

A project report
on
A Novel Method for Handwritten Digit Recognition System

Powered by IBM India

Submitted by
Vinoth T(Team Lead)

Soorya J

Vishal Rosario S

Siva Sairam S

Project Id:PNT2022TMID07123

Department of Computer Science Engineering
Jerusalem College of Engineering
Velachery main road, Narayanapuram, Chennai - 600100

Industrial Mentor: Pradeepthi

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 convolutional neural network on basis of performance, accuracy, sensitivity, positive productivity, and specificity with using different parameters with the classifiers. To make machines more intelligent, the developers are diving into machine learning and deep learning techniques. A human learns to perform a task by practicing and repeating it again and again so that it memorizes how to perform the tasks. Then the neurons in his brain automatically trigger and they can quickly perform the task they have learned. Deep learning is also very similar to this. It uses different types of neural network architectures for different types of problems. 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 strings. 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 attempting 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 MNIST data set of images of handwritten 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 Handwritten 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 and documents containing colored and complex background and varied intensities. In this, different preprocessing techniques like skew detection and correction, image enhancement techniques of contrast stretching, binarization, noise removal techniques, normalization and segmentation, morphological processing techniques are discussed.
2. Sandhya Arora , used four feature extraction techniques namely, intersection,

shadow feature, chain code histogram and straight line fitting features. Shadow features are computed globally for character image while intersection features, chain code histogram features and line fitting features are computed by dividing the character image into different segments. On experimentation with a dataset of 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 Convolutional Neural Network. Though the goal is to create a model which can recognize the digits, it can be extended to letters and an individual's handwriting. The major goal of the proposed system is understanding Convolutional Neural Network, and applying it to the handwritten recognition system.





3.IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

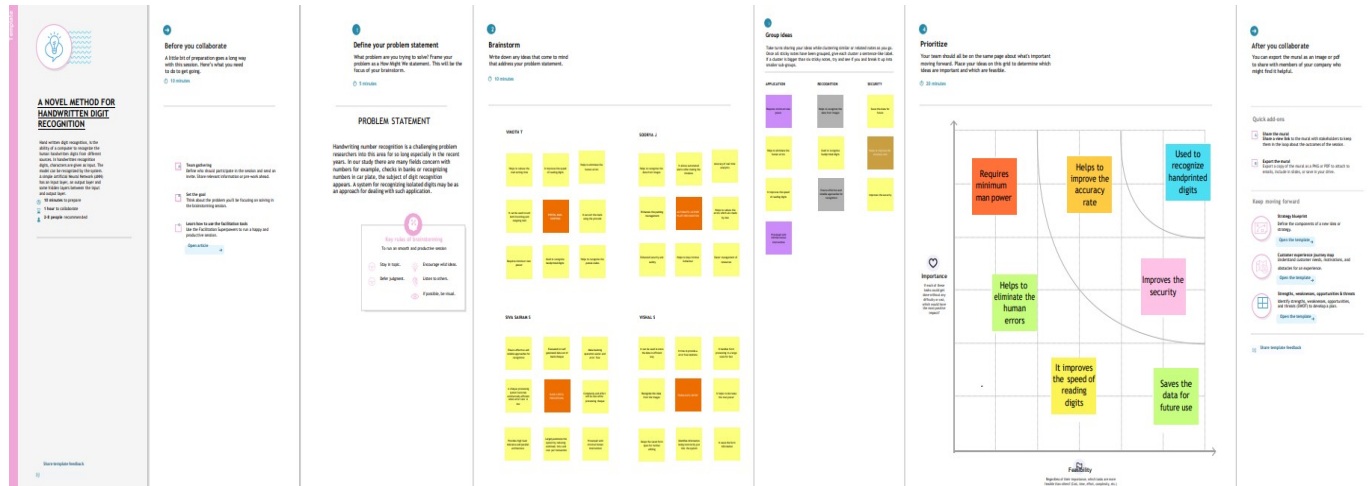


Reflect on the topic

Working silently and individually, have each person create a few sticky notes in all four quadrants below for about five minutes. With the remaining time, discuss notes in each quadrant.

What went well? What should we keep doing? What should we celebrate? Where did we make progress?	TOPIC Handwritten digit recognition	What went poorly? Where did we have problems? What was frustrating to us or others? What held us back?
<div>We should always improve the dataset and use efficient algorithms</div> <div>We can make progress by making the dataset more stronger</div> <div>The accuracy is increased by using different algorithms</div> <div>We should celebrate the accuracy that have increased</div> <div>CNN is always best for image processing</div> <div></div>	<div></div>	<div>The main problem lies in identification handwritten of digits from complex sources</div> <div>There is no possibility of obtaining information about the type of the input. The separation of text into characters is a massive task.</div> <div>The primary difficulty is to find the most efficient machine learning algorithm that is best suited for the real time usage.</div> <div>There is no constrained sources of data to study further.</div>
<div>Multilingual Handwritten Digits Recognition can be our future work</div> <div>We can develop a user-friendly interface to upload images from any sites/ urls</div> <div>system for object recognition, image segmentation, handwriting recognition can be designed</div> <div></div>	<div></div>	<div>We should change the dataset very specific for digits</div> <div>System should be designed to identify more complex handwritten text</div> <div>We should use different algorithms and deep learning techniques</div>
What ideas do you have? What ideas do you have for future work together? Where do you see opportunities to improve? What has untapped potential?		How should we take action? What do you believe we should do next? What specific things should we change? What should extend beyond this meeting?

3.2 IDEATION AND BRAINSTORMING



3.3 PROPOSED SOLUTION

S. No.

1.

Parameter

Problem Statement (Problem to be solved)

Descrption

Predic ng the performance level of cars is an important and interes ng problem. The main goal is to predict the performance of the car to improve certain behaviours of the vehicle. This can significantly help to improve the system's fuel consump on and increase efficiency. The performance analysis of the car is based on the engine type, no of engine cylinders, fuel type, horsepower, etc. These are the factors on which the health of the car can be predicted. It is an on-going

process of obtaining, researching, analysing, and recording health based on the above three factors.

The performance objectives like mileage, dependability, flexibility and cost can be grouped together to play a vital role in the prediction engine and engine management system. This approach is a very important step towards understanding the vehicle's performance.

1. Idea / Solution description To train the system with the dataset using a regression model and it will be integrated to the web-based application where the user is notified with the status.
2. Novelty / Uniqueness Giving the public and the manufacturer the feature to analyse their vehicle's performance.
3. Social Impact / Customer benefit The petrol/diesel cost can be reduced because of better mileage performance and the existing vehicle parts can be reused which increases the reusability thus decreases the cost on new products and the physically disabled people have better seat comfort because of accessories work. Better mileage and better engine maintenance

provides complete combustion thus emitting less harmful gases.

4. Business Model (Revenue The web-based application Model) has a friendly UI for the

customer to enter their vehicles detail and the system predicts the value within few seconds.

5. Scalability of the Solution The project will be scalable

when the parts used to measure data in vehicles is feasible and the ML model is fast in processing data.

3.4 PROBLEM SOLUTION FIT

PROJECT TITLE: A Novel Method for Handwritten Digit Recognition System		TEAM ID: PNT2022TMID07123		
Define CS, fit into CC	<div>1. CUSTOMER SEGMENT(S)CS</div> <div>The Bank Employee who makes the transactions through the cheque.</div>	<div>6. CUSTOMER CONSTRAINTSCC</div> <div>External dependencies are quite expensive and it is not offered by the people, So this process overcome the problem through their installation in mobile.</div>	<div>5. AVAILABLE SOLUTIONSAS</div> <div><div>--- Automatic digit recognition</div><div>--- In past, people identify the digits to their analysis sometimes it causes wrong transactions.</div><div>--- By using this application, they could easily identify the digits</div></div>	Explore AS, differentiate
	<div>2. JOBS-TO-BE-DONE / PROBLEMSJ&P</div> <div>Every single has their own style of writing which could not recognize by the computer.</div>	<div>9. PROBLEM ROOT CAUSERC</div> <div>Every single has their own style of writing which could not recognize by the computer.</div>	<div>7. BEHAVIOURBE</div> <div>To classify the digits in correct way, they could make the transactions easier without any doubtfulness.</div>	Focus on J&P, tap into BE, understand RC
Identify strong TR & EM	<div>3. TRIGGERSTR</div> <div>Feel free to make transactions without any fear about their style of writing</div>	<div>10. YOUR SOLUTIONSL</div> <div><div>--CNN model could be used to provide very High accuracy in image recognition problems and also reduces the high dimensionality of the images, without losing its information.</div><div>--It can be used to convert the handwritten digits to machine readable format.</div></div>	<div>8. CHANNELS OF BEHAVIOURCH</div> <div><div>ONLINE:</div><div>Promoting this application through the mobiles, the transaction could be done at any place without the presence in bank.</div><div>OFFLINE:</div><div>The identification of the digits which is in the handwritten form directly captured by using mobile application and that could be used to convert the those digits into machine readable forms.</div></div>	Extract online & offline CH of BE
	<div>4. EMOTIONS: BEFORE / AFTEREM</div> <div>If the person faces a problem regarding the transactions they could confidently handle the situation by using handwritten digit recognition system</div>			

4. REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

FR No.

Sub Requirement (Story / Sub-Task)

FR-1	<p>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).</p> <p>In the realm of deep learning, this has been the subject of countless studies.</p>
FR-2	<p>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 allocated to a website on a server. Shared, dedicated, VPS, and reseller hosting are the four basic varieties.</p>
FR-3	<p>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.</p>
FR-4	<p>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.</p>
FR-5	<p>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.</p>

Import the libraries: Libraries required are Keras, Tensor flow, Numpy, Pillow, Tkinter.

1. Keras: Keras is a powerful and easy-to-use free opensource Python library for developing and evaluating deep learning models. 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 for creating neural networks. 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.
2. 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 libraries that simplify the process built on top of TensorFlow. TensorFlow tutorial is designed for both beginners 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 opensource software library and designed in Python programming language, this tutorial is designed in such a way that we can easily implements deep learning project on TensorFlow in an easy and efficient 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 can run on single CPU systems, GPUs as well as mobile devices and largescale distributed systems of hundreds of machines.
3. Numpy: NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, Fourier transform, and

matrices. Numpy which stands for Numerical Python, is a library consisting 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.

NumPy aims to provide an array object that is up to 50x faster than traditional Python lists. The array object in NumPy 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 are very important.

4. Pillow: Pillow is a free and open source library for the Python programming language that allows you to easily create & 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 doesn'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 builtin convolution kernels, and color space conversions.
5. Tkinter: Tkinter is the standard and easy way to create Tk GUI toolkit. GUI library for Python. Python when combined with Tkinter provides a fast GUI applications We need to import all the. Tkinter provides a powerful objectoriented interface to the modules that we are going to need for training our model. library already contains some datasets and MNIST is one of them. The Keras 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.

Loading the Data Set:

6. MNIST Data Set: modified National Institute of Standards and Technology (MNIST) is a large set of computer vision dataset which is extensively used for training and testing different systems. It was created from the two special datasets of National Institute of Standards and Technology (NIST) which holds binary images of handwritten digits. The training set contains handwritten digits

from 250 people, among them 50% training dataset was employees from the Census Bureau and the rest of it was from high school students. However, it is often attributed as the first datasets among other datasets to prove the effectiveness of the neural networks. The database contains 60,000 images used for training as well as few of them can be used for crossvalidation purposes and 10,000 images used for testing. All the digits are grayscale and positioned in a fixed size where the intensity lies at the center of the image with 28×28 pixels. Since all the images are 28×28 pixels, it forms an array which can be flattened into 28*28=784 dimensional vector. Each component of the vector is a binary value which describes the intensity of the pixel.

4.2 NON-FUNCTIONAL REQUIREMENTS

FR No.	Non-Functional Requirement
--------	----------------------------

NFR-1	Usability
-------	-----------

NFR-2	Security
-------	----------

NFR-3	Reliability
-------	-------------

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. 1) The system generates a thorough description of the instance

on parameters, which might

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 quantity of 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

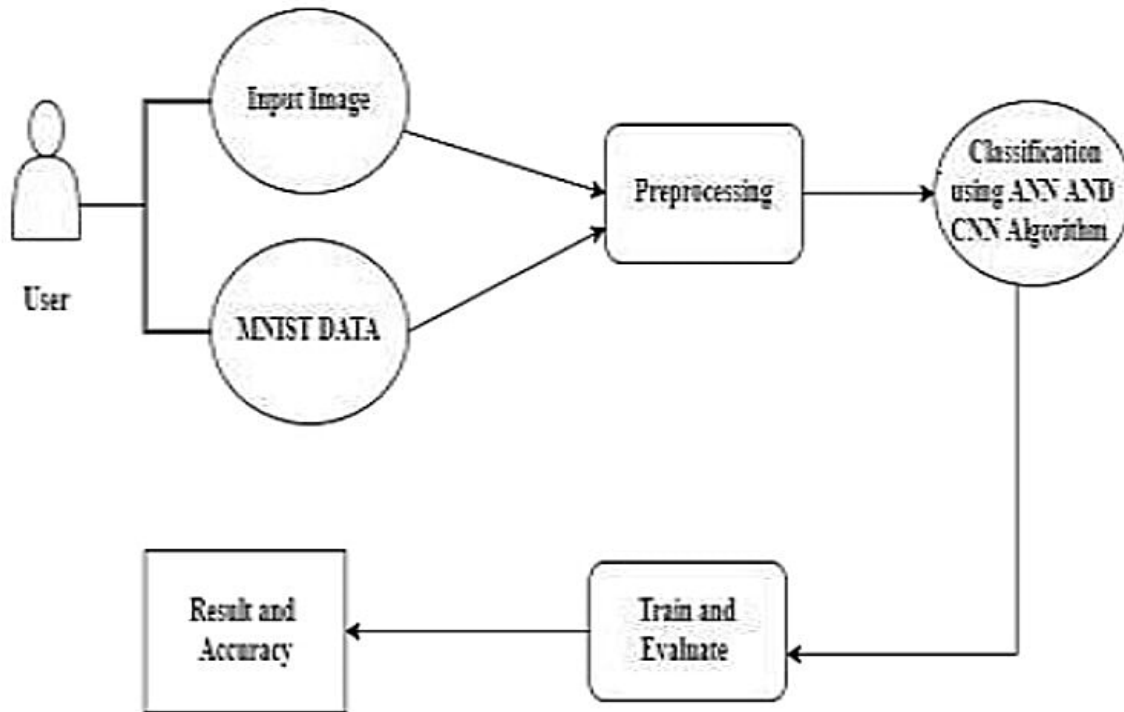
With typed text in high-quality photos, optical character recognition (OCR) technology offers accuracy rates of greater than 99%. However, variances in spacing, abnormalities in handwriting, and the variety of human writing styles result in less precise character identification.

NFR-5

Availability

5. PROJECT DESIGN

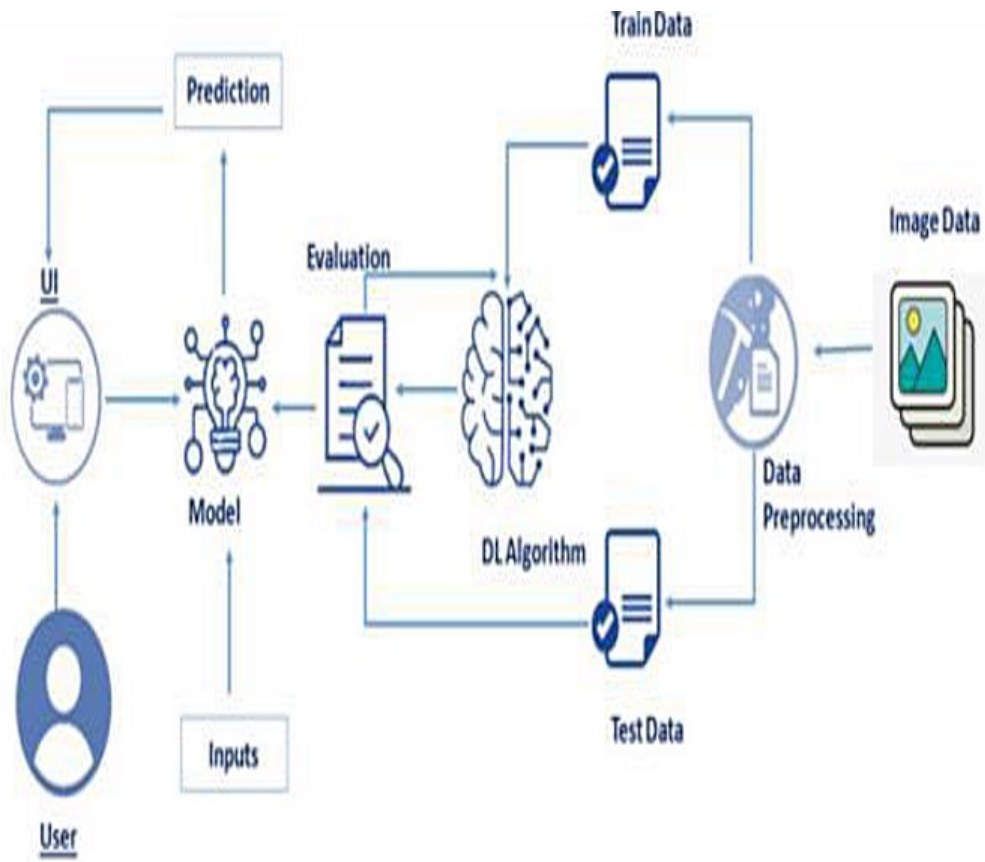
5.1 DATA FLOW DIAGRAM



5.2 SOLUTION AND TECHNICAL ARCHITECTURE

SOLUTION ARCHITECTURE

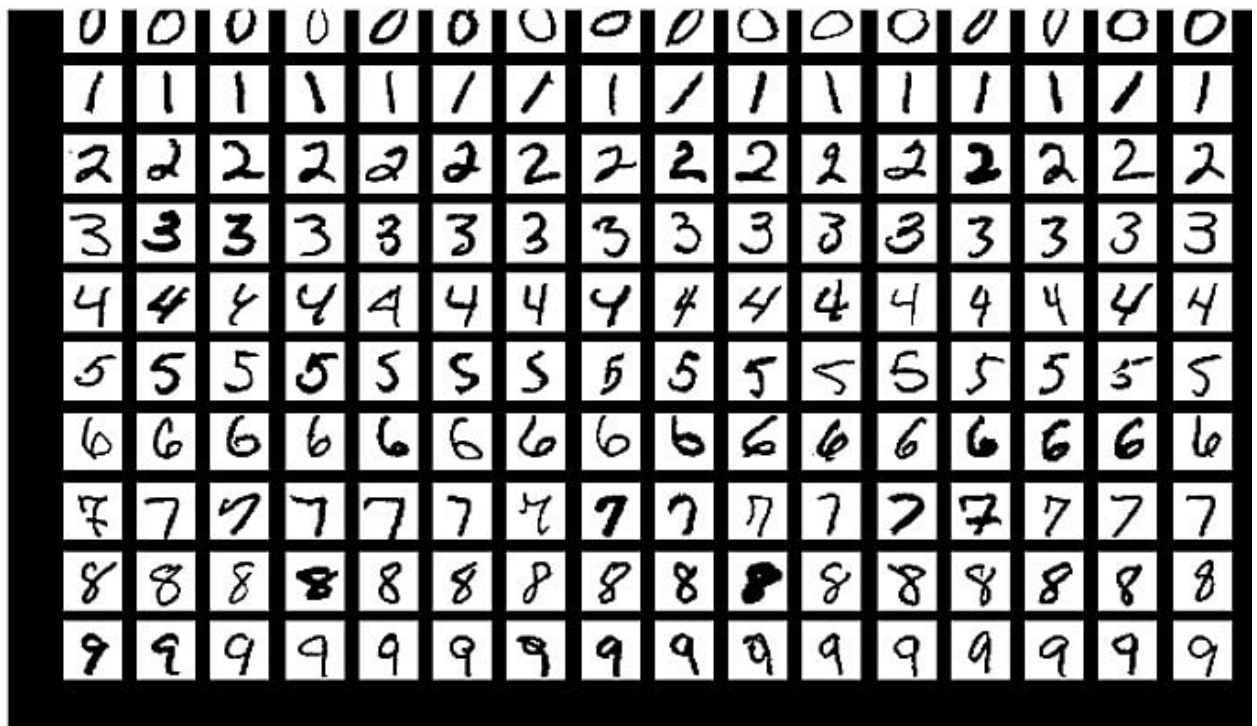
PROJECT DESCRIPTION: Given that everyone in the world has their own writing style, handwriting detection is one of the most intriguing research projects now underway. It is the computer's capacity to automatically recognise and understand handwritten figures or letters. Because of advances in science 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 Descrip on:

The MNIST Handwri en Digit Recogni on Dataset includes 60,000 training and 10,000 tes ng handwri en 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 almost 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.

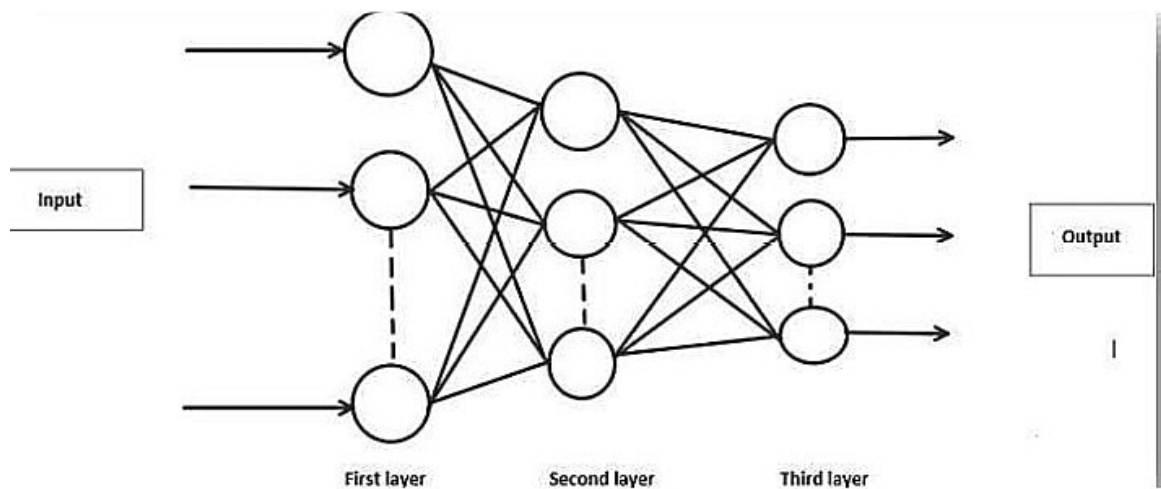
Approach:

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

1. **The input layer:** It distributes the features of our examples to the next layer for calculation of activations of the next layer.
2. **The hidden layer:** They are made of hidden units called activations providing nonlinearities for the network. A number of hidden layers can vary according to our requirements.
3. **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 to learn a set of parameters from a set of data, which may aid in recognising underlying links. 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 regularization parameter λ is adjusted to 0.1. The optimizer is run 70 times to get the best fit model.



ALGORITHM:

Forward Propaga on Architecture:

It is a brief descrip on 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 extrac on phase of the network involves mul ple layers, including convolu on and resampling.

Explana on of given system:

1. 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.
2. The next three layers is the frontend architecture of the applica on. The applica on will be developed using which is the open-source pla orm for HTML, CSS and JavaScript. The applica on is deployed in the localhost which is shown on the browser. Through the app, the user will be able to upload pictures of the handwri en digits and convert it into the digitalized form.
3. The one in between the database and view layer is the business layer which is the logical calcula ons on the basis of the request from the client side. It also has the service interface.
4. 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.
5. The training algorithm used is Convolu on 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:

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

Convolutional 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.

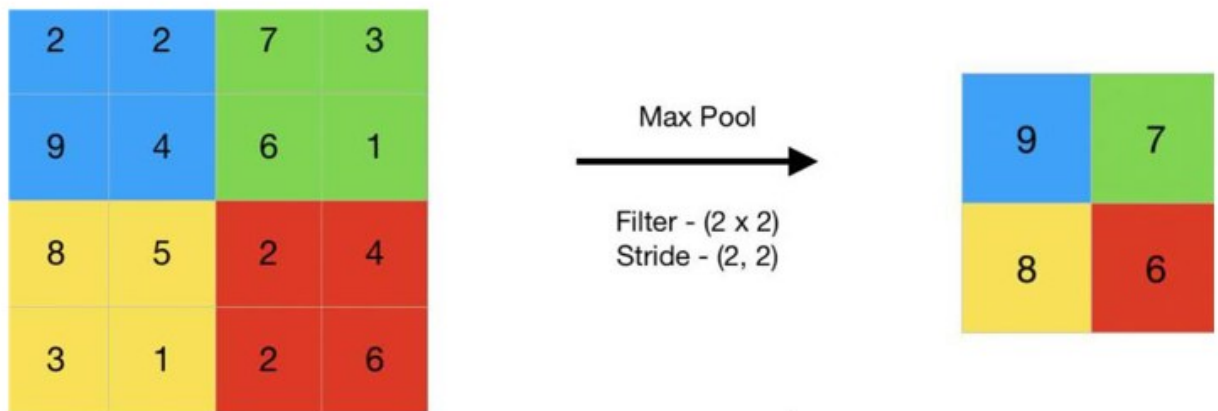
Pooling layer:

It is common to periodically insert a Pooling layer in-between successive Conv layers in a ConvNet 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:

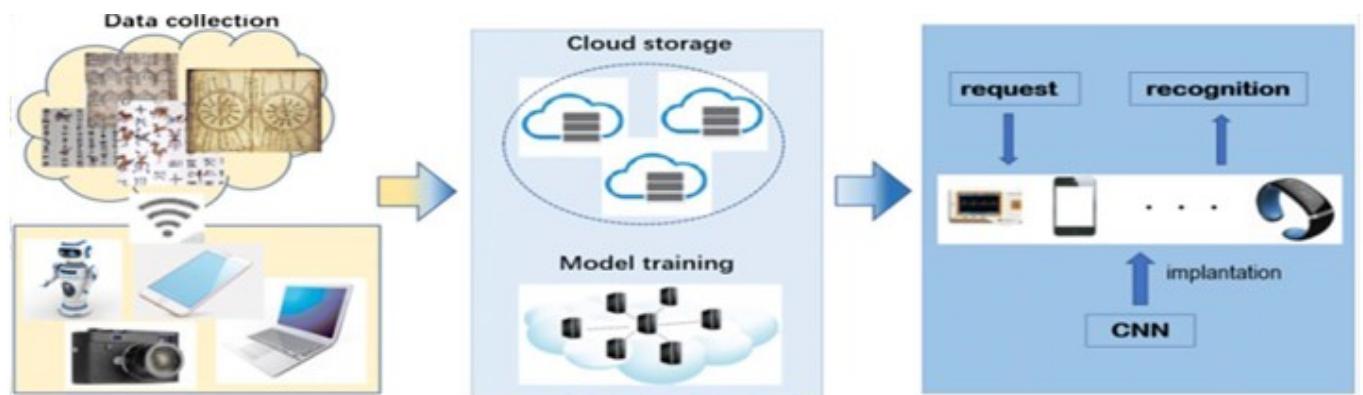
As with any study or project undertaken in the field of machine learning and image recognition, We do not consider our results to be perfect a er processing. Machine learning is a field that is always evolving, and there is always room for advancement improvement in your process; there will always be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

The applica on has been submi ed.

Three models were used: Mul -Layer Perceptron (MLP), Convolu on 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

Characteristics	Description	Technology
User Interface	How user interacts	HTML, CSS, JavaScript with application e.g. Web UI, Mobile App, Chatbot etc.
Application Logic-1	Logic for a process	Python in the application
Application Logic-2	Logic for a process	IBM Watson STT service in the application
Application Logic-3	Logic for a process	IBM Watson Assistant in the application
Database	Data Type	MySQL, NoSQL, Configurations ,etc
Cloud Database	Database Service	IBM DB2, IBM Cloudant on Cloud
File Storage	File Storage	IBM Block Storage requirements
External API-1	Purpose of	IBM Weather API in the application
External API-2	Purpose of	Aadhar API External API used in the application
Machine Learning Model	Purpose of	Object Recognition Model Machine Learning Model
Infrastructure (Server / Cloud)	Application	Local, Cloud Foundry Deployment on Local System / Cloud Local Server Configuration Cloud Server Configuration

Application Characteristics

1. Open-Source Frameworks The open-source frameworks used are listed. The technology of Opensource framework
2. Security Implementations Listing all the security / Controls, OWASP access controls implemented, use of firewalls etc. SHA-256, Encryptions, IAM

3. Scalable Architecture To justify the scalability 3 – tier, Micro-services of architecture used in system. User friendly and highly flexible.
4. Availability Figures and abstract. The Distributed servers, IBM cloud capabilities for recognizing handwritten digits have been implemented. These characteristics extract slope or slant information from the digit image based on shape analysis. They are successful in achieving high recognition accuracy.
5. Performance The handwritten digits are accurately classified with an accuracy of typical neural network . number of requests per sec, use of Cache

6.PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNING AND ESTIMATION

Product Backlog, Sprint Schedule, and Estimation

	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					
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

	Total Story Points		Sprint Start Date	Sprint End Completed Date (Planned)	Story Points
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

Sprint

Duration

Velocity:

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint).

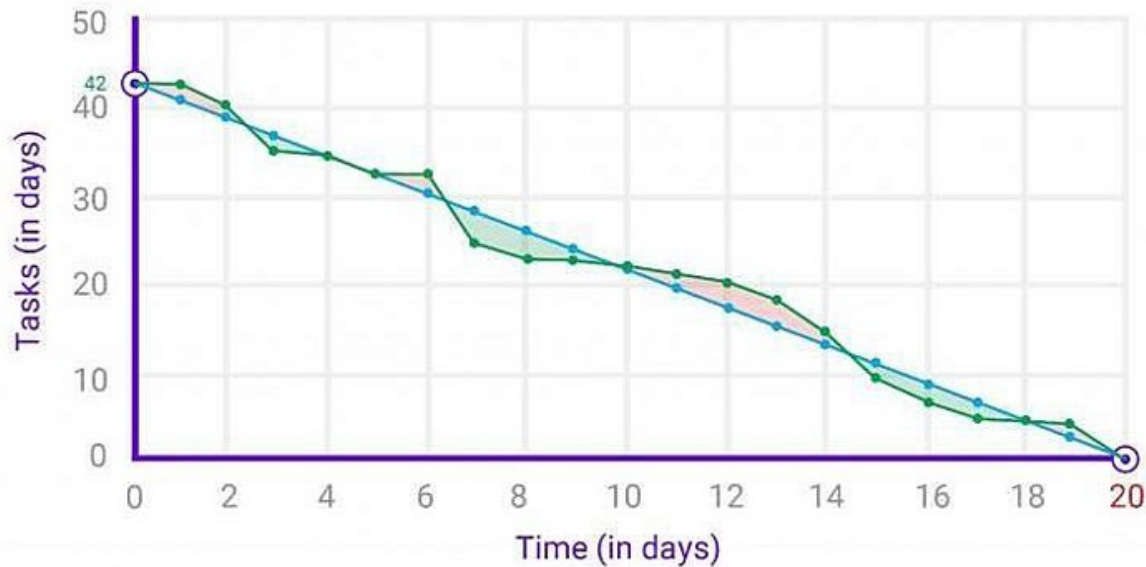
Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

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



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

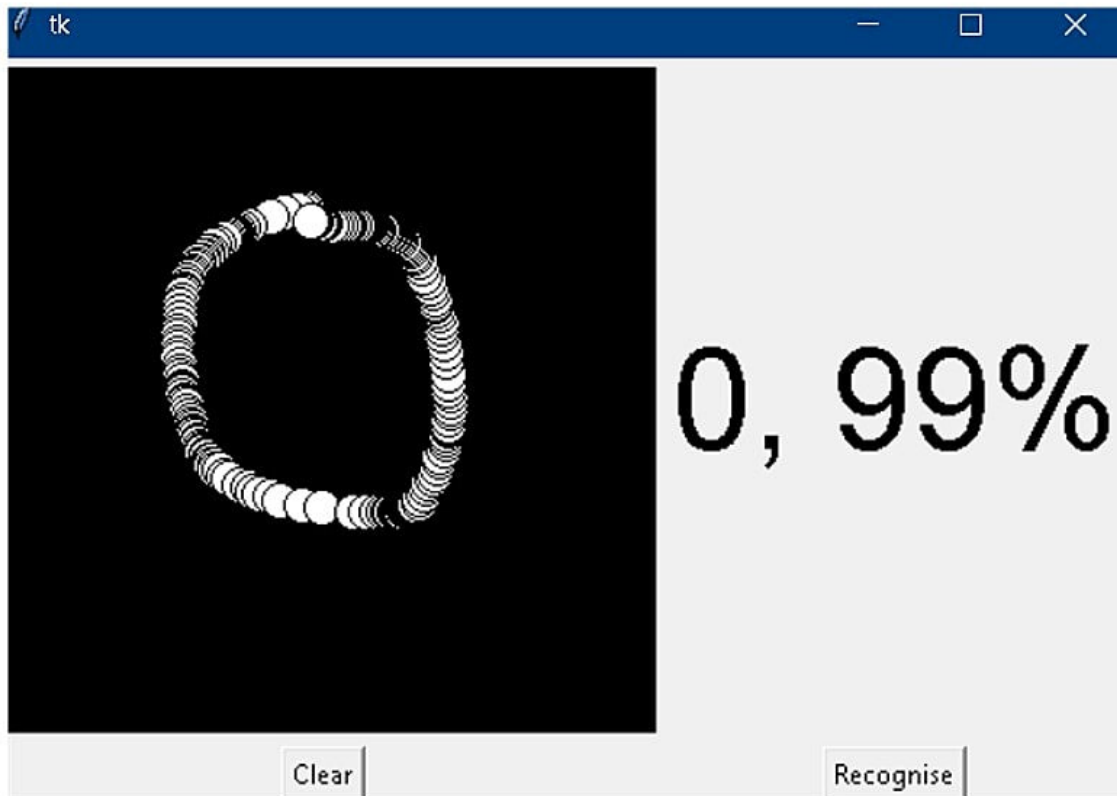
for image classification problems.

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

7.2 FEATURE 2







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:

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 from different sources like emails, bank cheque, papers, images, etc. and in different real-world scenarios for 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 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.

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

10. CONCLUSION

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 the recognition 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 to achieve high performance on the digit string recognition problem.

11. FUTURE SCOPE

The proposed system takes 28x28 pixel sized images as input. The same system with further modifications and improvements in the dataset and the model can be used to build

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 libraries that 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 designed in such a way that we can easily implement deep learning project on TensorFlow in an easy and efficient 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 can run on single CPU systems, GPUs as well as mobile devices and large scale distributed systems of hundreds of machines.

Numpy:

NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, Fourier transform, and matrices. Numpy which stands for Numerical Python, is a library consisting 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 open source project and you can use it freely. NumPy stands for Numerical Python. NumPy aims to provide an array object that is up to 50x faster than traditional Python lists. The array object in NumPy 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 are very important. **Pillow:**

Pillow is a free and open source library for the Python programming language that allows you to easily create & 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.

Tkinter:

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: write plugins 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-27943-1660101782>

PROJECT DEMO LINK

https://drive.google.com/file/d/1WJTetz_YxJWjo1O0MFhGMXk_isV7scrN/view?usp=sharing