)           (None, 11, 11, 64)        18496

 max_pooling2d_5 (MaxPooling  (None, 5, 5, 64)         0
 2D)

 flatten_2 (Flatten)         (None, 1600)              0

 dropout_2 (Dropout)         (None, 1600)              0

 dense_2 (Dense)             (None, 10)                16010

=================================================================
Total params: 34,826
Trainable params: 34,826
Non-trainable params: 0
```

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

```
#callbacks

#EarlyStopping

es = EarlyStopping(monitor = "val_acc", min_delta= 0.01,
patience = 5 , verbose= 1)

 #Model Checpoint

mc = ModelCheckpoint('./bestmodel.h5' , monitor = "val_acc",
verbose = 1, save_best_only = True)

cb = [es,mc]
```

```
his = model.fit(x_train,y_train, epochs = 50,
validation_split = 0.3,callbacks = cb)
```

```
Epoch 1/50
1311/1313 [============================>.] - ETA: 0s - loss: 0.2182 - accuracy: 0.9335

WARNING:tensorflow:Early stopping conditioned on metric `val_acc` which is not available. Available metrics are: loss,accuracy,val_loss,val_accuracy
WARNING:tensorflow:Can save best model only with val_acc available, skipping.

1313/1313 [==============================] - 59s 45ms/step - loss: 0.2180 - accuracy: 0.9335 - val_loss: 0.1077 - val_accuracy: 0.9665
Epoch 2/50
1312/1313 [============================>.] - ETA: 0s - loss: 0.0759 - accuracy: 0.9760

WARNING:tensorflow:Early stopping conditioned on metric `val_acc` which is not available. Available metrics are: loss,accuracy,val_loss,val_accuracy
WARNING:tensorflow:Can save best model only with val_acc available, skipping.

1313/1313 [==============================] - 46s 35ms/step - loss: 0.0759 - accuracy: 0.9760 - val_loss: 0.0567 - val_accuracy: 0.9830
Epoch 3/50
1313/1313 [==============================] - ETA: 0s - loss: 0.0541 - accuracy: 0.9832

WARNING:tensorflow:Early stopping conditioned on metric `val_acc` which is not available. Available metrics are: loss,accuracy,val_loss,val_accuracy
WARNING:tensorflow:Can save best model only with val_acc available, skipping.

1313/1313 [==============================] - 47s 35ms/step - loss: 0.0541 - accuracy: 0.9832 - val_loss: 0.0523 - val_accuracy: 0.9839
Epoch 4/50
1311/1313 [============================>.] - ETA: 0s - loss: 0.0439 - accuracy: 0.9859

WARNING:tensorflow:Early stopping conditioned on metric `val_acc` which is not available. Available metrics are: loss,accuracy,val_loss,val_accuracy
WARNING:tensorflow:Can save best model only with val_acc available, skipping.
```

```
#save the model

model.save("digitreg.h5")
model_path = "/content/drive/MyDrive/digitreg.h5"
model_s = keras.models.load_model( "/content/digitreg.h5")
```

```
#model evalutaion/ accuracy

score = model_s.evaluate(x_test, y_test)
print(f"the model accuracy is {score[1]} ")
print(f"the model loss is {score[0]} ")
```

```
313/313 [==============================] - 4s 11ms/step - loss: 0.0397 - accuracy: 0.9926
the model accuracy is 0.9926000237464905
the model loss is 0.03968239203095436
```

12.1.2. IBM Cloud Deployment

```
!pip install watson-machine-learning-client
```

```
from ibm_watson_machine_learning import APIClient
wml_credentials = {
    "url": "https://us-south.ml.cloud.ibm.com",
```

```
    "apikey": "ydsVJmqjGOQz-DPlKpo72hizlL3prSNkuQ1Jj2bjPoFR"
}
client = APIClient(wml_credentials)
client
```

<ibm_watson_machine_learning.client.APIClient at 0x7fd65c11c670>

```
def guid_space_name(client,Newspace):
    space = client.spaces.get_details()
    return(next(item for item in space['resources'] if item['entity']['name']==Newspace)['metadata']['id'])
```

```
space_uid = guid_space_name(client, 'Newspace')
print("SPACE UID "+space_uid)
```

SPACE UID 936dc63b-8437-4183-a109-b29cf7928a6e

```
client.set.default_space(space_uid)
```

'SUCCESS'

```
client.software_specifications.list(200)
software_space_uid=client.software_specifications.get_uid_by_name('tensorflow_rt22.1-py3.9')
software_space_uid
```

'acd9c798-6974-5d2f-a657-ce06e986df4d'

```
model_details=client.repository.store_model(model='handwritten-digit-recognition-model.tgz',meta_props={
    client.repository.ModelMetaNames.NAME:"CNN Model Building",
    client.repository.ModelMetaNames.TYPE:'tensorflow_2.7,
    client.repository.ModelMetaNames.SOFTWARE_SPEC_UID:software_space_uid
})
model_id=client.repository.get_model_id(model_details)
model_id
client.repository.download(model_id, 'Handwrittendigitreg.tar.gb')
```

12.1.3. HTML Home page

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>Handwritten Digit Recognition</title>
    <link rel="icon" href="static/images/digit_image.png"
type="image/x-icon">


  <meta name="viewport" content="width=device-width">
  <!-- GoogleFont -->
  <link
href="https://fonts.googleapis.com/css2?family=Prompt:wght@6
00&display=swap" rel="stylesheet">
  <link
href="https://fonts.googleapis.com/css2?family=Varela+Round&
display=swap" rel="stylesheet">
  <link
href="https://fonts.googleapis.com/css2?family=Source+Code+P
ro:wght@500&display=swap" rel="stylesheet">
  <link
href="https://fonts.googleapis.com/css?family=Calistoga|Jose
fin+Sans:400,700|Pacifico&display=swap" rel="stylesheet">
  <!-- bootstrap -->
  <!-- <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css
/bootstrap.min.css" integrity="sha384-
ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2M
Zw1T" crossorigin="anonymous">
  <link rel="stylesheet" type= "text/css" href= "{{
url_for('static',filename='css/style.css') }}"> -->
  <link rel="stylesheet" href="static/css/bootstrap.min.css"
type="text/css">
  <link rel="stylesheet" href="static/css/STYLES.css"
type="text/css">
  <!-- fontawesome -->
```

```
  <script src="https://kit.fontawesome.com/b3aed9cb07.js"
crossorigin="anonymous"></script>

  <script src="https://code.jquery.com/jquery-
3.3.1.slim.min.js" integrity="sha384-
q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6
jizo" crossorigin="anonymous"></script>
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7
/umd/popper.min.js" integrity="sha384-
UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHND
z0W1" crossorigin="anonymous"></script>
  <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/b
ootstrap.min.js" integrity="sha384-
JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B0
7jRM" crossorigin="anonymous"></script>
  <script
src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@latest"><
/script>

</head>

<script>
  function preview() {
    frame.src=URL.createObjectURL(event.target.files[0]);
}

    $(document).ready(function() {
          $('#clear_button').on('click', function() {
              $('#image').val('');
              $('#frame').attr('src',"");
            });
        });

</script>

<body>
```

```
  <div class="container">
    <div class="row">
      <div class="col-lg-12" style="text-align:
center;"><br><br>
        <h3 class="heading">Handwritten Digit
Recognition</h3><br><br>
    </div>
        <div class="col-lg-4"></div>
        <div class="col-lg-4 s1">
  <!-- <section id="content"> -->

        <!-- <div class="leftside"> -->
        <form action="/predict" method="POST"
enctype="multipart/form-data">
        <label>Select a image:</label>
        <input id="image" type="file" name="image"
accept="image/png, image/jpeg" onchange="preview()"><br><br>
          <div class="buttons_div">
            <button type="submit" class="btn btn-dark"
id="predict_button">Predict</button>
            <button type="button" class="btn btn-dark"
id="clear_button">  Clear  </button>
          </div>
        </form>
        </div>
  <!-- </section> -->
  </div>
  </div>

</body>

</html>
```

12.1.4. HTML Output Page

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Prediction</title>
    <link rel="icon" href="static/images/digit_image.png"
type="image/x-icon">
    <meta name="viewport" content="width=device-width">
  <!-- GoogleFont -->
  <link
href="https://fonts.googleapis.com/css2?family=Prompt:wght@6
00&display=swap" rel="stylesheet">
  <link
href="https://fonts.googleapis.com/css2?family=Varela+Round&
display=swap" rel="stylesheet">
  <link
href="https://fonts.googleapis.com/css2?family=Source+Code+P
ro:wght@500&display=swap" rel="stylesheet">
  <link
href="https://fonts.googleapis.com/css?family=Calistoga|Jose
fin+Sans:400,700|Pacifico&display=swap" rel="stylesheet">
  <link rel="stylesheet" href="static/css/bootstrap.min.css"
type="text/css">
  <link rel="stylesheet" href="static/css/STYLES.css"
type="text/css">
  <script src="https://kit.fontawesome.com/b3aed9cb07.js"
crossorigin="anonymous"></script>

  <script src="https://code.jquery.com/jquery-
3.3.1.slim.min.js" integrity="sha384-
q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6
jizo" crossorigin="anonymous"></script>
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7
/umd/popper.min.js" integrity="sha384-
UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHND
z0W1" crossorigin="anonymous"></script>
```

```
  <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/b
ootstrap.min.js" integrity="sha384-
JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B0
7jRM" crossorigin="anonymous"></script>
  <script
src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@latest"><
/script>
</head>
<body>
    <div class="container">
        <div class="row">
            <div class="col-lg-12" style="text-align:
center;"><br><br>
                <h3 class="heading">Handwritten Digit
Recognition</h3><br><br><br>
            </div>
            <!-- <div class="col-lg-4"></div> -->
            <div class="col-lg-6 s1">
      <!-- <section id="content"> -->

            <!-- <div class="leftside"> -->
            <form action="/predict" method="POST"
enctype="multipart/form-data">
            <label>Select a image:</label>
            <input id="image" type="file" name="image"
accept="image/png, image/jpeg" onchange="preview()"><br><br>
              <div class="buttons_div">
                <button type="submit" class="btn btn-dark"
id="predict_button">Predict</button>
                <button type="button" class="btn btn-dark"
id="clear_button">  Clear  </button>
              </div>
            </form>
            </div>
            <div class="col-lg-6">
                <label class="pred">Predicted Number :
<label style='color:#0000ff'>{{num}}</label></label>
```

```
            </div>
      <!-- </section> -->
      </div>
      </div>

</body>
</html>
```

12.1.5. CSS

```
html,
body {
  height: 100%;
  font-family: "Josefin Sans", sans-serif;
  -webkit-font-smoothing: antialiased;
  overflow-x: hidden;
  background-image: url('../IMAGES/INDEX IMAGE.jpg');
  background-size: cover;
}

h1,
h2,
h3,
h4,
h5,
h6 {
  margin: 0;
  color: #111111;
  font-weight: 400;
  font-family: "Play", sans-serif;
}


h1 {
  font-size: 50px;
  font-weight: 600;
  padding-bottom: 25px;
  color: #ffffff;
}
```

```css
h2 {
  font-size: 34px;
  padding-bottom: 15px;
  color: #ffffff;
}

h3 {
  font-size: 28px;
  padding-bottom: 15px;
  color: #adadad;
}

h4 {
  font-size: 24px;
  padding-bottom: 15px;
}

h5 {
  font-size: 18px;
  padding-bottom: 15px;
  color: #00bfe7;
}

h6 {
  font-size: 16px;
  padding-bottom: 15px;
}

p {
  padding-bottom: 15px;
  font-size: 16px;
  font-family: "Josefin Sans", sans-serif;
  color: #adadad;
  font-weight: 400;
  line-height: 26px;
  margin: 0 0 15px 0;
}
```

```css
img {
  width: fit-content;
  overflow: hidden;
}

input {
  background-color: #d1d1d1;
}
input:focus,
select:focus,
button:focus,
textarea:focus {
  outline: none;
  background-color: #7fe8e1;
  color: black;
}

a:hover,
a:focus {
  text-decoration: none;
  outline: none;
  color: #fff;
}

ul,
ol {
  padding: 0;
  margin: 0;
}

td {
  /* color: #00bfe7; */
  color: #ffffff;
  padding-bottom: 10px;
}

.td1 {
```

```css
  /* color: #adadad; */
  color: #00bfe7;
}

.s1 {
  align-items: center;
  padding-left: 40px;
}

.pred {
  color: #100028;
  /* font-style: italic; */
  font-size: 24px;
}

.heading {
  color:rgb(194, 71, 71);
  font-family: "Josefin Sans", sans-serif !important;
  text-align: center;
}

/* buttons */
.primary-btn-1 {
  display: inline-block !important;
  font-size: 15px !important;
  font-family: "Play", sans-serif !important;
  font-weight: 700 !important;
  padding: 14px 32px 12px !important;
  color: #ffffff !important;
  background: none !important;
  text-transform: uppercase !important;
  letter-spacing: 2px !important;
  position: relative !important;
  z-index: 1!important;
  border-color: #00bfe7 !important;
}
.primary-btn-1:hover:before {
  height: 100%!important;
```

```
  width: 100%!important;
}
.primary-btn-1:hover:after {
  height: 100%!important;
  width: 100%!important;
}
.primary-btn-1:before {
  position: absolute!important;
  left: 0!important;
  top: 0!important;
  height: 30px!important;
  width: 30px!important;
  border-left: 2px solid #00bfe7!important;
  border-top: 2px solid #00bfe7!important;
  content: ""!important;
  z-index: -1!important;
  -webkit-transition: all, 0.7s!important;
  -o-transition: all, 0.7s!important;
  transition: all, 0.7s!important;
}
.primary-btn-1:after {
  position: absolute !important;
  right: 0 !important;
  bottom: 0 !important;
  height: 30px !important;
  width: 30px !important;
  border-right: 2px solid #00bfe7 !important;
  border-bottom: 2px solid #00bfe7 !important;
  content: "" !important;
  z-index: -1!important;
  -webkit-transition: all, 0.7s!important;
  -o-transition: all, 0.7s!important;
  transition: all, 0.7s!important;
}

/* buttons */
.primary-btn {
  display: inline-block;
```

```
  font-size: 15px;
  font-family: "Play", sans-serif;
  font-weight: 700;
  padding: 14px 32px 12px;
  color: #ffffff;
  text-transform: uppercase;
  letter-spacing: 2px;
  position: relative;
  z-index: 1;
}
.primary-btn:hover:before {
  height: 100%;
  width: 100%;
}
.primary-btn:hover:after {
  height: 100%;
  width: 100%;
}
.primary-btn:before {
  position: absolute;
  left: 0;
  top: 0;
  height: 30px;
  width: 30px;
  border-left: 2px solid #00bfe7;
  border-top: 2px solid #00bfe7;
  content: "";
  z-index: -1;
  -webkit-transition: all, 0.7s;
  -o-transition: all, 0.7s;
  transition: all, 0.7s;
}
.primary-btn:after {
  position: absolute;
  right: 0;
  bottom: 0;
  height: 30px;
  width: 30px;
```

```
  border-right: 2px solid #00bfe7;
  border-bottom: 2px solid #00bfe7;
  content: "";
  z-index: -1;
  -webkit-transition: all, 0.7s;
  -o-transition: all, 0.7s;
  transition: all, 0.7s;
}

.portfolio__btn {
  position: absolute;
  text-align: center;
}

#clear_button{
  margin-left: 15px;
  font-weight: bold;
  color: blue;
}

#confidence{
  font-family: 'Josefin Sans', sans-serif;
  margin-top: 7.5%;
}

#content{
  margin: 0 auto;
  padding: 2% 15%;
  padding-bottom: 0;
}

.welcome{
   text-align: center;
   position: relative;
   color: honeydew;
   background-color: greenyellow;
   padding-top: 1%;
   padding-bottom: 1%;
```

```
   font-weight: bold;
   font-family: 'Prompt', sans-serif;
}

#team_id{
   text-align: right;
   font-size: 25px;
   padding-right: 3%;
}

#predict_button{
  margin-right: 15px;
  color: blue;
  font-weight: bold;
}

#prediction_heading{
  font-family: 'Josefin Sans', sans-serif;
  margin-top: 7.5%;
}

#result{
  font-size: 5rem;
}

#title{
  padding: 1.5% 15%;
  margin: 0 auto;
  text-align: center;
}

.btn {
    font-size: 15px;
    padding: 10px;

    : none;
    background: #eee;
    border: 1px solid #888;
```

```
    margin-top: 20px;
    margin-bottom: 20px;
}

.buttons_div{
  margin-bottom: 30px;
  margin-right: 80px;
}

.heading{
  font-family: 'Varela Round', sans-serif;
  font-weight: 700;
  font-size: 2rem;
  display: inline;
}

.leftside{
  text-align: center;
  margin: 0 auto;
  margin-top: 2%;
  /* padding-left: 10%; */
}

#frame{
  margin-right: 10%;
}

.predicted_answer{
  text-align: center;
  margin: 0 auto;
  padding: 3% 5%;
  padding-top: 0;
  /* padding-left: 10%; */
}

p{
  font-family: 'Source Code Pro', monospace,sans-serif;
  margin-top: 1%;
```

```
}

@media (min-width: 720px) {
  .leftside{
    padding-left: 10%;
  }
}
```

12.2. GitHub and Project Demo Link

GitHub Repo: https://github.com/IBM-EPBL/IBM-Project-37905-1660364543

Project Demo Link: https://drive.google.com/drive/folders/1dUDoldVZHxYqIY_Hj31Wv4D2Lxp_6eig?usp=sharing