A Novel Method for Handwritten Digit Recognition System
PROJECT REPORT

Submitted by

ARUL JOSHUA J	**(130719104006)**
ARUL JOTHI B	**(130719104007)**
BEAUTIN GOLD G J	**(130719104012)**
CALVIN JOEL RAJAA S	**(130719104014)**
KEERTHIVASAN K	**(130719104044)**

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

JERUSALEM COLLEGE OF ENGINEERING

CHENNAI 600 100

CONTENTS

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

6.2 Sprint Delivery Schedule

6.3 Reports from JIRA

7. CODING & SOLUTIONING (Explain the features added in the project along with code)

7.1 Feature 1

7.2 Feature 2

7.3 Database Schema (if Applicable)

8. TESTING

8.1 Test Cases

8.2 User Acceptance Testing

9. RESULTS

9.1 Performance Metrics

10. ADVANTAGES & DISADVANTAGES

11. CONCLUSION

12. FUTURE SCOPE

13. APPENDIX

Source Code

GitHub Link

1. INTRODUCTION

1.1 PROJECT OVERVIEW

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

1.2 PURPOSE

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 realworld scenarios for online handwriting recognition on computer tablets or system, recognize number plates of, numeric entries in forms filled up by hand and soon. Handwritten character recognition is one of the practically important issues in pattern recognition applications. The main purpose of this project is to build an automatic handwritten digit recognition method for the recognition of handwritten digit string s. To accomplish the recognition task, first, the digits will be segmented into individual digits. Then, a digit recognition module is employed to classify each segmented digit completing the handwritten digit string recognition task. The applications of digit recognition include postal mail sorting, bank check processing, form data entry, etc. The

heart of the problem lies within the ability to develop an efficient algorithm that can recognize handwritten digits and which is submitted by users by the way of a scanner

2. Literature survey

2.1 EXISTING PROBLEM

Hand writing recognition of characters has been around since the 1980s.The task of handwritten digit recognition, using a classifier, has great importance and use such as online handwriting recognition on computer tablets, recognize zip codes on mail for postal mail sorting, processing bank check amounts, numeric entries in forms filled up by hand (for example tax forms) and soon. There are different challenges faced while attempt to solve problem. The handwritten digits are not always of the same size, thickness, or orientation and position relative to the margins. Our goal was to implement a pattern classification method to recognize the handwritten digits provided in the MINIST data set of images of hand written digits (0 to 9). The data set used for our application is composed of 300 training images and 300 testing

images, and is a subset of the MNIST data set. Handwriting digits and character recognitions

have become increasingly important in today's digitized world due to their practical applications in various day to day activities. It can be proven by the fact that in recent years, different recognition systems have been developed or proposed to be used in different fields where high classification efficiency is needed. Systems that are used to recognize Handwriting letters, characters, and digits help people to solve more complex tasks that otherwise would be time consuming and costly. A good example is the use of automatic processing systems used in banks to process bank cheques. Without automated bank cheque processing systems, the bank would be required to employ many employees who may not be as efficient as the computerized processing system.

2.2 REFERENCES

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

2. Sandhya Arora , used four feature extraction techniques namely, intersection, shadow feature, chain code histogram and straight line fitting features. Shadow features are computed

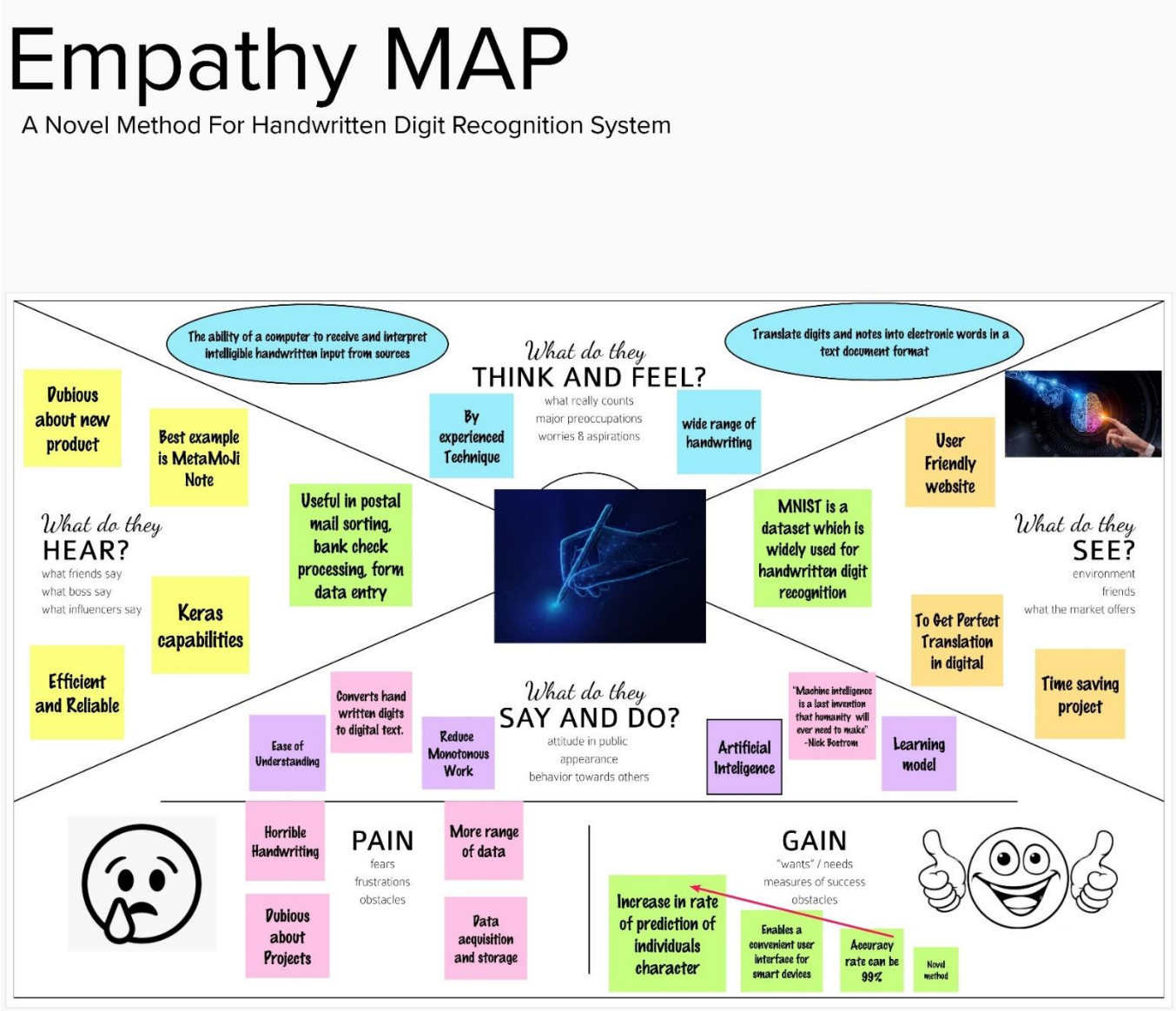
globally for character image while intersection features, chain code histogram features and line fitting features are computed by dividing the character image into different segments. On experimentation with a dataset of 49 was 92.80% for Devanagari characters. 00 samples the overall recognition rate observed

2.3 Problem statement definition

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

3. IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS



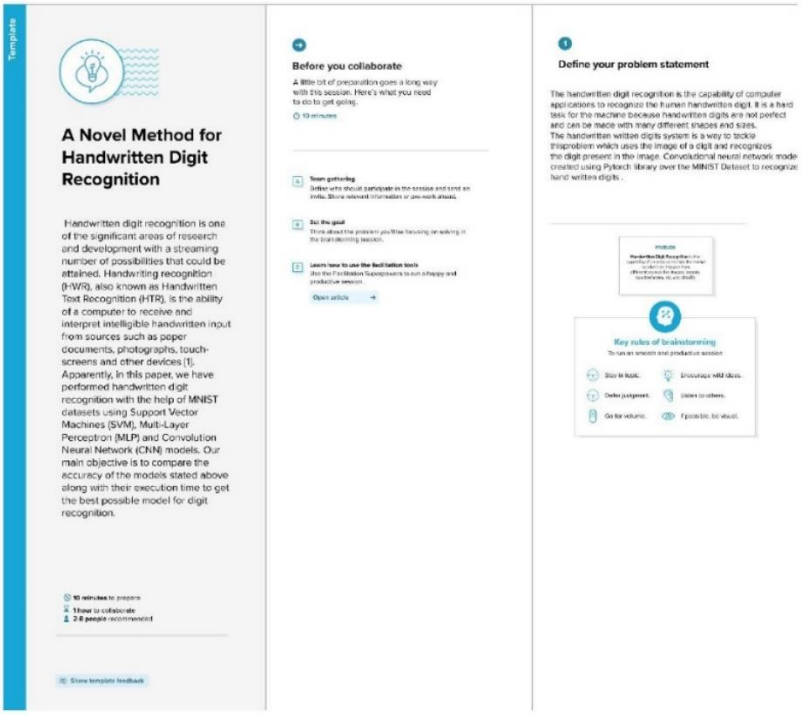
3.2 IDEATION AND BRAINSTORMING

Ideation Phase
Brainstorm & Idea Prioritization Template

Date	21 October 2022
Team ID	PNT2022TMID07039
Project Name	A Novel method for Handwritten Digit Recognition System
Maximum Marks	4 Marks

Brainstorm & Idea Prioritization Template:

Step-1: Team Gathering, Collaboration and Select the Problem Statement



Step-2: Brainstorm, Idea Listing and Grouping

Step-3: Idea Prioritization

3.3 PROPOSED SOLUTION

Project Design Phase-I
Proposed Solution Template

Date	21 October 2022
Team ID	PNT2022TMID07039
Project Name	Project – A Novel Method For Handwritten Digit Recognition System.
Maximum Marks	2 Marks

Proposed Solution Template:

Project team shall fill the following information in proposed solution template.

S.NO.	Parameter	Description
1.	Problem Statement (Problem to be solved)	It is easy for the human to perform a task accurately by practicing it repeatedly and memorizing it for the next time. Human brain can process and analyze images easily. Also, recognize the different elements present in the images. In this project, the goal is to correctly identify digits from a dataset of tens of thousands of handwritten images and experiment with different algorithms to learn first-hand what works well and how techniques compare.
2.	Idea / Solution description	The task of HDR is accomplished by using the CNN, incorporating a sequential CNN framework, with rectified linear units (RELU) activations that have never been reported. The goal is achieved by establishing a model that can recognize and determine the handwritten digits from its image with high accuracy and low computation time. We aim to complete this by using the concepts of convolutional neural network. The proposed CNN framework is well equipped with suitable parameters for high accuracy of MNIST digit classification.
3.	Novelty / Uniqueness	The Handwritten digits are not always of the same size, width, orientation and justified to margins as they differ from writing of person to person. There is similarity between numbers. So, classifying between these numbers is also a major problem for computers. The uniqueness and variety in the handwriting of different individuals also influence the formation and appearance of the digits.
4.	Social Impact / Customer Satisfaction	Character recognition plays an important role in the modern world. It can solve more complex problems and makes humans' job easier. An

		example is handwritten character recognition. This is a system widely used in the world to recognize zip code or postal code for mail sorting. There are different techniques that can be used to recognize handwritten characters. Two techniques researched in this paper are Pattern Recognition and Artificial Neural Network (ANN). Both techniques are defined and different methods for each technique is also discussed.
5.	Business Model (Revenue Model)	In recent days, Artificial Neural Network (ANN) can be applied to a vast majority of fields including business, medicine, engineering, etc. The most popular areas where ANN is employed nowadays are pattern and sequence recognition, novelty detection, character recognition, regression analysis, speech recognition, image compression, stock market prediction, Electronic nose, security, loan applications, data processing, robotics, and control.
6.	Scalability of the Solution	It is flexible and suitable for text and document format. It has high speed ,robustness ,etc.

3.4 PROBLEM SOLUTION FIT

Project Title: A Novel method for Handwritten Digit Recognition System

project Design Phase-I - Solution Fit Template **Team ID: PNT2022TMIDO7039**

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Customer who need to identify the digit from handwritten form

6. CUSTOMER CONSTRAINTS CC

It requires much more computation than more standard OCR techniques

5. AVAILABLE SOLUTIONS AS

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

Explore AS, differentiate

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

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

Diffferent people handwriting varies from each other and they struggle to identify

9. PROBLEM ROOT CAUSE RC

From the number 0 to 9 it's shapes and design are vary.Further according to individual person their handwriting also varies.Thus this handwritten digit recognition is needed

7. BEHAVIOUR BE

The output of an OCR run for an clear image and comparing it to the original version of the same text gives good accuracy

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

Identify strong TR & EM

3. TRIGGERS TR

While they recognition the handwritten digit

4. EMOTIONS: BEFORE / AFTER EM

Dilemma,exhausted into satified ,hopeful and comfort

10. YOUR SOLUTION SL

Neural Network is used to recognise and predict the handwritten digits.Dataset are trained using gradient descent back propagation algorithm and tested using the feed forward algorithm.

Observing the system performance with variation of number of hidden units and iteration.Using this method, digits recognised and its accuracy will be high upto 99% .So we get good output

8.CHANNELS of BEHAVIOUR CH

8.1.ONLINE

Here extract from block

8.2.OFFLINE

Here extract from different user for handwriting

Extract online & offline CH of BE

4. REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

Project Design Phase-II
Solution Requirements (Functional & Non-functional)

Date	21 October 2022
Team ID	PNT2022TMID07039
Project Name	Project - - A Novel Method For Handwritten Digit Recognition System.
Maximum Marks	4 Marks

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Implementation	To import all the modules need for training our model. Import the libraries and load the MINST dataset.
FR-2	User Registration	Registration through Gmail
FR-3	User Confirmation	Confirmation via Email
FR-4	Pre processing	Model cannot take the image data directly so we need to perform some basic operations and process the data. The CNN model will require one more dimension so we reshape the matrix to shape (60000,28,28,1)
FR-5	Create and Train the model	Creating CNN model in Python data science project. A CNN model generally consists of convolutional and pooling layers. Keras will start the training of the model.
FR-6	Evaluation	We have 10,000 images in our dataset. The MNIST dataset is well balanced so we can get around 99% accuracy.

Non-functional Requirements:

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

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Handwritten character recognition is one of the practically important issues in pattern recognition applications. The applications of digit recognition include in postal mail sorting, bank check processing, form data entry, etc.
NFR-2	**Security**	Most PC efforts to establish safety include information encryption and passwords, OCR plays an important role for digital libraries, allowing the entry of image textual information into computers by digitization, image restoration, and recognition methods.
NFR-3	**Reliability**	The overall highest accuracy 90.37% is achieved in the recognition process by Multilayer Perceptron.

NFR-4	**Performance**	Most standard implementations of neural networks achieve an accuracy of ~(98–99) percent in correctly classifying the handwritten digits.
NFR-5	**Availability**	The established CNN model can determine and recognize handwritten digits with high accuracy, as it combines the weights of convolution layers during feature extraction with fully connected layers.
NFR-6	**Scalability**	High speed, robustness, flexible and suitable for text and document formats.

5. PROJECT DESIGN

5.1 DATA FLOW DIAGRAM

PROJECT DESIGN PHASE - II

Data Flow Diagram & User Stories

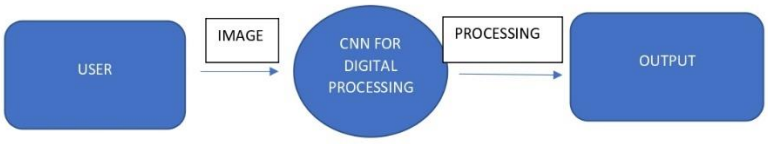
Date	22 October 2022
Team ID	PNT2022TMID07039
Project Name	A Novel Method for Handwritten Digit Recognition System
Maximum Marks	4 Marks

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.

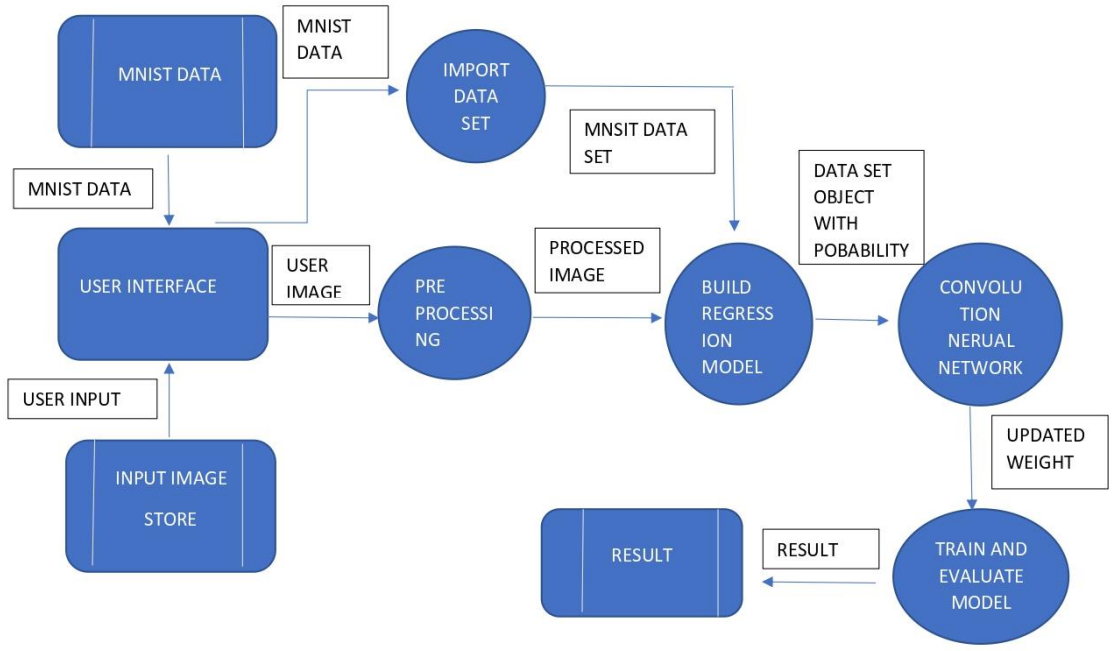
DFD Level-0

The DFD Level-0 consists of two external entities, the UI and the Output, along with a process, representing the CNN for Digit Recognition .Output is obtained after processing.



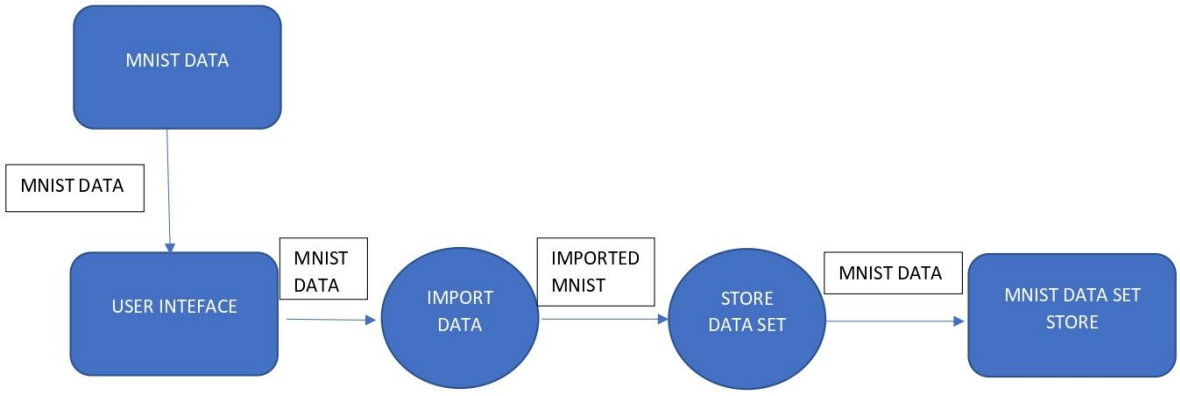
DFD Level-1

The DFD Level-1 consists of 2 external entities, the GUI and the Output, along with five process blocks and 2 data stores MNIST data and the Input image store, representing the internal workings of the CNN for Digit Recognition System. Process block imports MNIST data from library. Process block imports the image and process it and sends it to block where regression model is built. It sends objects with probabilities to CNN where weights are updated and multiple layers are built. Block trains and evaluates the model to generate output.



DFD Level-2

The DFD Level-2 for import data(figure 4) consists of two external data and one entity UI along with three process blocks, representing the three functionalities of the CNN for Digit Recognition System. It imports data from MNIST data store and stores on the system.



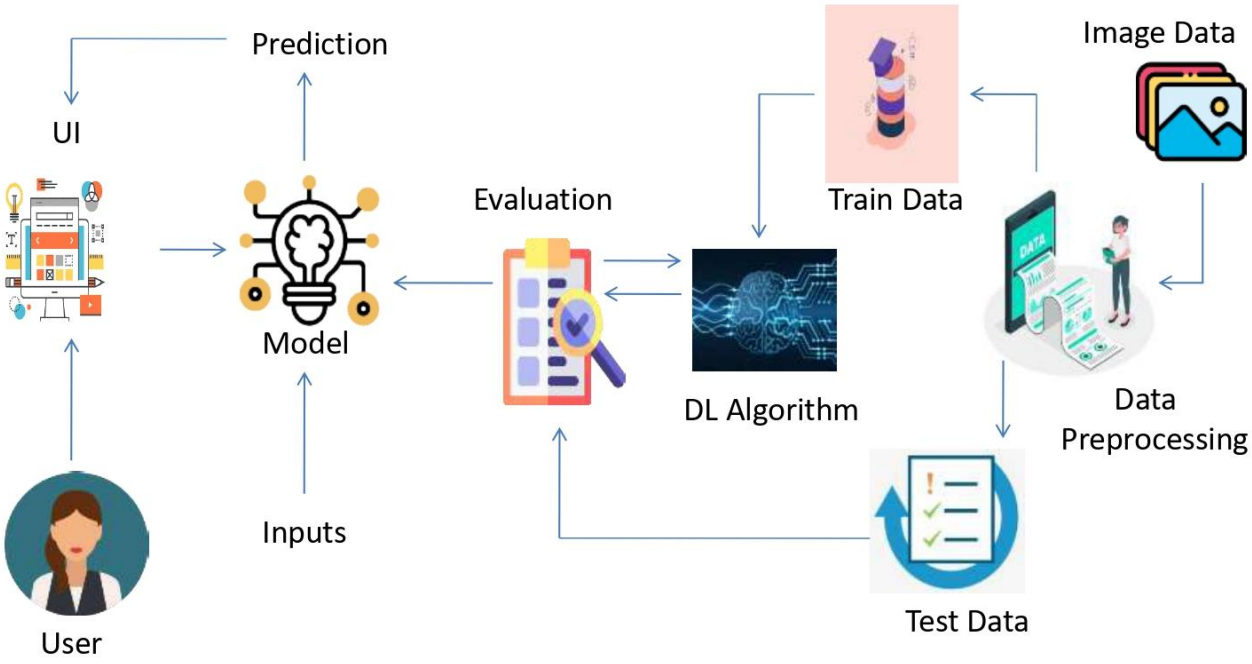
USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-2
		USN-3	As a user, I can register for the application through gmail or facebook	I can register & access the dashboard with Facebook Login	Medium	Sprint-2
	Login	USN-4	As a user, I can log into the application by entering email & password	I can login to the application	High	Sprint-1
	Dashboard	USN-5	Go to dashboard and refer the content about our project	I can read instructions also and the home page is user-friendly.	Low	Sprint-1
	Upload Image	USN-6	As a user, I can able to input the images of digital documents to the application	As a user, I can able to input the images of digital documents to the application	High	Sprint-3
	Predict	USN-7	As a user I can able to get the recognised digit as output from the images of digital documents or images	I can access the recognized digits from digital document or images	High	Sprint-3
		USN-8	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.	Medium	Sprint-4
Customer (Web user)	Login	USN-9	As a user, I can use the application by entering my email, password.	I can access my account	Medium	Sprint-4
Customer Care Executive	Dashboard	USN-10	upload the image	Recognize and get the output	High	Sprint-1
Administrator	Security	USN-11	updated the features	checking the security	Medium	Sprint-1

5.2 SOLUTION AND TECHNICAL ARCHITECTURE

SOLUTION ARCHITECTURE

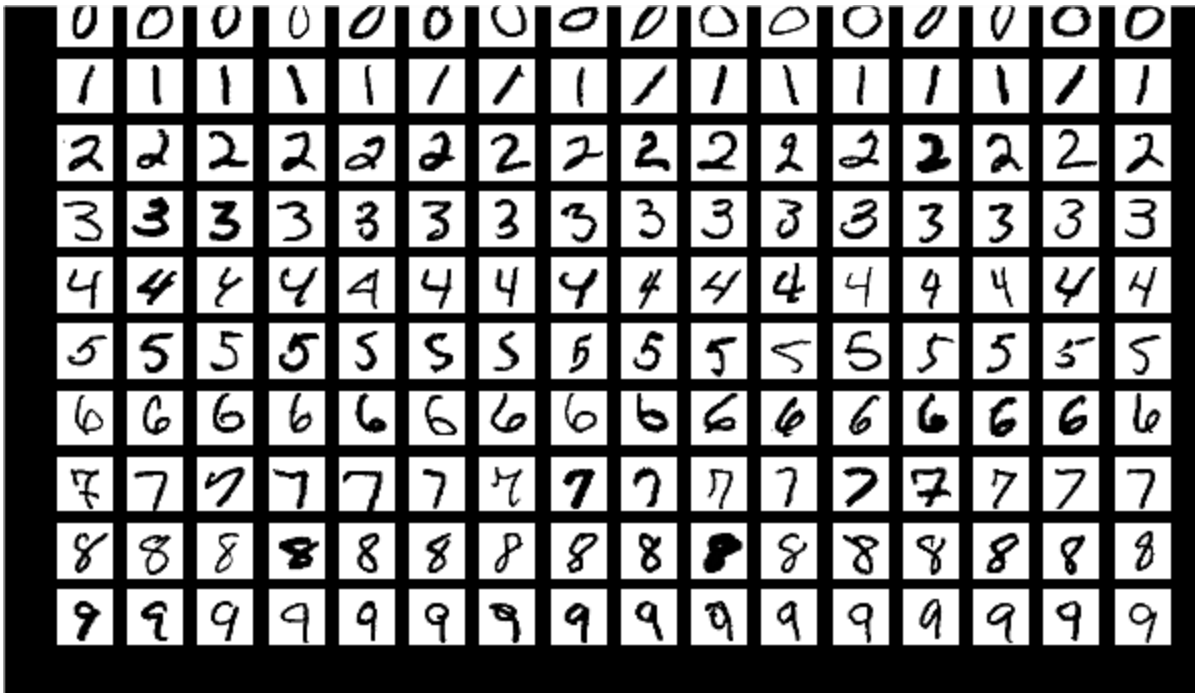
PROJECT DESCRIPTION: Given that everyone in the world has their own writing style, handwriting detection is one of the most intriguing research projects now underway. It is the computer's capacity to automatically recognise and understand handwritten figures or letters. Because of advances in science and technology, everything is being digitalized in order to reduce human effort. As a result, handwritten digit identification is required in many real-time applications. The MNIST data collection, which contains 70000 handwritten digits, is commonly employed in this recognition process. To train these photos and create a deep learning model, we use artificial neural networks. A web application is developed that allows the user to upload an image of a handwritten digit.



SOLUTION

MNIST Dataset Description:

The MNIST Handwritten Digit Recognition Dataset includes 60,000 training and 10,000 testing handwritten digit images. Each image has a height of 28 pixels and a width of 28 pixels, for a total of 784 (2828) pixels. Each pixel is connected with a single pixel value. It indicates how bright or dark that pixel is (larger numbers indicates darker pixel). This pixel value is an integer between 0 and 255.



PROCEDURE:

1. Install the latest TensorFlow library.
2. Prepare the dataset for the model.
3. Develop Single Layer Perceptron model for classifying the handwritten digits.
4. Plot the change in accuracy per epochs.
5. Evaluate the model on the testing data.
6. Analyse the model summary.
7. Add hidden layer to the model to make it Multi-Layer Perceptron.
8. Add Dropout to prevent overfitting and check its effect on accuracy.
9. Increasing the number of Hidden Layer neuron and check its effect on accuracy.
10. Use different optimizers and check its effect on accuracy.
11. Increase the hidden layers 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've taken it a step further, and our handwritten digit recognition technology not only recognises scanned images of handwritten numbers, but also allows you to write digits on the screen and have them recognised using an integrated GUI.

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.

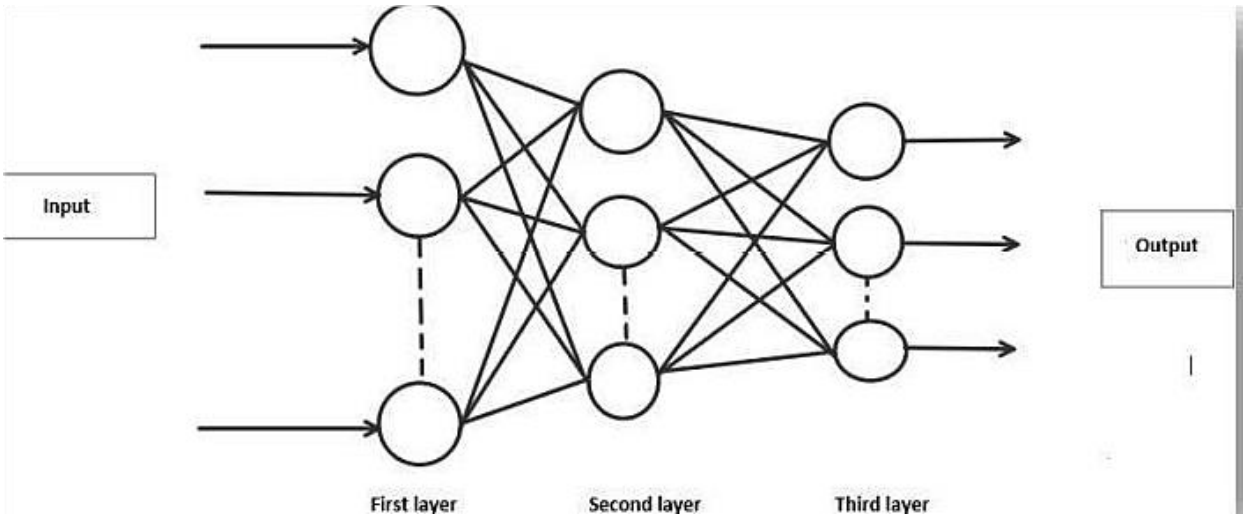
• **The hidden layer**: They are made of hidden units called activations providing nonlinear ties for the network. A number of hidden layers can vary according to our requirements.

• **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 based on how the brain functions. It is made up of several layers with numerous activations, which mirror neurons in our brain. A neural network attempts tolearn a set of parameters from a set of data, which may aid in recognising underlyinglinks. Because neural networks can adapt to changing input, they can produce the best possible results without having to rethink the output criteria.

METHODOLOGY:

We created a Neural Network with one hidden layer and 100 activation units (excluding bias units). Data is loaded from a.mat file, then features (X) and labels (Y) are extracted. Then, to avoid overflow during computation, features are divided by 255 and rescaled into a range of [0,1]. The data is divided into 60,000 training instances and 10,000 testing examples. Feedforward is used with the training set to calculate the hypothesis, followed by backpropagation to reduce the error between the layers. To solve the issue of overfitting, the regularisation parameter lambda is adjusted to 0.1. The optimizer is run 70 times to get the best fit model.



ALGORITHM:

Forward Propagation Architecture:

It is a brief description of how the CNN module will extract features and categorize the image based on them. The network's input layer, hidden layers, and output layer are depicted in the design. The feature extraction phase of the network involves multiple layers, including convolution and resampling.

Explanation of given system:

• The first layer of the architecture is the User layer. User layer will comprise of the people who interacts with the app and for the required results.

• The next three layers is the frontend architecture of the application. The application will be developed using which is the open-source platform for HTML, CSS and JavaScript. The application is deployed in the localhost which is shown on the browser. Through the app, the user will be able to upload pictures of the handwritten digits and convert it into the digitalized form.

• The one in between the database and view layer is the business layer which is the logical calculations on the basis of the request from the client side. It also has the service interface.

• The backend layer consists of two datasets: Training Data and Test Data. The MNIST database has been used for that which is already divided into training set of 60,000 examples and test of 10,000 examples.

• The training algorithm used is Convolution Neural Network. This will prepare the trained model which will be used to classify the digits present in the test data. Thus, we can classify the digits present in the images as: Class 0,1,2,3,4,5,6,7,8,9

WORKING:

• Neural Networks receive an input and transform it through a series of hidden layers.

• Each hidden layer is made up of a set of neurons, where each neuron is fully connected to all neurons in the previous layer.

• Neurons in a single layer function completely independently.

• The last fully connected layer is called the "output layer.

Convolution Layer:

The Convolutional layer is the core building block of a CNN. The layer's parameters consist of a set of learnable filters (or kernels), which have a small receptive field, but extend through the full depth of the input volume. During the forward pass, each filter is convolved across the width and height of the input volume, computing the dot product between the entries of the filter and the input and producing a 2- dimensional activation map of that filter. As a result, the network learns filters that activate when they see some specific type of feature at some spatial

position in the input.

Feature Extraction:

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

Subsampling Layer:

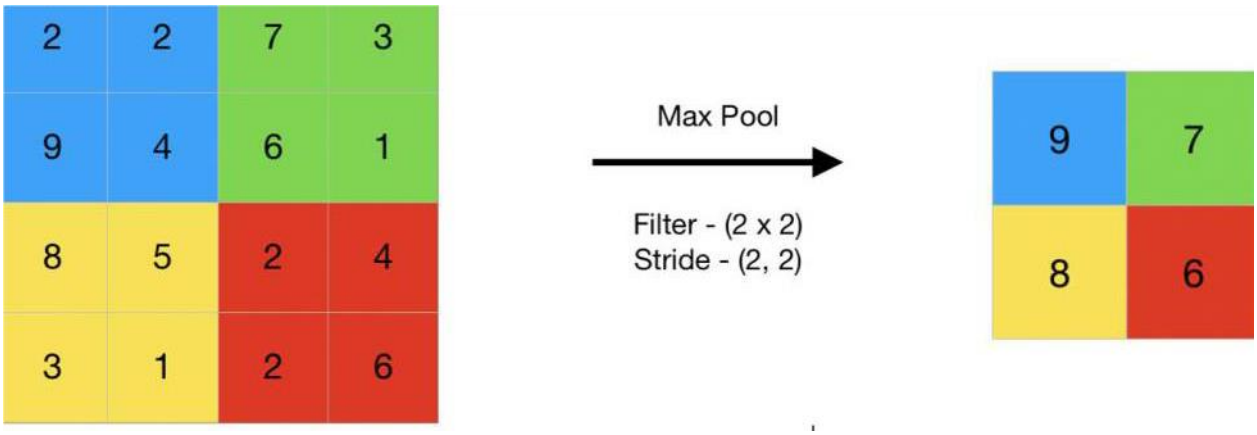
Subsampling, or down sampling, refers to reducing the overall size of a signal The subsampling layers reduce the spatial resolution of each feature map. Reduce the effect of noises and shift or distortion invariance is achieved.

Pooling layer:

It is common to periodically insert a Pooling layer in-between successive Conv layer in a Convent architecture. Its function is to progressively reduce the spatial size of the representation to reduce the number of parameters and computation in the network, and hence to also control overfitting. The Pooling Layer operates independently on every depth slice of the input and resizes it spatially, using the MAX operation.

TensorFlow:

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



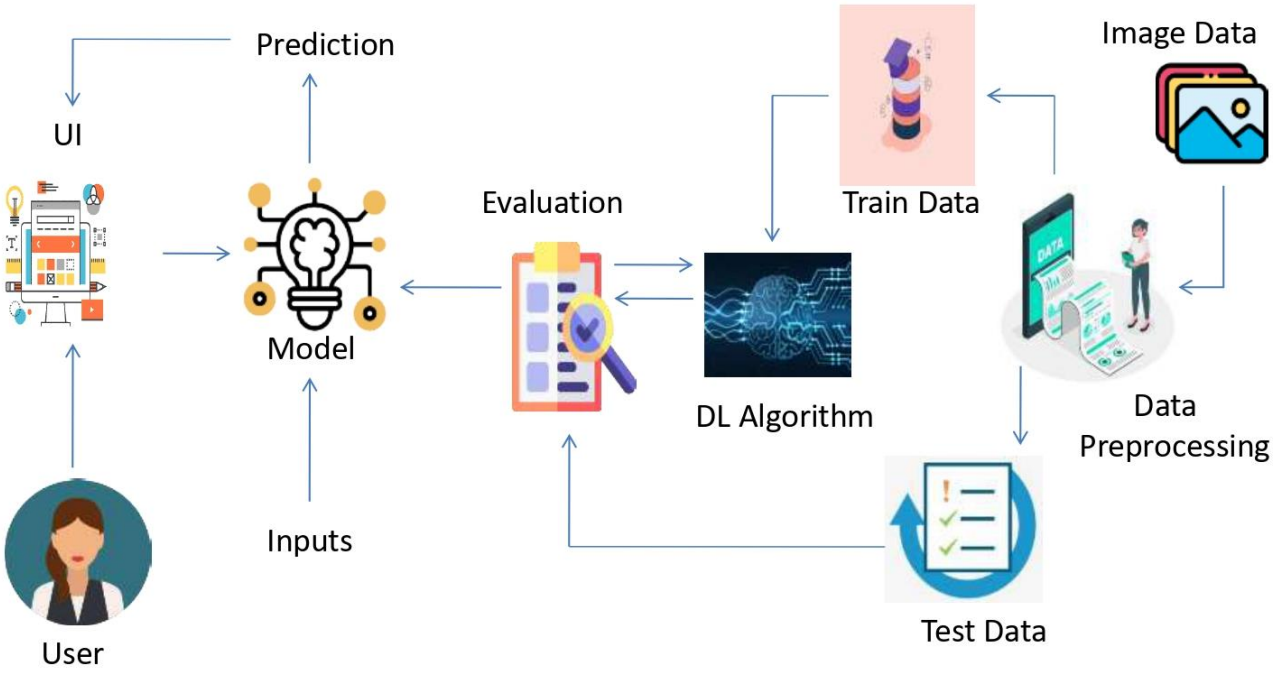
RESULT:

As with any study or project undertaken in the field of machine learning and image recognition, We do not consider our results to be perfect after processing. Machine learning is a field that is always evolving, and there is always room for advancement improvement in your process; there will always be something new strategy that produces superior outcomes for the same challenge The application has been submitted.

Three models were used: Multi-Layer Perceptron (MLP), Convolution Neural Network (CNN),

and Network (CNN). The accuracy of the classifier varies depending on the model which demonstrates which is superior.

TECHNICAL ARCHITECTURE



COMPONENTS AND TECHNOLOGIES

	Component	Description	Technology
1.	User Interface	user interfaces with application for the recognition of the handwritten digits.	HTML, CSS, JavaScript
2.	Application Logic-1	Logic for a process in the application	Python IBM Watson STT service
3.	Application Logic-2	Logic for a process in the application	IBM Watson Assistant
4.	Application Logic-3	Logic for a process in the application	MINIST dataset
5.	Database	digits dataset will be stored.	

6.	Cloud Database	Database Service on Cloud	IBM watson cloud.
7.	File Storage	File storage requirements	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	CNN, ANN ,RNN.
11.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration Cloud Server Configuration	cloud application

Application Characteristics

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Notification and redistribution	Linux,Python
2.	Security Implementations	Listing all the security / access controls implemented, use of firewalls etc.	encryption algorithm
3.	Scalable Architecture	To justify the scalability of architecture used in system. User friendly and highly flexible.	HTML,IBM CLOUD

4.	Availability	The web dashboard must be availableto users recognition accuracy.	IBM CLOUD HOSTING
5.	Performance	The handwritten digits are accurately classified with an accuracy of typical neural network .	IBM load balancing

6.PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNING AND ESTIMATION

Product Backlog, Sprint Schedule, and Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority
Sprint-1	Data Collection	USN-1	As a user, I can collect the dataset from various resources with different handwritings.	10	Low
Sprint-1	Data Preprocessing	USN-2	As a user, I can load the dataset, handling the missing data, scaling and split data into train and test.	10	Medium
Sprint-2	Model Building	USN-3	As a user, I will get an application with ML model which provides high accuracy of recognized handwritten digit.	5	High
Sprint-2	Add CNN layers	USN-4	Creating the model and adding the input, hidden, and output layers to it.	5	High
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority
Sprint-2	Compiling the model	USN-5	With both the training data defined and model defined, it's time to configure the learning process.	2	Medium

Sprint-2	Train & test the model	USN-6	As a user, let us train our model with our image dataset.	6	Medium
Sprint-2	Save the model	USN-7	As a user, the model is saved & integrated with an android application or web application in order to predict something.	2	Low
Sprint-3	Building UI Application	USN-8	As a user, I will upload the handwritten digit image to the application by clicking a upload button.	5	High
Sprint-3	USN-9	As a user, I can know the details of the fundamental usage of the application.	5		Low
Sprint-3	USN-10	As a user, I can see the predicted / recognized digits in the application.	5		Medium
Sprint-4	Train the model on IBM	USN-11	As a user, I train the model on IBM and integrate flask/Django with scoring end point.	10	High
Sprint-4	Cloud Deployment	USN-12	As a user, I can access the web application and make the use of the product from anywhere.	10	High

6.2 SPRINT DELIVERY SCHEDULE

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20

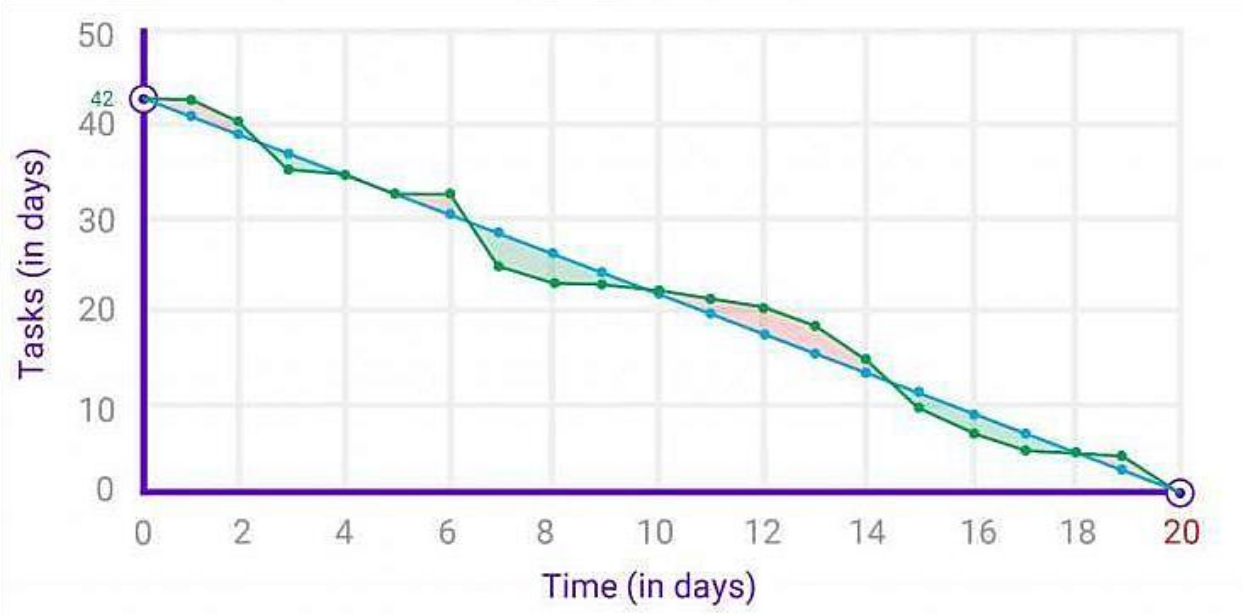
Velocity:

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

Average Velocity = 20 / 6 = 3.33

Burndown Chart:

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


50
42
40
30
20
10
0
Tasks (in days)
0
2
4
6
8
10
12
14
16
18
20
Time (in days)

7. CODING AND SOLUTIONING

7.1 FEATURE 1

Import the libraries and load the dataset**:**

First, we are going to import all the modules that we are going to need for training our model. The Keras
library already contains some datasets and MNIST is one of them. So we can easily import the dataset
and start working with it.

The **mnist.load_data()** method returns us the training data, its labels and also the testing data and its labels

```
import keras
from keras.datasets import mnist
from keras.models import Sequential
from keras.layers import Dense, Dropout, Flatten
from keras.layers import Conv2D, MaxPooling2D
from keras import backend as K
# the data, split between train and test sets
(x_train, y_train), (x_test, y_test) = mnist.load_data()
print(x_train.shape, y_train.shape)
```

```
(60000, 28, 28) (60000,)
```

Preprocess the data**:**

The image data cannot be fed directly into the model so we need to perform some operations and **process**
the data to make it ready for our neural network. The dimension of the training data is (60000,28,28).

The CNN model will require one more dimension so we reshape the matrix to shape (60000,28,28,1

```
x_train = x_train.reshape(x_train.shape[0], 28, 28, 1)
x_test = x_test.reshape(x_test.shape[0], 28, 28, 1)
input_shape = (28, 28, 1)
# convert class vectors to binary class matrices
y_train = keras.utils.to_categorical(y_train, 10
y_test = keras.utils.to_categorical(y_test, 10)
x_train = x_train.astype('float32')
x_test = x_test.astype('float32')
x_train /= 255
x_test /= 255
print('x_train shape:', x_train.shape)
print(x_train.shape[0], 'train samples')
print(x_test.shape[0], 'test samples')
```

```
x_train shape: (60000, 28, 28, 1)
60000 train samples
10000 test samples
```

Create the model**:**

Now we will create our CNN model in Python data science project. A CNN model generally consists
of convolutional and pooling layers.

It works better for data that are represented as grid structures, this is the reason why CNN works well
for image classification problems.

The dropout layer is used to deactivate some of the neurons and while training, it reduces offer fitting
of the model. We will then compile the model with the **Adadelta** optimizer

7.2 FEATURE 2

Prediction

Other Predictions

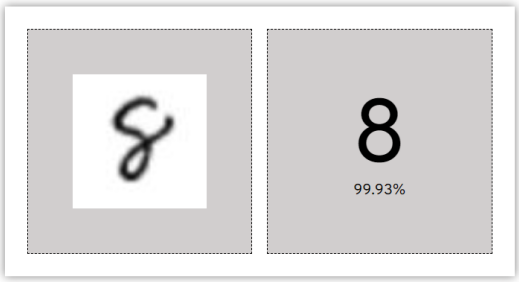
0	1	2	3	4	5	6	8	9
0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%

Prediction

Other Predictions

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

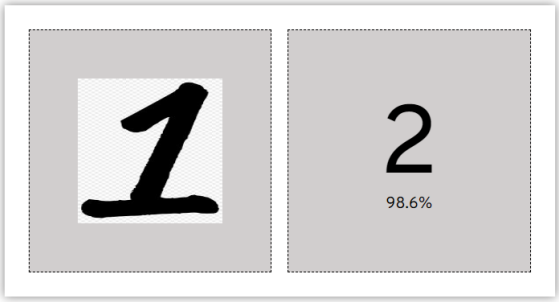
Prediction



Other Predictions

0	1	2	3	4	5	6	7	9
0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.06%

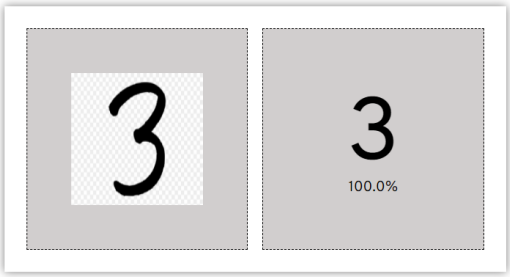
Prediction



Other Predictions

0	1	3	4	5	6	7	8	9
0.0%	0.11%	0.01%	0.02%	0.0%	0.0%	1.24%	0.0%	0.02%

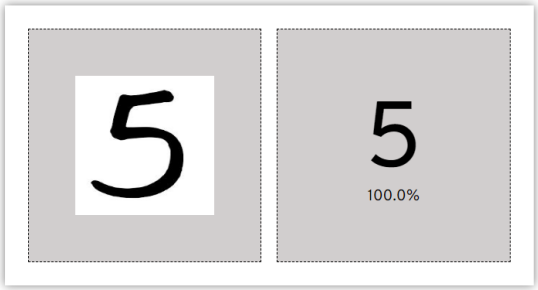
Prediction



Other Predictions

0	1	2	4	5	6	7	8	9
0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%

Prediction



Other Predictions

0	1	2	3	4	6	7	8	9
0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%

8.RESULTS

8.1 PERFORMANCE METRICS

Our model is built to work on real-world data, and real-world images are not even close to MNIST raster images, a lot of pre-processing was done to make a real image to look like a raster image.

Accuracy:

score Our model stopped training at the 2nd epoch as it reached 98.21% training accuracy and 98.51% validation accuracy with 5% training loss and 4% validation loss. The progression of accuracy and loss are represented

Prediction:

Our model is able to recognize computer-generated digits as well as handwritten digits. Computer-generated digit prediction is more accurate compared to real-world digit prediction, which can be observed in F

Model Evaluation & Prediction

For real-world image classification prediction, we need to do a little image pre-processing on the real-world images as model training was done with greyscale raster images. The steps of image pre-processing are,

1. Loading image
2. Convert the image to greyscale
3. Resize the image to 28x28
4. Converting the image into a matrix form
5. Reshape the matrix into 28x28x1

After pre-processing, we predict the label of the image by passing the pre-processed image through the neural network. The output we get is a list of 10 activation values 0 to 9, respectively. The position having the highest value is the predicted label for the image [18].

RESULTS AND DISCUSSION

Our model is built to work on real-world data, and real-world images are not even close to MNIST raster images, a lot of pre-processing was done to make a real image to look like a raster image

9. ADVANTAGES AND DISADVANTAGES

Digit recognition system is the working of a machine to train itself or recognizing the digits fromdifferent 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. The handwritten digit recognition is the ability of computers to recognize human handwritten digits.
It is a hard task for the machine because handwritten digits are not perfect and can be made withmany different flavors. The handwritten digit recognition is the solution to this problem which usesthe image of a digit and recognizes the digit present in the image.

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. Thoughthe 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 ConvolutionalNeural Network, and applying it to the handwritten recognition system

10.CONCLUTION

Our 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 therecognition of handwritten digit strings.
In this project, different machine learning methods, which are SVM (Support Vector Machine), ANN(Artificial Neural Networks), and CNN (Convolutional Neural Networks) architectures are used toachieve 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 HandwrittenCharacter Recognition System which recognizes human handwritten characters and predicts the output

12. APPENDIX

Python:

Python is an interpreted, high-level, general purpose programming language created by Guido Van Rossum and first released in 1991, Python's design philosophy emphasizes code Readability with its notable use of significant White space. Its language constructs and object oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is dynamically type and garbage collected. It supports multiple programming paradigms, including procedural, object oriented ,and functional programming.

Keras :

Keras is a powerful and easy-to-use free open source Python library for developing and evaluating **deep learning** model **.**It wraps the efficient numerical computation libraries **Theano** and **TensorFlow** and allows you to define and train neural network models in just a few lines of

code. It uses libraries such as Python, C#,C++ or standalone machine learning toolkits. Theano and TensorFlow are very powerful libraries but difficult to understand neural network.Keras is based on minimal structure that provides a clean and easy way to create deep learning models based on TensorFlow or Theano. Keras is designed to quickly define deep learning models. Well, Keras
is an optimal choice for deep learning applications.

Steps for creating a keras model:

1)First we must define a network model.
2)Compile it, which transforms the simple sequence of layers into a complex group of matrix operations.
3)Train or fit the network.
To import: from keras.models import Sequential
From keras.layers import Dense, Activation, Dropout

TensorFlow:

TensorFlow is a Python library for fast numerical computing created and released by Google. It is a foundation library that can be used to create Deep Learning models directly or by using wrapper librarie sthat simplify the process built on top of **TensorFlow.** TensorFlow tutorial is designed for both beginner and professionals. Our tutorial provides all the basic and advanced concept of machine learning and deep learning concept such as deep neural network, image processing and sentiment analysis. TensorFlow is one of the famous deep learning frameworks, developed by **Google** Team. It is a free and
open source software library and designed in **Python** programming language, this tutorial is designedin such a way that we can easily implements deep learning project on TensorFlow in an easy andefficient way. Unlike other numerical libraries intended for use in Deep Learning like **Theano**,**TensorFlow** was designed for use both in research and development and in production systems. It canrun on single CPU systems, GPUs as well as mobile devices and largescale distributed systems ofhundreds of machines.

Numpy:

NumPy is a Python library used for working with arrays. It also has functions for working in domain oflinear algebra, Fourier transform, and matrices. Numpy which stands for Numerical Python, is a libraryconsisting of multidimensional array objects and a collection of routines for processing those arrays.Using NumPy, mathematical and logical operations on arrays can be performed. This tutorial explains
the basics of NumPy such as its architecture and environment. It also discusses the various array

functions, types of indexing, etc. It is an opensource project and you can use it freely.
NumPy stands for Numerical Python. NumPyaims to provide an array object that is up to 50x faster than traditional Python lists. The array object inNumPy is called **ndarray**, it provides a lot of supporting functions that make working with **ndarray** very easy. Arrays are very frequently used in data science, where speed and resources arevery important.

Pillow:

Pillow is a free and ope nsource library for the Python programming language that allows you to easily create &s manipulate digital images. Pillow is built on top of PIL (Python Image Library). PIL is one of the important modules for image processing in Python. However, the PIL module is not supported since 2011 and does n't support python 3.
Pillow module gives more functionalities, runs on all major operating system and support for python
3. It supports wide variety of images such as "jpeg", "png", "bmp", "gif", "ppm", "tiff". You can do almost anything on digital images using pillow module. Apart from basic image processing functionality, including point operations, filtering images using built-in convolution kernels, and color space conversions.

Tkinkter:

Tkinter is the standard **GUI library** for Python. Python when combined with Tkinter provides a fast and easy way to create **GUI applications**. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.We need to import all the modules that we are going to need for training our model. The Keras library already contains some datasets and MNIST is one of them. So we can easily import the dataset through Keras. The mnist.load_data() method returns the training data, its labels along with the testing data and its labels.

Jupyter Notebook:

Jupyter Lab is a web-based interactive development environment for Jupyter notebooks, code, and data. JupyterLab is flexible: configure and arrange the user interface to support a wide range of workflows in data science, scientific computing, and machine learning.JupyterLab is extensible and modular: writeplugins that add new components and integrate with existing ones.

Machine Learning:

Machine learning is a method of data analysis that automates analytical model building. It is a branch of artificial intelligence based on the idea that systems can learn from data, identify patterns and make decisions with minimal human intervention.

Deep Learning:

Deep learning is an artificial intelligence (AI) function that imitates the workings of the human brain in processing data and creating patterns for use in decision making.learning is a subset of machine learning in artificial intelligence that has networks capable of learning unsupervised from data that is unstructured or unlabeled. Also known as deep neural learning or deep neural network.

Neural Networks:

A neural network is a series of algorithms that endeavors to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates. In this sense, neural networks refer to systems of neurons, either organic or artificial in nature.

Github Link

https://github.com/IBM-EPBL/IBM-Project-25807-1659973757