PROJECTREPORT

A NOVEL METHOD FOR HANDWRITTEN DIGIT RECOGNITION SYSTEM

submittedby

PNT2022TMID35172

Vignesh G	-	2018103078
Jayakrishna K	-	2019103023
Murali R	-	2019103038
Raghuraj S V	-	2019103563

TABLEOFCONTENTS

CHAPTER 1

INTRODUCTION

1.1 PROJECTOVERVIEW

Machine learning and deep learning play an important role in computer technology andartificial intelligence. With the use of deep learning and machine learning, human effort can be reduced in recognizing,learning, predictions and in many moreareas.

HandwrittenDigitRecognitionistheabilityofcomputersystemstorecognise Handwrittendigits from various sources,such as images, documents, and so on.This project aims to let users take advantage of machine learning to reduce manual tasks in recognizing digits.

1.2 PURPOSE

Digit recognition systems are capable of recognizing the digits from different sources like emails,bankcheque,papers,images,etc.and in different real-world scenarios for online Hand writing recognition on computer tablets or system, recognize number plates of vehicles,processing bank cheque amounts,numeric entries informs filled up by handandsoon.

CHAPTER2
LITERATURESURVEY

2.1 EXISTINGPROBLEM

The fundamental problem with hand written digit recognition is that handwritten digits do not always have the same size, width, orientation, and margins since they vary from person to person. Additionally, there would be issues with identifying the numbers because of similarities between numerals like 1and7, 5and6 ,3and8, 2and5,2and7,etc.Finally,the Individuality and variation of each individual's handwriting influence the structure and appearance of the digits.

2.2 REFERENCES

ImprovedHandwrittenDigitRecognitionUsingConvolutionalNeuralNetworks(CNN)(2020)

AyushKumarAgrawalandVineetKumarAwasthi

An artificial neural network has one hidden layer between the input and output layers, whereas a deep neuralnetworkhasnumeroushiddenlayerswithinputandoutputlayers.Deepneuralnetworksuseseveralhiddenlayersto increasemodelperformanceandachievehigheraccuracycomparedtoaccuracyofmachinelearningmodels.

Most researchers do their research in the area of pattern recognition. In the field of pattern recognition, there aremany patterns that can be used, including handwritten numbers, characters, pictures, faces, sounds, and speech.Thisstudyfocusesontheclassificationandrecognitionofhandwrittendigits.1000wereutilizedastestsamplesand1 000 were training samples.10000 picture samples make up the USPS dataset, of which 7291 serve as trainingsamples and 2007 serve as testing samples. We've used the proposed deep neural network technique in this paperto classify and identify data from the ARDIS and USPS datasets. The suggested model consists of six layers withsoftmax and relu activation functions. After model implementation, accuracy for ARDIS samples reached 98.70%testingand99.76%training,whichisgreaterthanaccuracyfrompriorresearch.Additionally,usingtheUSPS

samplesdataset,98.22%trainingaccuracyand93.01%testingaccuracywereattained.Whencomparedtoearliermetho dologies,thedatashowthatdeepneuralnetworksperformincrediblywell.

Recognitionofisolatedandsimplyconnectedhandwrittennumerals,PatternRecognition.(198 6)

M.ShridharandA.Badreldin

In this paper the authors describe the results of their investigation into the development of a recognition algorithmfor identifying numerals that may be isolated or connected, broken or continuous. Using a structural classificationscheme,therecognitionalgorithmisderivedasatreeclassifier.Inanextensivetestexperiment,anaccuracyof 99%was realized with isolated numerals. When connected numerals were also included a recognition accuracy of 93%wasobtained.

HandwrittenCharacterRecognitionusingNeuralNetworkandTensorFlow(2019)

MeghaAgarwal,Shalika,VinamTomar,PriyankaGupta

The offline handwritten character recognition in this study will be carried out using Tensorflow and a convolutionalneural network. a process known as using SoftMax Regression, one may assign probabilities to one of the manycharacters in the handwritten text that offers the range of values from 0 to 1, summed to 1. The objective is tocreatesoftwarethatisextremelyaccurateandthathasaminimumlevelofspatialandtemporalcomplexity.Itwasdetermi ned that strategies for feature extraction like diagonal and direction are significantly better at producinghigh accuracy. Outcomes in comparison to other conventional vertical and horizontal techniques moreover use thebest Neural network tried layers provides the benefit of a higher accurate outcome by having a high noisetolerance.Thefeedforwardmodelinneuralnetworksistheback-propagationalgorithmthatwasprimarilyusedtoclassify the characters, recognise them, and receive training continually more. In addition to these, normalizingalong with feature extraction, the results were better and more effective. Character recognition is the outcome ofaccuracy. The paper will describe the best approach to get more than 90% accuracy in the field of HandwrittenCharacterRecognition(HCR).

HandwrittenDigitRecognitionofMNISTdatasetusingDeepLearningstate-of-the-artArtificialNeuralNetwork(ANN) andConvolutional NeuralNetwork(CNN) (2021)

DrishtiBeohar,A.Rasool

Handwrittendigitrecognitionisanintricateassignmentthatisvitalfordevelopingapplications,incomputervisiondigit recognition is one of the major applications. There has been a copious exploration done in the HandwrittenCharacterRecognitionutilizingdifferentdeeplearningmodels.Deeplearningisrapidlyincreasingindemandd uetoits resemblance to the human brain. The two major Deep learning algorithms Artificial Neural Network andConvolutional Neural Network which have been compared in this paper considering their feature extraction andclassification stages of recognition. The models were trained using categorical cross-entropy loss and ADAMoptimizer on the MNIST dataset. Backpropagation along with Gradient Descent is being used to train the networksalong with reLU activations in the network which do automatic feature extraction. In neural networks, ConvolutionNeural Network (ConvNets or Convolutional neural networks) is one of the primary classifiers to do imagerecognition,imageclassificationtasksinComputerVision.

2.3 PROBLEMSTATEMENTDEFINITION

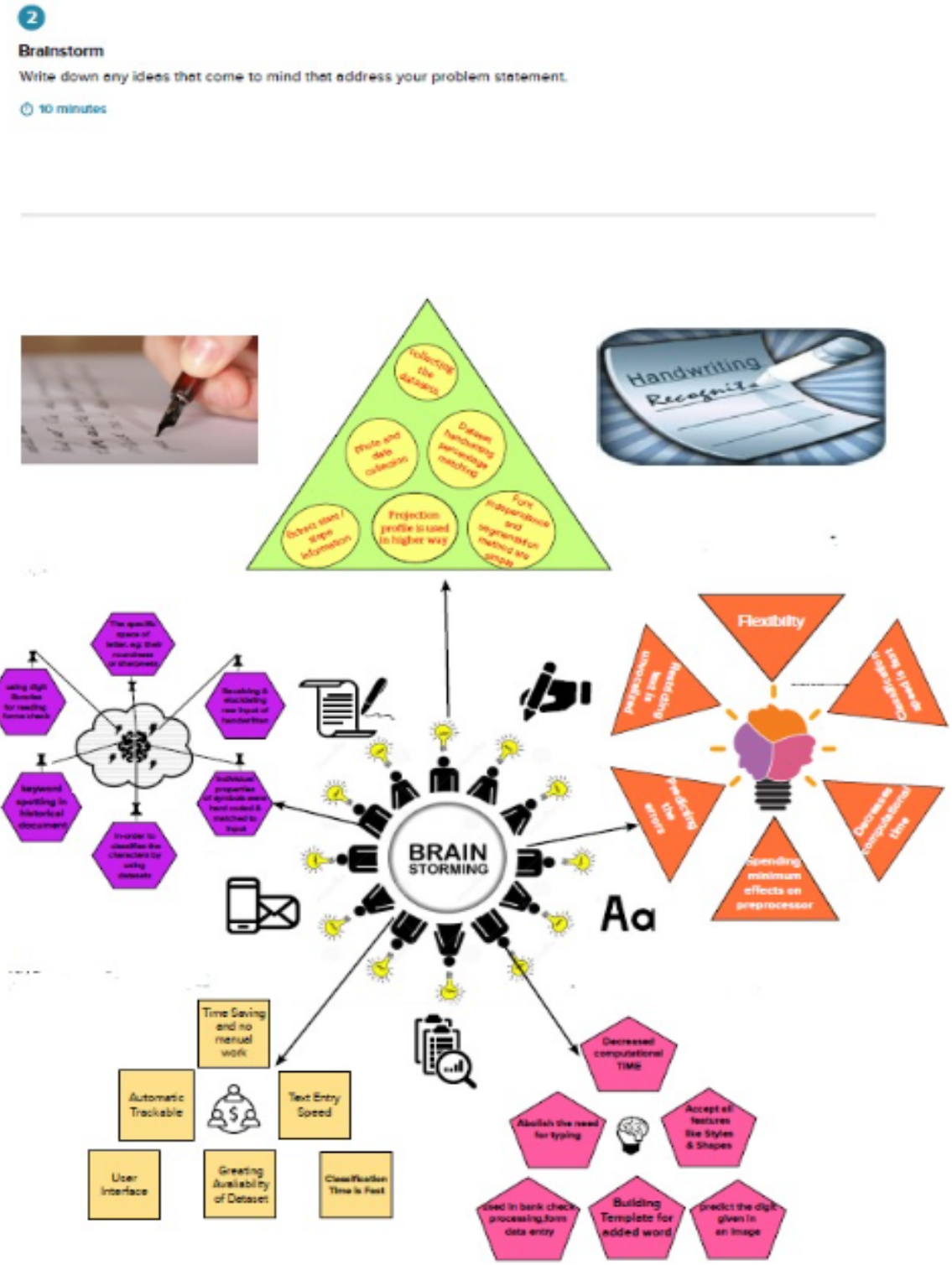
Handwrittendigitrecognitionistheabilityofcomputerstorecognizehuman handwrittendigits.Itisahardtaskforthemachinebecausehandwrittendigitsarenotperfectandcan bemadewithmanydifferentflavors.Thehandwrittendigitrecognitionisthesolutiontothisproblem whichusestheimageofadigitandrecognizesthedigitpresentintheimage.

Digitrecognitionsystemsarecapableofrecognizingthedigitsfromdifferentsourceslikeemail s,bankcheque,papers,images,etc.andindifferentreal-worldscenariosforonline handwritingrecognitiononcomputertabletsorsystem,recognizenumberplatesofvehicles,proc essingbankchequeamounts,numericentries informsfilledupby handandsoon.

CHAPTER3
IDEATIONANDPROPOSEDSOLUTION

3.1 EMPATHYMAPCANVAS

3.2 IDEATION&BRAINSTORMING

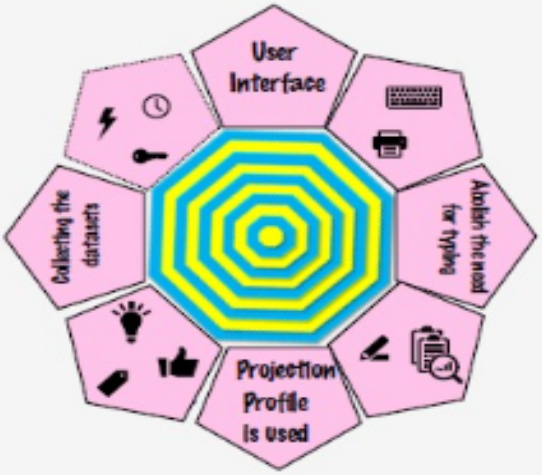


3

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.

20 minutes

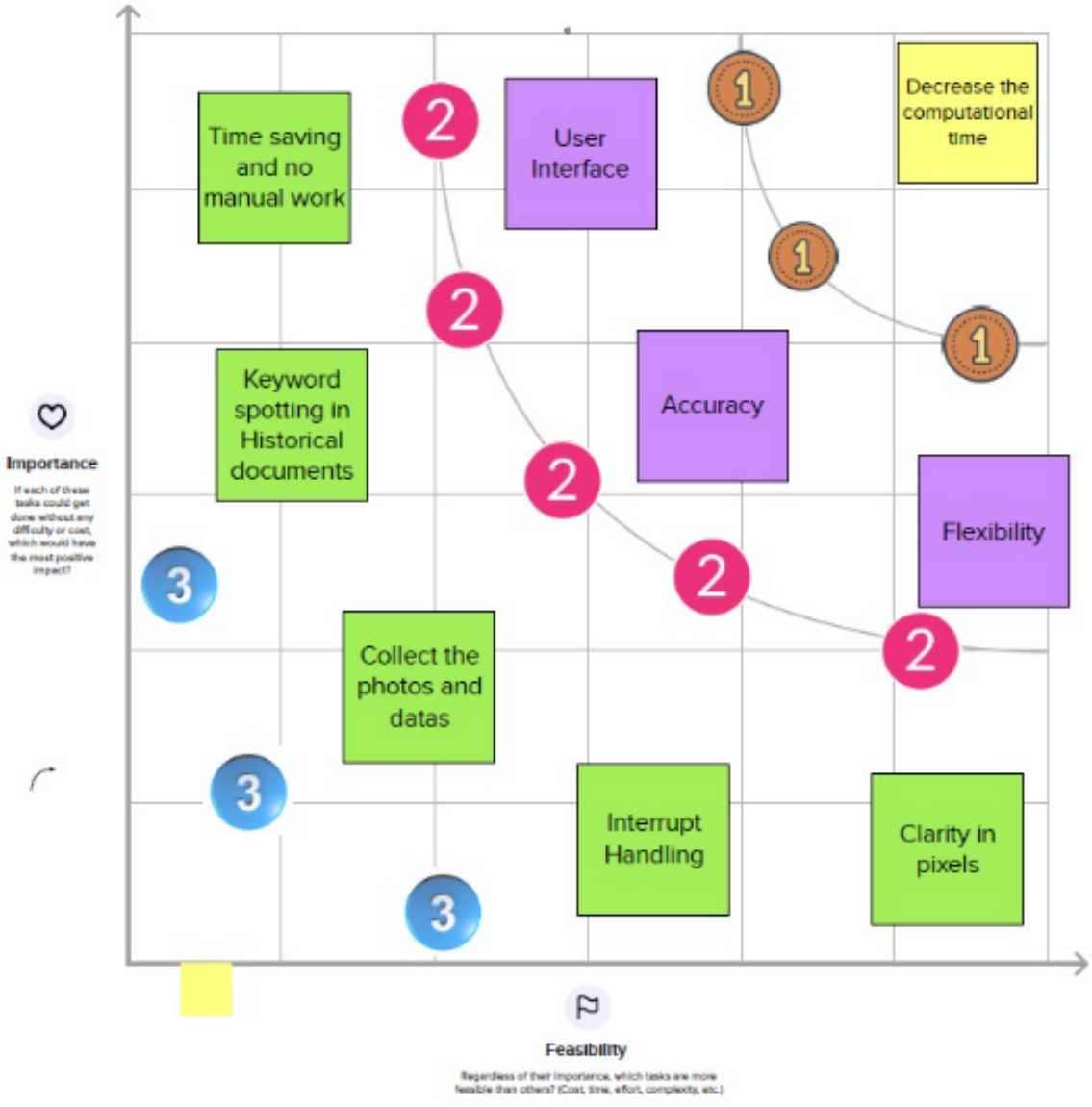


4

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes



3.3 PROPOSEDSOLUTION

S.NO	PARAMETER	DESCRIPTION
1	Problem Statement(Problemto besolved)	Manually written digits are of a different size,thickness,positionanddirection.Inthismanner,various difficulties must be considered todetermine the issue of handwritten digitrecognition
2	Idea / Solutiondescri ption	To solve this problem, we are going toimplement a classification algorithm torecognize the handwritten digits. This algorithmwill be effective in order to recognize digitswhichareofdifferentcompositions.
3	Novelty / Uniqueness	● Strategy for perceiving and arrangingtranscribeddigits. ● Canbeusedoffline ● Providedmoredatasetsformoreaccuracy ● The uniqueness and assortment in thecomposition styles of various individualsadditionally influence the example andpresenceofthedigits
4	Social Impact / Customer Satisfaction	The main social impact of this work is to ensureeffective and reliable approaches for recognitionof handwritten digits and make bankingoperations easier and error free. Customers willfeel at ease, as it is easy and convenient to use.Astheaccuracyisacceptable, thiscanhavemanyapplications.
5	Business Model(Revenue Model)	This novel method for Handwritten DigitRecognitionSystemcanbeapproachedbymanyindustries which needs this applicationincluding, programmed bank checks, postallocationsandtaxdocumentsandsoon. Humansrecognizing the handwritten digits with theirnaked eye can be difficult at times as it ofdifferent sizes, thickness, direction and ca alsolead to making errors due to these factors. Thisis when our proposed solutions comesinto help.We provide different data sets which helps inrecognitionwithaccuracy, sothathumanmakingerrorscanbeavoidedrespectively.
6	Scalability of theSolution	Financial and other business organizations suchas banks are facing issues in Recognizing writtendigitssuchasinchequesetc.Thiscanbehandled by our handwritten digit recognitionproject as they expand into different businessdomains without impacting performance. Ourproposedsolutionisscalableasitisdynamicandalsotrainedusing AI anddeeplearningModels

3.4 PROBLEMSOLUTIONFIT

TEAMID:PNT2022TMID35172

1.CUSTOMER SEGMENT(S): Here customers are the one who is defined to work with reading handwritten digits. They are present in places like bank, school, college, post offices, etc.,**.**

1. **JOBS-TO-BE-DONE / PROBLEMS :**There is a wide range of handwriting around the world. It is not possible to understand every handwriting precisely. It may lead to errors while dealing with rugged handwritings.

3.TRIGGERS: To quickly and precisely obtain the digits.

4. **EMOTIONS - BEFORE and AFTER:** Customers become irate and frustrated because they can't properly read the handwritten digits. They become confused and anxious as a result of not being able to finish their work on time.

5. **AVAILABLE SOLUTIONS :** Currently there are no popular programs and softwares to detect the handwritten digits.

6. **CUSTOMER CONSTRAINTS:** They believe such alternatives might result in mistakes and flaws and might not be practical.

7.BEHAVIOUR: Designing the best software that more quickly and accurately identifies the handwritten digits.

8.CHANNELS OF BEHAVIOUR: Utilizing software that is offered in the online market. Enlisting the assistance of nearby people in order to identify the numbers that their clients have scribbled.

9.PROBLEM ROOT CAUSE: Because handwritten number recognition is not an optical character recognition, there are numerous difficulties due to the wide variety of writing styles used by different people. Customers find it difficult to read the handwritten digits as different people use different writing styles and different languages. This investigation offers a thorough comparison of various deep literacy and machine literacy algorithms for handwritten number recognition.

10.**YOUR SOLUTION:** A novel method for handwritten digit recognition system helps in recognizing the handwritten digits that uses MNIST dataset for training the model. The model gets the image of the handwritten digit and recognizes the handwritten digit. Convolution neural networks algorithm is used over the MNIST dataset to recognize the handwritten digits

CHAPTER4
REQUIREMENTANALYSIS

4.1 FUNCTIONALREQUIREMENTS

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Sub Requirement (Story / Sub-Task)
FR-1	Image Data: Handwritten digit recognition refers to a computer's capacity to identify human handwritten digits from a variety of sources, such as photographs, documents, touch screens, etc., and categorise them into ten established classifications (0-9). In the realm of deep learning, this has been the subject of countless studies.
FR-2	Website: Web hosting makes the code, graphics, and other items that make up a website accessible online. A server hosts every website you've ever visited. The type of hosting determines how much space is allotted to a website on a server. Shared, dedicated, VPS, and reseller hosting are the four basic varieties.
FR-3	Digit Classifier Model: To train a convolutional network to predict the digit from an image, use the MNIST database of handwritten digits. get the training and validation data first.
FR-4	Cloud: The cloud offers a range of IT services, including virtual storage, networking, servers, databases, and applications. In plain English, cloud computing is described as a virtual platform that enables unlimited storage and access to your data over the internet.
FR-5	Modified National Institute of Standards and Technology dataset: The abbreviation MNIST stands for the MNIST dataset. It is a collection of 60,000 tiny square grayscale photographs, each measuring 28 by 28, comprising handwritten single digits between 0 and 9.

4.2 NONFUNCTIONALREQUIREMENTS

Non-functional Requirements:

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

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	One of the very significant problems in pattern recognition applications is the recognition of handwritten characters. Applications for digit recognition include filling out forms, processing bank checks, and sorting mail.
NFR-2	**Security**	1) The system generates a thorough description of the instantiation parameters, which might reveal information like the writing style, inaddition to a categorization of the digit. 2) The generative models are capable of segmentation driven by recognition. 3)The procedure uses a relatively.
NFR-3	**Reliability**	The samples are used by the neural network to automatically deduce rules for reading handwritten digits. Furthermore, the network may learn more about handwriting and hence enhance its accuracy by increasing the quantityof training instances. Numerous techniques and algorithms, such as Deep Learning/CNN, SVM, Gaussian Naive Bayes, KNN, Decision Trees, Random Forests, etc., can be used to recognise handwritten numbers.
NFR-4	**Accuracy**	With typed text in high-quality photos, optical character recognition (OCR) technology offers accuracy rates of greater than 99%. However, variances in spacing, abnormalities in handwriting, and the variety of human writingstyles result in less precise character identification.
NFR-5	**Availability**	Availability of system functionality andservices for use

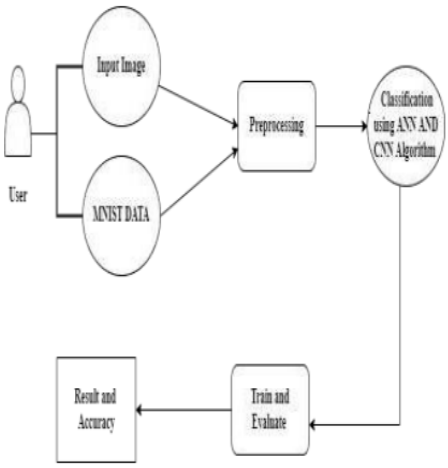
CHAPTER5
PROJECTDESIGN

5.1 DATAFLOWDIAGRAM

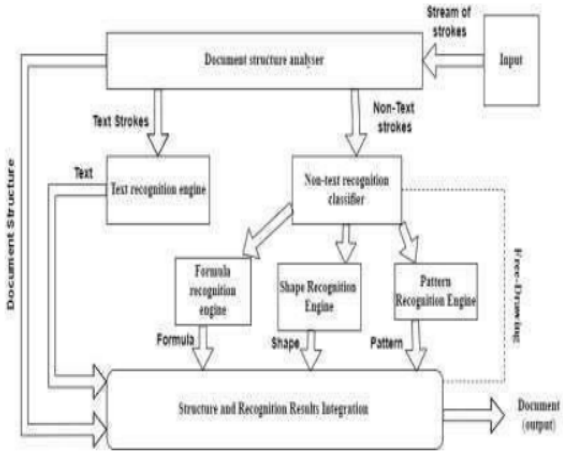
Data Flow Diagrams:

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: (Simplified) FLOW



Example:DFDLevel0(IndustryStandard)



5.2 SOLUTION&TECHNICALARCHITECTURE

Technical Architecture:

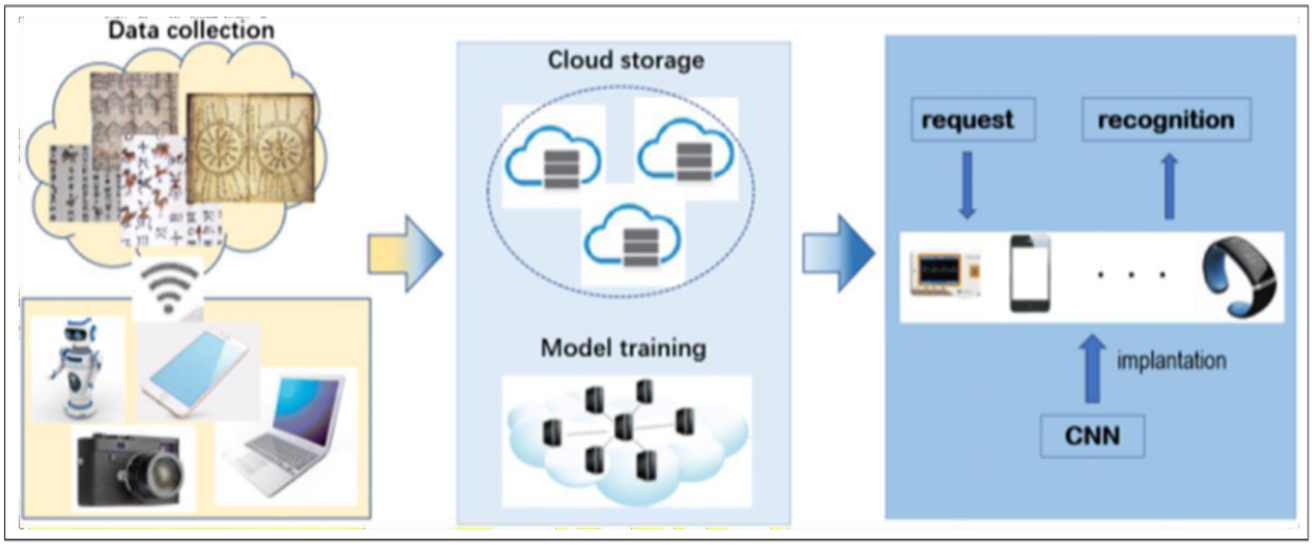


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g. Web UI, Mobile App, Chatbot etc.	HTML, CSS, JavaScript
2.	Application Logic-1	Logic for a process in the application	Python
3.	Application Logic-2	Logic for a process in the application	IBM Watson STT service
4.	Application Logic-3	Logic for a process in the application	IBM Watson Assistant
5.	Database	Data Type, Configurations etc.	MySQL, NoSQL, etc.
6.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant
7.	File Storage	File storage requirements	IBM Block Storage
8.	External API-1	Purpose of External API used in the application	IBM Weather API
9.	External API-2	Purpose of External API used in the application	Aadhar API
10.	Machine Learning Model	Purpose of Machine Learning Model	Object Recognition Model
11.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration Cloud Server Configuration	Local, Cloud Foundry

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used	Technology of Opensource framework
2.	Security Implementations	List all the security / access controls implemented, use of firewalls etc.	SHA-256, Encryptions, IAM Controls, OWASP

S.No	Characteristics	Description	Technology
3.	Scalable Architecture	Justify the scalability of architecture	3 – tier, Micro-services
4.	Availability	Abstract and Figures. The features for handwritten digit recognition have been introduced. These features are based on shape analysis of the digit image and extract slant or slope information. They are effective in obtaining good recognition accuracies	Distributed servers, IBM cloud
5.	Performance	The standard implementations of neural networks achieve an accuracy of ~ (98–99) percent in correctly classifying the handwritten digits.	number of requests per sec, use of Cache, use of CDN's

5.3 USERSTORIES

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 view the guide and awareness 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 read the instructions to use this 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 and choose the image to be uploaded	I can upload and choose the image from the system storage and also in any virtual storage.	Medium	Sprint-3
		USN-7	As a user, I will train and test the input to get the maximum accuracy of output.	I can able to train and test the application until it 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 MNIST data 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 the web app.	I can view the awareness of this application and its limitations.	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 and awareness 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 read the instructions to use this 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 application virtually anywhere.	I can use the application portably anywhere.	High	Sprint-1
		USN-11	As it is an open source, can use it cost freely.	I can use it without any payment to be paid for it to access.	Medium	Sprint-2
		USN-12	As it is a web application, it is installation free	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 and choose the image to be uploaded	I can upload and choose the image from the system storage and also in any virtual storage.	Medium	Sprint-3

CHAPTER6
PROJECTPLANNINGANDSCHEDULING

6.1 SPRINTPLANNINGANDESTIMATION

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data Collection & preprocessing	USN-1	As a user, I can upload any kind of image withthe pre-processing step is involved in it.	10	High	Kanaparathi jayakrishna, Murali R
Sprint-1		USN-2	As a user, I can upload the image in anyresolution.	10	High	Vignesh G, Raghuraj S V
Sprint-2	Building the Machinelearning model	USN-3	As a user, I will get a application with MLmodel which provides high accuracy of recognized handwritten digit	3	Medium	Vignesh G, Murali R
Sprint-2		USN-4	As a user, I can pass the handwritten digitimage for recognizing the digit.	2	Medium	Vignesh G, Murali R
Sprint-2		USN-5	As a user, I can get the most suitablerecognized digit.	10	High	Kanaparathi jayakrishna, Murali R
Sprint-3	Building User Interface Application	USN-6	As a user, I will upload the handwritten digit image to the application by clicking a uploadbutton.	8	Medium	Raghuraj S V,Kanaparathi jayakrishna
Sprint-3		USN-7	As a user, I can know the details of thefundamental usage of the application.	2	High	Raghuraj S V,Kanaparathi jayakrishna
Sprint-3		USN-8	As a user, I can see the predicted / recognizeddigits in the application	10	Medium	Vignesh G, Murali R
Sprint-4	Train and deployment of modelin IBM Cloud	USN-9	As a user, I can access the web application andmake the use of the product from anywhere	20	High	Vignesh,Muarli, Kanaparathi Jayakrishna, Raghuraj S V

6.2 SPRINTDELIVERYSCHEDULE

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	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

CHAPTER7

CODING&SOLUTIONING

Import Required Librauries

In [52]:
```
import numpy as np
import tensorflow as tf
import matplotlib.pyplot as plt
from keras.utils import np_utils
import tensorflow as tf
from tensorflow.keras.layers import Conv2D, Dense, Flatten
```

In [2]:
```
print(tf.__version__)
```

```
2.9.2
```

In [3]:
```
mnist_ds = tf.keras.datasets.mnist.load_data()
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz
11490434/11490434 [==============================] - 0s 0us/step
```

In [4]:
```
mnist_ds
```

Building The Model

In [53]:
```
model = tf.keras.models.Sequential([tf.keras.layers.Conv2D(64, (3, 3), input_shape=(28, 28, 1), activation="relu"),
                                    tf.keras.layers.Conv2D(32, (3, 3), activation="relu"),
                                    tf.keras.layers.Flatten(),
                                    tf.keras.layers.Dense(10, activation=tf.nn.softmax)])
```

In [54]:
```
model.compile(loss='categorical_crossentropy', optimizer="Adam", metrics=["accuracy"])
```

In [55]:
```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 26, 26, 64)        640

 conv2d_1 (Conv2D)           (None, 24, 24, 32)        18464

 flatten (Flatten)           (None, 18432)             0

 dense (Dense)               (None, 10)                184330

=================================================================
Total params: 203,434
Trainable params: 203,434
Non-trainable params: 0
_________________________________________________________________
```

Training The Model

In [56]:
```
model.fit(training_images, training_labels, batch_size=32, epochs=5, validation_data=(test_images,test_labels))
```

```
Epoch 1/5
1875/1875 [==============================] - 16s 4ms/step - loss: 0.1254 - accuracy: 0.9625 - val_loss: 0.0519 - val_accuracy: 0.9831
Epoch 2/5
1875/1875 [==============================] - 7s 4ms/step - loss: 0.0464 - accuracy: 0.9861 - val_loss: 0.0385 - val_accuracy: 0.9868
Epoch 3/5
1875/1875 [==============================] - 7s 4ms/step - loss: 0.0296 - accuracy: 0.9907 - val_loss: 0.0409 - val_accuracy: 0.9872
Epoch 4/5
1875/1875 [==============================] - 7s 4ms/step - loss: 0.0201 - accuracy: 0.9937 - val_loss: 0.0403 - val_accuracy: 0.9878
Epoch 5/5
1875/1875 [==============================] - 7s 4ms/step - loss: 0.0139 - accuracy: 0.9957 - val_loss: 0.0457 - val_accuracy: 0.9872
```

Out[56]:

Test The Model

In [58]:
```
metrics = model.evaluate(test_images, test_labels, verbose=0)

print("Test Loss -> {}  \nTest Accuracy -> {}".format(metrics[0],metrics[1]))
```

```
Test Loss -> 0.04573516175150871
Test Accuracy -> 0.9872000217437744
```

In [67]:
```
model.predict(test_images[2:8])
```

```
1/1 [==============================] - 0s 15ms/step
```

Out[67]:
```
array([[2.32068427e-08, 9.99983430e-01, 8.10439190e-07, 1.28179977e-07,
        9.64922492e-06, 1.83879649e-06, 1.62838049e-07, 1.56461965e-06,
        2.34936374e-06, 3.22469944e-08],
       [9.99998927e-01, 1.04071238e-13, 7.69856399e-07, 1.84226245e-09,
        3.37900985e-13, 4.71777106e-09, 8.84182239e-09, 2.02508791e-11,
        3.22932721e-07, 9.56373647e-09],
       [8.73478698e-13, 4.26847549e-13, 1.15858136e-10, 3.97662771e-11,
        9.99999881e-01, 2.68545906e-12, 8.56604648e-11, 7.41609482e-11,
        4.87753553e-08, 1.05269102e-07],
       [2.15423035e-09, 9.99581635e-01, 2.15949945e-06, 1.08863398e-08,
        2.10376020e-05, 2.63231090e-08, 3.26978977e-08, 3.81208694e-04,
        1.27898356e-05, 1.17525337e-06],
       [2.41032138e-18, 9.36788008e-11, 3.97475330e-10, 3.54850779e-13,
        9.99299288e-01, 6.94019908e-09, 6.61158953e-14, 4.28452246e-10,
        7.00477336e-04, 2.29253416e-07],
       [1.89875802e-16, 2.21634187e-11, 1.76986703e-09, 2.65193867e-09,
        2.56875592e-06, 2.88839956e-08, 1.60236594e-14, 2.67538752e-11,
        5.46009005e-06, 9.99991894e-01]], dtype=float32)
```

In [74]:
```
history=model.predict(np.array([test_images[7]]))
history
```

```
1/1 [==============================] - 0s 17ms/step
```

Out[74]:
```
array([[1.8987580e-16, 2.2163419e-11, 1.7698670e-09, 2.6519387e-09,
        2.5687532e-06, 2.8883996e-08, 1.6023692e-14, 2.6753875e-11,
        5.4600901e-06, 9.9999189e-01]], dtype=float32)
```

In [75]:
```
np.argmax(history, axis=1)
```

Out[75]:
```
array([9])
```

In [73]:
```
#It predicted as 9
```

Let us see , It is correct or not?

In [78]:
```
tl=test_labels[7]
tl
```

Out[78]:
```
array([0., 0., 0., 0., 0., 0., 0., 0., 0., 1.], dtype=float32)
```

In [81]:
```
np.argmax(tl)
```

Out[81]:
```
9
```

It Predicted Correctly!!!

CHAPTER8
TESTING

8.1 TESTCASES

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 g, muvali r
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	Raghucrai sv, lavakrishna k
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 g, muvali r
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 g, muvali r
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		Raghucrai sv, lavakrishna k
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 results page	Sample 1.png	All the routes should properly work	Working as expected	PASS		Raghucrai sv, lavakrishna k
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		Raghucrai sv, lavakrishna k

M_TC_002	Functional	Model	Check if the model predicts the digit	1) Open the page 2) Click on select button 3) Select the input image 4) Check the results	Sample 1.png	The model should predict the number	Working as expected	PASS		Vignesh g, murali r
M_TC_003	Functional	Model	Check if the model can handle complex input image	1) Open the page 2) Click on select button 3) Select the input image 4) Check the results	Complex Sample.png	The model should predict the number in the complex image	The model fails to identify the digit since the model is not built to handle such data	FAIL	BUG_M_001	Vignesh g, murali r
RP_TC_001	UI	Result Page	Verify UI elements in the Result Page	1) Open the page 2) Click on select button 3) Select the input image 4) Check if all the UI elements are displayed properly	Sample 1.png	The Result page must be displayed properly	Working as expected	PASS		Vignesh g, murali r
RP_TC_002	UI	Result Page	Check if the input image is displayed properly	1) Open the page 2) Click on select button 3) Select the input image 4) Check if the input image are displayed	Sample 1.png	The input image should be displayed properly	The size of the input image exceeds the display container	FAIL	BUG_RP_001	Raghuraj sv, lavakrishna k
RP_TC_003	UI	Result Page	Check if the result is displayed properly	1) Open the page 2) Click on select button 3) Select the input image 4) Check if the result is displayed	Sample 1.png	The result should be displayed properly	Working as expected	PASS		Vignesh g, murali r
RP_TC_004	UI	Result Page	Check if the other predictions are displayed properly	1) Open the page 2) Click on select button 3) Select the input image 4) Check if all the other predictions are displayed	Sample 1.png	The other predictions should be displayed properly	Working as expected	PASS		Vignesh g, murali r

8.2 USERACCEPTANCETESTING

8.2.1 DEFECTANALYSIS

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

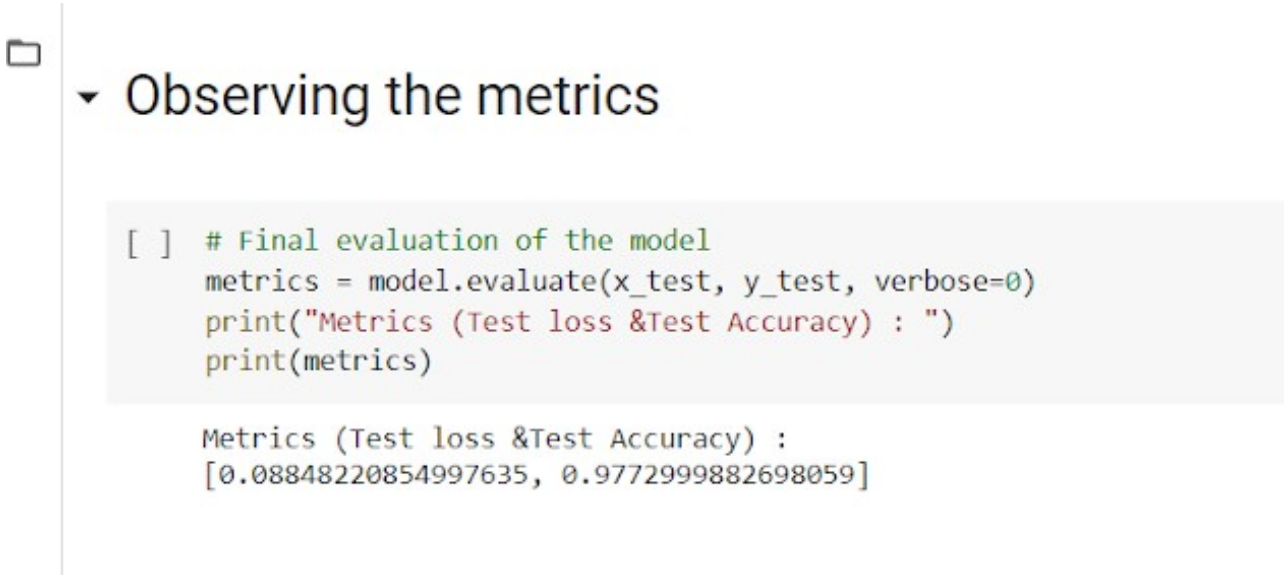
8.8.2 TESTCASEANALYSIS

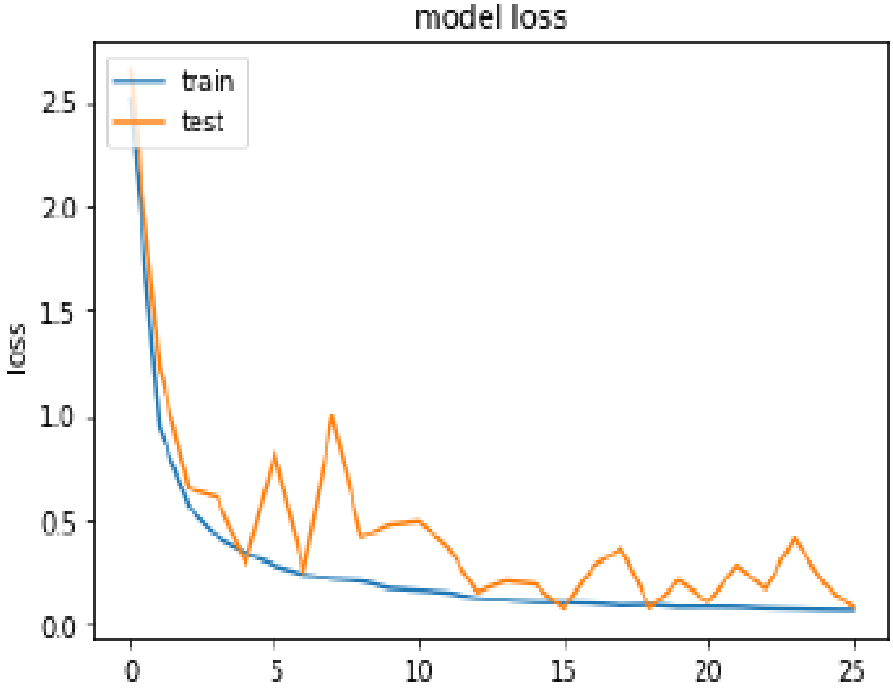
Section	Total Cases	Not Tested	Fail	Pass
Client Application	10	0	3	7
Security	2	0	1	1
Performance	3	0	1	2
Exception Reporting	2	0	0	2

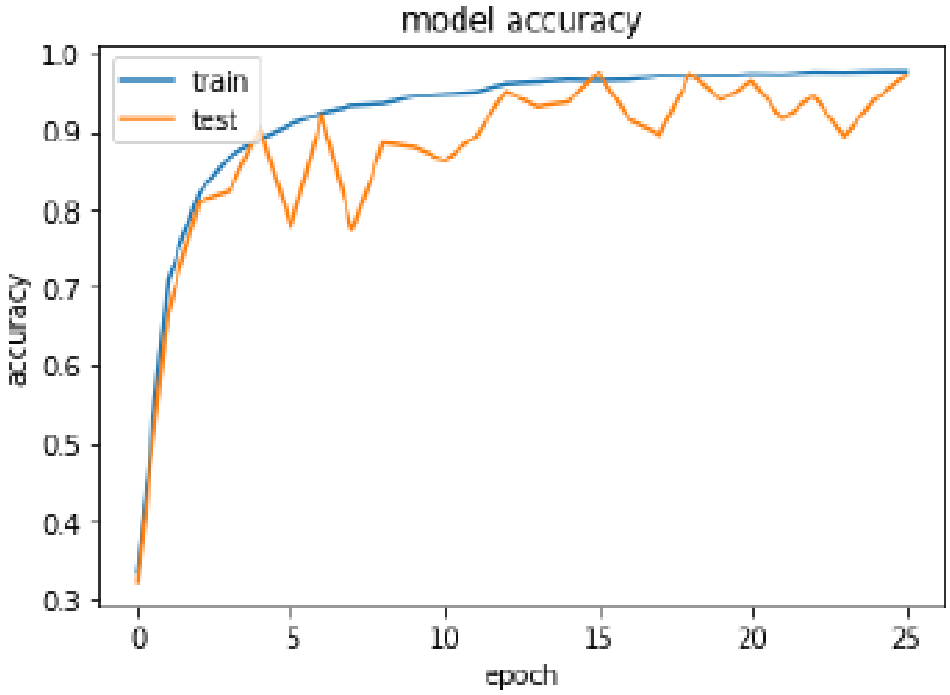
CHAPTER9

RESULTS

9.1 PERFORMANCEMETRICS




model loss
train
test
loss
2.5
2.0
1.5
1.0
0.5
0.0
0
5
10
15
20
25


model accuracy
train
test
accuracy
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0
5
10
15
20
25
epoch

CHAPTER10
ADVANTAGES & DISADVANTAGES

ADVANTAGES

- Reduces manual work
- More accurate than average human
- Capable of handling a lot of data
- Can be used anywhere from any device

DISADVANTAGES

- Can not handle complex data
- All the data must be in digital format
- Requires a high performance server for faster predictions
- Prone to occasional errors

CHAPTER11
CONCLUSION

This project demonstrated a web application that uses machine learning to recognize Handwritten numbers.Flask, HTML, CSS, JavaScript, and a few other technologies were used to create this project. The model predicts the handwritten digit using aC NN network. During Testing, the model achieved a 99.61% recognition rate.The proposed project is scalable and can easily handle a huge number of users. Since it is a webapplication, it is compatible with any device that can run a browser.This project is extremely useful in real-world scenarios such as recognizing number plates of vehicles,
Processing bank cheque amounts, numeric entries in forms filled up by hand(taxforms) and so on. There is so much room for improvement, which can be implemented in subsequentversions.

CHAPTER12
FUTURESCOPE

This project is far from complete and there is a lot of room for improvement. Some of the improvements that can be made to this project are as follows:

- Add support to detect from digits multiple images and save the results
- Add support to detect multiple digits
- Improve model to detect digits from complex images
- Add support to different languages to help users from all over the world

This project has endless potential and can always be enhanced to become better. Implementing this concept in the real world will benefit several industries and reduce the workload on many workers, enhancing overall work efficiency.

APPENDIX

SOURCECODE

MODELCREATION

Building The Model

In [53]:
```
model = tf.keras.models.Sequential([tf.keras.layers.Conv2D(64, (3, 3), input_shape=(28, 28, 1), activation="relu"),
                                    tf.keras.layers.Conv2D(32, (3, 3), activation="relu"),
                                    tf.keras.layers.Flatten(),
                                    tf.keras.layers.Dense(10, activation=tf.nn.softmax)])
```

In [54]:
```
model.compile(loss='categorical_crossentropy', optimizer="Adam", metrics=["accuracy"])
```

In [55]:
```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 26, 26, 64)        640

 conv2d_1 (Conv2D)           (None, 24, 24, 32)        18464

 flatten (Flatten)           (None, 18432)             0

 dense (Dense)               (None, 10)                184330

=================================================================
Total params: 203,434
Trainable params: 203,434
Non-trainable params: 0
```

Training The Model

In [56]:
```
model.fit(training_images, training_labels, batch_size=32, epochs=5, validation_data=(test_images,test_labels))
```

```
Epoch 1/5
1875/1875 [==============================] - 16s 4ms/step - loss: 0.1254 - accuracy: 0.9625 - val_loss: 0.0519 - val_accuracy: 0.9831
Epoch 2/5
1875/1875 [==============================] - 7s 4ms/step - loss: 0.0464 - accuracy: 0.9861 - val_loss: 0.0385 - val_accuracy: 0.9868
Epoch 3/5
1875/1875 [==============================] - 7s 4ms/step - loss: 0.0296 - accuracy: 0.9907 - val_loss: 0.0409 - val_accuracy: 0.9872
Epoch 4/5
1875/1875 [==============================] - 7s 4ms/step - loss: 0.0201 - accuracy: 0.9937 - val_loss: 0.0403 - val_accuracy: 0.9878
Epoch 5/5
1875/1875 [==============================] - 7s 4ms/step - loss: 0.0139 - accuracy: 0.9957 - val_loss: 0.0457 - val_accuracy: 0.9872
```

Out[56]:

Test The Model

In [58]:
```
metrics = model.evaluate(test_images, test_labels, verbose=0)

print("Test Loss -> {}  \nTest Accuracy -> {}".format(metrics[0],metrics[1]))
```

```
Test Loss -> 0.04573516175150871
Test Accuracy -> 0.9872000217437744
```

FLASK APP

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

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


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


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

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

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

        pred = model.predict(im2arr)

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

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


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

Home Page (HTML)

```
<body>
  <section align='center'>
  <h1 class="welcome">IBM PROJECT
  <div align="'center" id="team_id">TEAM ID :PNT2022TMID35172 </div>
  </h1>
  </section>

  <section id="title">
    <h4 class="heading">Handwritten Digit Recognition Web Application</h4>
    <br><br>
      <p>
        The website is designed to predict the handwritten digit.
      </p>
    <p>
        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 uses the image of a digit and recognizes the digit present in the image. </p>
      <br>
    <p> Hence, our handwritten digit recognition  web-app is a real-time applications.
        MNIST data set is widely used for this recognition process and it has 70000 handwritten digits.
        We use Artificial neural networks particularly cnn 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 analyzed by the model and the detected result is returned on to UI</p>
  </section>
  <section id="content" align='center'>
        <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>
          <img id="frame" src="" width="200px" height="200px"/>
          <div class="buttons_div">
            <button type="submit" class="btn btn-dark" id="predict_button">Predict Number</button>
            <button type="button" class="btn btn-dark" id="clear_button">  Clear  </button>
          </div>
        </form>
        </div>
  </section>

</body>
```

Prediction page(HTML)

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Prediction</title>
</head>
<style>
    body{
     background-image: url({{ url_for('static',filename='images.png') }});
     background-repeat: no-repeat;
     background-size: cover;
    }

    #rectangle{
     width:400px;
     height:150px;
     background-color: #5796a5;
     opacity: 0.8;
     border-radius: 25px;
     position:absolute;
     top:25%;
     left:50%;
     transform:translate(-50%,-50%);
    }
    #ans{
  text-align: center;
  font-size: 40px;
  margin: 0 auto;
  padding: 3% 5%;
  padding-top: 9%;
  color: white;
    }
</style>
<body>
    <div id="rectangle">
        <h1 id="ans">Predicted Number : {{num}}</h1>
    </div>
</body>
</html>
```

Style sheet (CSS)

```css
#clear_button{
  margin-left: 15px;
  font-weight: bold;
  color: blue;
}
#confidence{
  font-family: 'Josefin Sans', sans-serif;
  margin-top: 7.5%;
}
#content{
  background-color: blanchedalmond;
  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: center;
    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%;
}
```

```css
#result{
  font-size: 5rem;
}
#title{
  background-color: aquamarine;
  padding: 1.5% 15%;
  margin: 0 auto;
  text-align: center;
}
.btn {
    font-size: 15px;
    padding: 10px;
    -webkit-appearance: 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%;
}
```

Javascript(js)

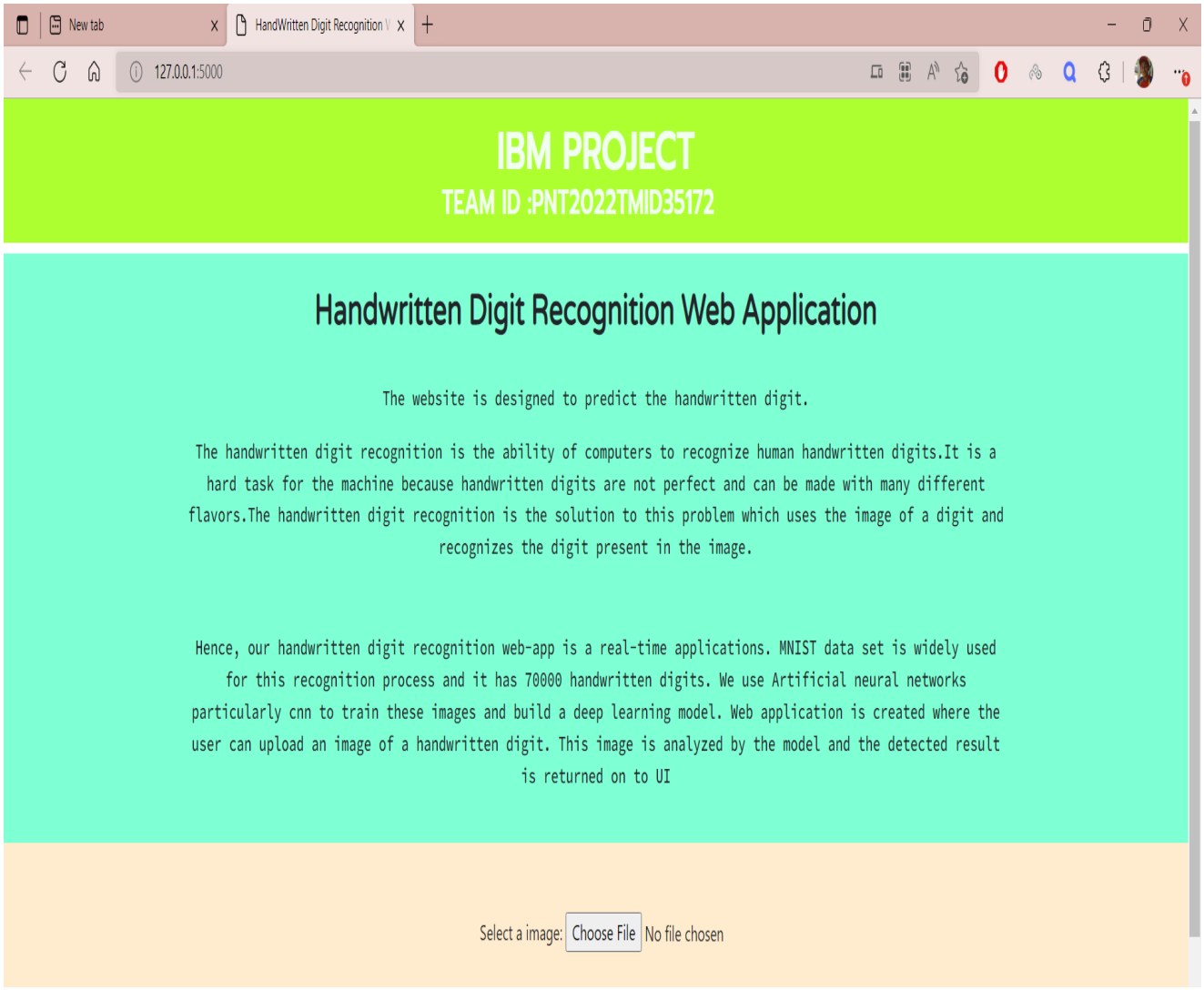
```
<script>
  function preview() {
    frame.src=URL.createObjectURL(event.target.files[0]);
}


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

</script>
```

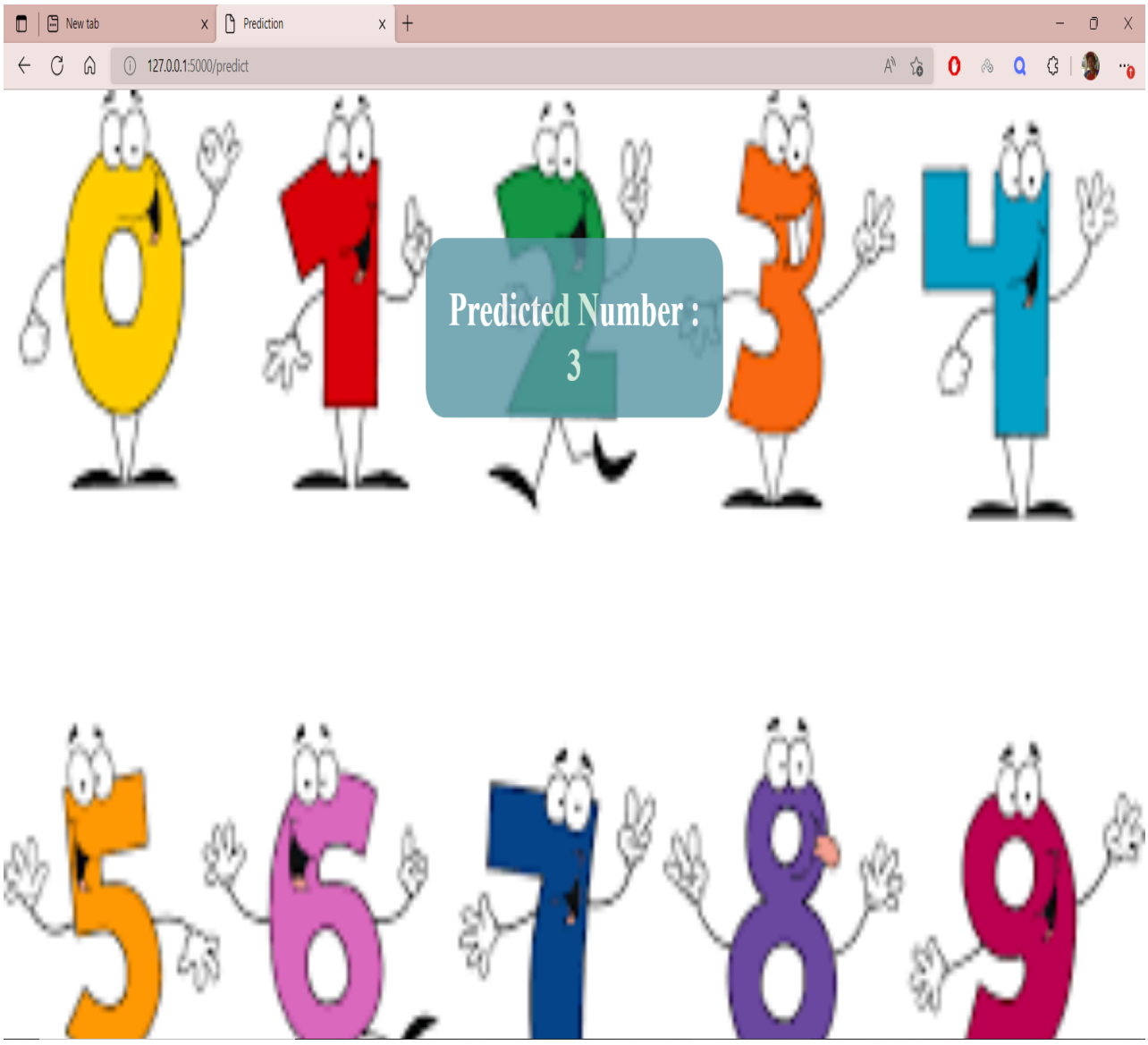
Output Screenshots:

Home page




New tab
HandWritten Digit Recognition
127.0.0.1:5000
The website is designed to predict the handwritten digit.
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 uses the image of a digit and recognizes the digit present in the image.
Hence, our handwritten digit recognition web-app is a real-time applications. MNIST data set is widely used for this recognition process and it has 70000 handwritten digits. We use Artificial neural networks particularly cnn 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 analyzed by the model and the detected result is returned on to UI
Select a image:
Choose File
images.png
Predict Number
Clear

Prediction page output



GITHUB

https://github.com/IBM-EPBL/IBM-Project-50960-1660955991

PROJECT DEMO

1.https://github.com/IBM-EPBL/IBM-Project-50960-1660955991/blob/main/Final%20Deliverables/Project%20Demo/Project-demo.mp4

2. https://youtu.be/b8pQb2JKvjo