A
Project Report
On

"Handwritten Digits Recognition"

Prepared By

Prishita Modi [19DCS064]
Jay Pandya [19DCS074]

Under the guidance of
Prof. Chandrashekhar Pawar

A report submitted to

Charotar University of Science and Technology

For the Partial Fulfillment of the Requirements for
5th Semester Software Group Project-III (CS348)

Submitted at

CSE DEPSTAR
At: Changa, Dist: Anand-388421
October 2021

CERTIFICATE

This is to certify that the report entitled "**Handwritten Digits Recognition**" is a bonafied work carried out by **Prishita Modi[19DCS064] and Jay Pandya [19DCS074]** under the guidance and supervision of **Prof. Chandrashekhar Pawar** for the subject **CS348 Software Group Project-III** (CSE) of 5th Semester of Bachelor of Technology in **DEPSTAR** at Faculty of Technology & Engineering – CHARUSAT, Gujarat.

To the best of our knowledge and belief, this work embodies the work of candidates themselves, has duly been completed, and fulfills the requirement of the ordinance relating to the B.Tech. Degree of the University and is up to the standard in respect of content, presentation and language for being referred to the examiner.

Prof. Chandrashekhar Pawar
Assistant Professor
CSE,
DEPSTAR,
Changa, Gujarat

Dr. Amit Ganatra
Principal,
DEPSTAR Dean,
FTE
CHARUSAT, Changa, Gujarat.

Devang Patel Institute of Advance Technology And Research At: Changa,

Ta. Petlad, Dist. Anand, PIN: 388 421. Gujarat

ACKNOWLEDGEMENT

We, the developer of a Machine Learning project namely “Handwritten Digit Recognition”, with immense pleasure and commitment would like to present the project assignment. The development of this project has given us wide opportunity to think, implement and interact with various aspects of management skills as well as the new emerging technologies. Every work that one completes successfully stands on the constant encouragement, good will and support of the people around. We hereby avail this opportunity to express our gratitude to number of people who extended their valuable time, full support, and cooperation in developing the project.

We express deep sense of gratitude towards our Head of the CSE Department, Mr. Parth Goel and project guides Prof. Chandrashekhar Pawar for the support during the whole session of study and development. It is because of them, that we were prompted to do hard work, adopting new technologies.

We are sincerely thankful to all the people who helped us complete the project in one way or the other. They altogether provided us favorable environment, and without them it would not have been possible to achieve the goal.

Sincere Thanks,
Prishita Modi
Jay Pandya

ABSTRACT

For humans, identifying numbers or items in a picture is extremely simple, but how do you train the machine to recognize these different things in images? Artificial Neural Networks (ANN) can solve this problem. In this report, a neural network has been trained by identifying pictures in MNIST handwritten digital database to predict exactly what the numbers in the picture are. Obviously, human beings can perceive that there is a hierarchy or conceptual structure in the image, but the machine does not. No matter what the environment of the image (image background) is, it is unchallenging for human beings to judge whether there is such a figure in the image and it is unnecessary to repeat the learning training.

TABLE OF CONTENT

Chapter 1:- Introduction

1.1) Introduction:

In the current age of digitization, handwriting recognition plays an important role in information processing. A lot of information is available on paper, and processing of digital files is cheaper than processing traditional paper files.

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 aim of a handwriting recognition system is to convert handwritten characters into machine readable formats. Here, we used a neural network, which is having wide applications in image classification, object recognition, recommendation systems, signal processing, natural language processing, computer vision, and face recognition. The ability to automatically detect the important features of an object without any human supervision.

The main applications are vehicle license-plate recognition, postal letter-sorting services, Cheque truncation system (CTS) scanning and historical document preservation in archaeology departments, old documents automation in libraries and banks, etc. All these areas deal with large databases and hence demand high recognition accuracy, lesser computational complexity and consistent performance of the recognition system.

1.2) Background and Motivation

Handwriting recognition of characters has been around since the 1980s.The task of handwritten digit recognition, 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 so on. There are different challenges faced while attempting to solve this problem. The handwritten digits are not always of the same size, thickness, or orientation and position relative to the margins.

Our goal was to build the system in such a way that it can recognize the handwritten digits provided in the MINIST data set of images of hand written digits (0-9). The data set used for our application is the MNIST data set (originally composed of 60,000 training images and 10,000 testing images). Each image is a 28 x 28 grayscale (0-255) labelled representation of an individual digit.

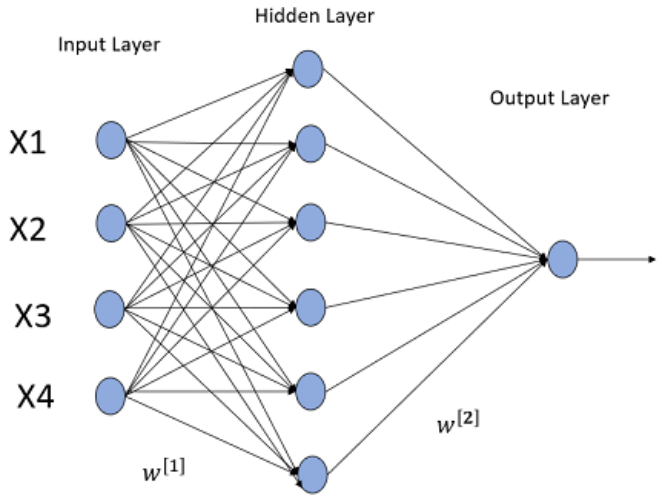
The general problem we predicted we would face in this digit classification problem was the similarity between the digits like 1 and 7, 5 and 6, 3 and 8, 9 and 8 etc. Also, people write the same digit in many different ways - the digit '1' is written as '1', '**1**', '*1*' or '1'. Similarly, 7 may be written as 7, 7, or 7. Finally the uniqueness and variety in the handwriting of different individuals also influences the formation and appearance of the digits.

Chapter 2:-About Model

2.1 Neural Networks

A neural network is a model inspired by how the brain works. It consists of multiple layers having many activations, this activation resembles neurons of our brain. A neural network tries to learn a set of parameters in a set of data which could help to recognize the underlying relationships. Neural networks can adapt to changing input so, the network generates the best possible result without needing to redesign the output criteria.
We approached 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.



The input layer is where the features of the input features (X) is fed, the Hidden layer will be the weighted sum of the inputs with parameters (W) followed by an activation function. There will always be bias (b) for each hidden layer. The output of the neural network will be the weighted sum of outputs of previous hidden layer followed by an activation function.

$Z_1 = W_1 . X + b_1$, where Z_1 is the weighted sum of inputs and b_1 is the bias.

X is the input matrix where each training example is stacked horizontally via columns. So, dimensions of X are (N_x, m) where N_x is the number of features, m is the number of training examples. If the number of units in the hidden layer is N_1, then dimensions of W_1 will be (N_1, N_x). So, dimensions of Z_1 will be (N_1, m). b_1 is a scalar quantity and when we add it in python, it will be added by broadcasting rules. The output of the hidden layer is the result of passing Z_1 to the activation function. Similarly for an N layer neural network, at any hidden layer, dimensions of W_n will be $(N_n, Nn -1)$.

So, g(Z) can be any activation function such as RELU or sigmoid. In practice, we always choose RELU activation function for hidden layers and sigmoid function for output layers. This is because using sigmoid will lead to vanishing gradients. It means derivative of sigmoid dg(z)=g(z)*(1-g(z)). So, if z is large, g(z) tends to 1. So, the derivative becomes 0. Similarly, if z is small, g(z) tends to 0 leading to derivative becoming 0 again.

Steps involved in Developing a NN model:

Architecture: Choosing the neural network architecture i.e., number of hidden layers, number of units in hidden layers. In fact, these two are the hyperparameters which are needed to be tuned for better results and improved accuracy.

Parameter Initialization: Initialize parameters randomly. While implementing, we used dictionaries to store parameters and gradients.

Forward propagation: Once weights are initialized, input X is propagated through hidden units at each layer and produce the output that. It usually contains two steps:

1. Linear Forward
2. Linear-Activation Forward: Result of Linear Forward step is processed into activation function to produce the output of the layer. Activation function at hidden layers is generally chosen as RELU to avoid vanishing gradient concept and activation function at the output layer is sigmoid as Sigmond's range will always be within the range of [0,1] which can be interpreted in a probabilistic way.

Compute Cost: We use cost function which is used in logistic regression (Cross-Entropy cost).

Backward propagation: This is the most important step in training as it gives the gradients of the parameters and bias which are used in gradient descent step to update weights.

Update Weights: The final step of the training process is to update weights which we get during backward propagation. We use gradient descent to update weights.

2.2 The MNIST Database

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 cross-validation 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 centre 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.

Chapter 3:-System Requirements

3.1) Hardware Requirements

- Fast computer processing
- Sufficient network bandwidth
- Adequate RAM

3.2) Software Requirements

- Operating system : Windows
- Jupyter notebook or Google Colab

3.3) Technologies used

Keras:-

Keras is an open-source software library that provides a Python interface for artificial neural networks. Keras acts as an interface for the TensorFlow library. Designed to enable fast experimentation with deep neural networks, it focuses on being user-friendly, modular, and extensible.

NumPy:-

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

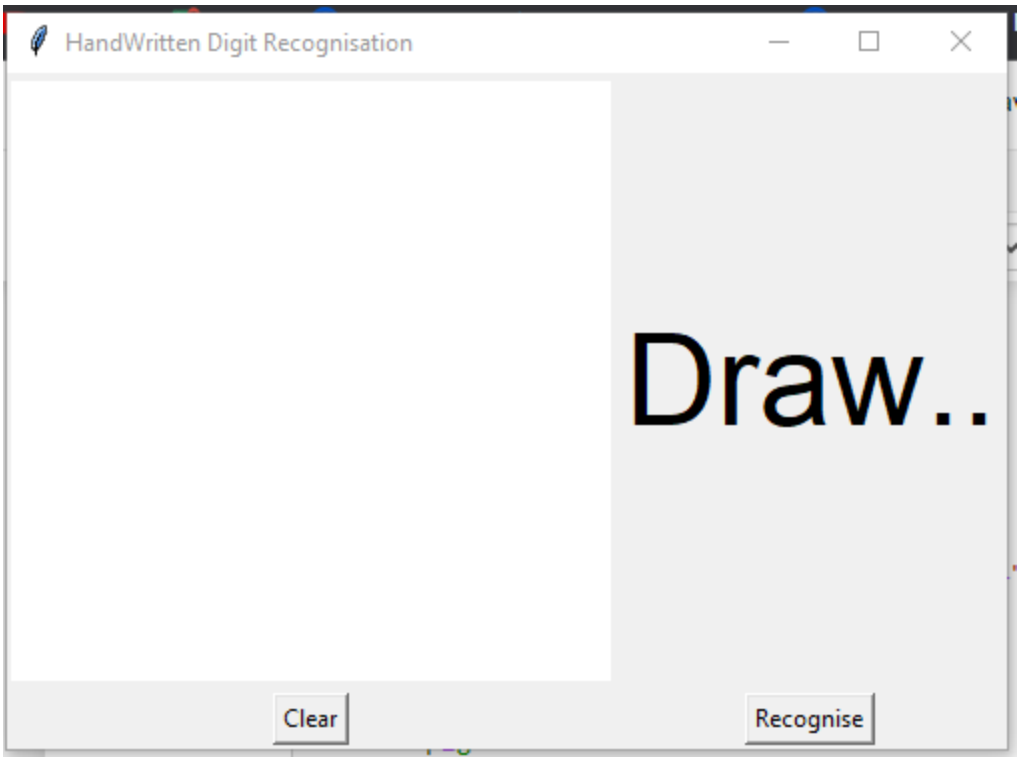
Tkinter:-

Tkinter is a Python binding to the Tk GUI toolkit. It is the standard Python interface to the Tk GUI toolkit, and is Python's de facto standard GUI. Tkinter is included with standard GNU/Linux, Microsoft Windows and macOS installs of Python.

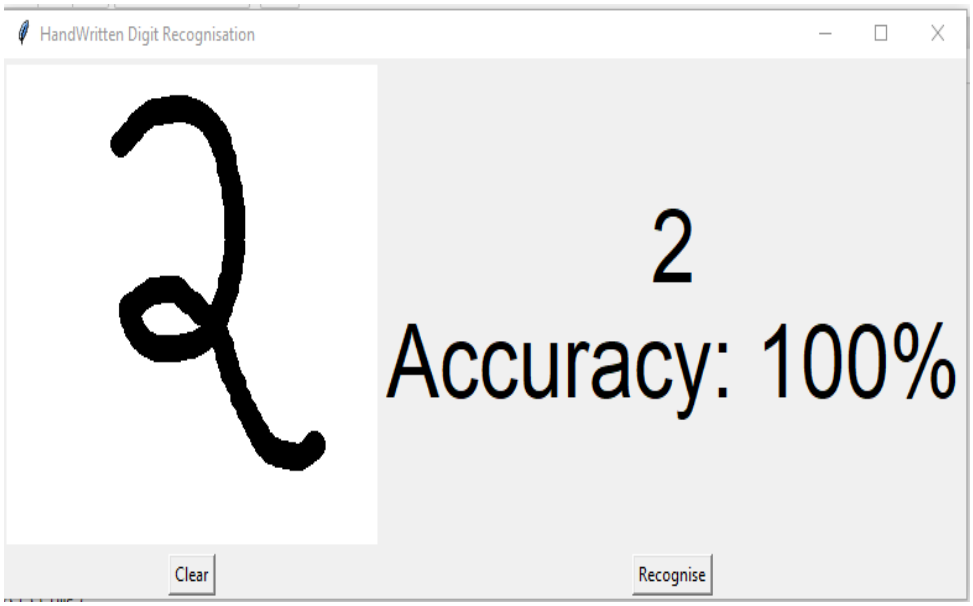
win32:-

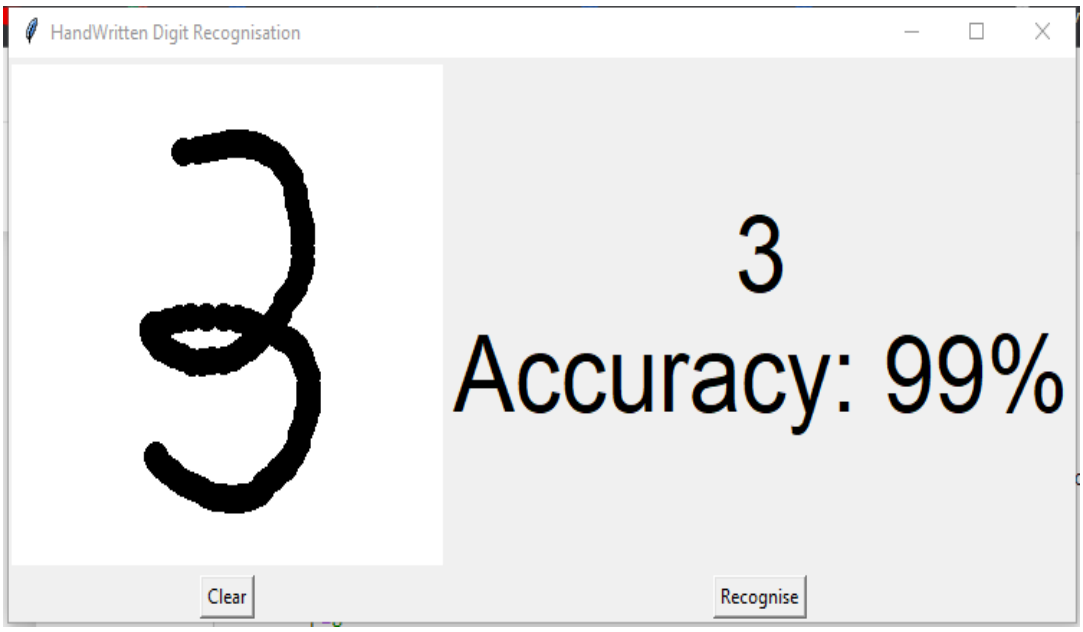
Alternatively referred to as the Windows API and WinAPI, Win32 is the main set of Microsoft Windows APIs used for developing 32-bit applications. These APIs are responsible for functions in the following categories: Administration and Management - Install, configure, and service applications or systems.

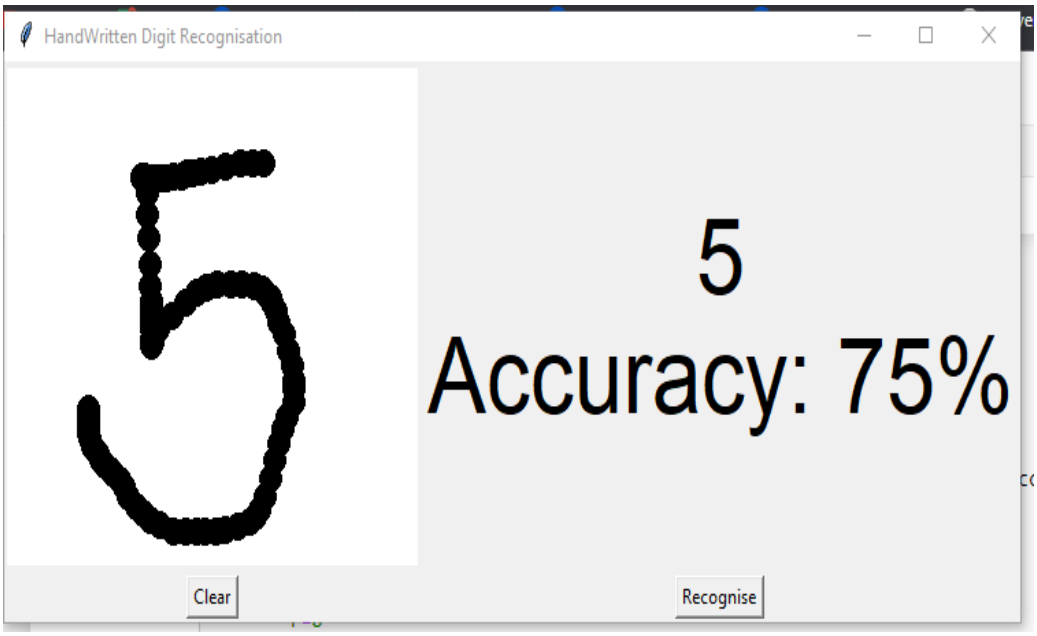
Chapter 4:- Project Screenshots


HandWritten Digit Recognisation
Draw..
Clear
Recognise


HandWritten Digit Recognisation
1
Accuracy: 97%
Clear
Recognise


HandWritten Digit Recognisation
2
Accuracy: 100%
Clear
Recognise


HandWritten Digit Recognisation
3
Accuracy: 99%
Clear
Recognise


HandWritten Digit Recognisation
5
Accuracy: 75%
Clear
Recognise

Chapter 5:- Project Limitations

5.1) limitations still existing in the implementation of Handwritten Digit Recognition

Recognition of handwriting digits is a difficult task because of the different degrees and styles of writing that can be changed between the inter-digits and between intra-digits. The most challenging part of the handwritten digits recognition is needed to address the variety of writing style. There is numerous variation exist in person to the person writing style.

In a handwriting recognition system, 100% accuracy cannot be expected in practical applications. Because even trained humans are not able to recognize every handwritten digit without a doubt Maximum misclassifications are complex to be recognized for the poor contrast, image text vagueness, complicated image environment, and disrupted text stroke.

Chapter 6:-Future Enhancement

6.1) Future enhancements to be added to the project are briefly listed below:

Machines will power our future. We hope that this has given us a glimpse into how they learn. Here, we have used digit images pixels as features vector and ANN as classifiers for handwritten digits recognition. We have used publicly available MNIST database for evaluating our experiments. From the results, we can see that our experiment result achieved good recognition accuracy.

In future work, we plan to work on more datasets and we will further optimize the parameters of ANN to obtain higher accuracies with low implementation time. Finally, we are interested in using a combination hybrid method of feature extraction with ensemble classifier.

Chapter 7:-References

7.1) References Used:

[1] MNIST Database of Handwritten digits:
http://yann.lecun.com/exdb/mnist/

[2] Sequential model:
https://keras.io/models/sequential/

[3] Concept of TensorFlow:
https://www.tensorflow.org/overview

[4] Sample project Learning:
https://data-flair.training/blogs/python-deep-learning-project-handwritten-digit-recognition/

[5] Concept Learning:
https://www.geeksforgeeks.org/handwritten-digit-recognition-using-neural-network/
https://machinelearningmastery.com/handwritten-digit-recognition-using-convolutional-neural-networks-python-keras/

7.2) YouTube Video of Learning

https://www.youtube.com/watch?v=mKSWAlvXSmw

https://www.youtube.com/watch?v=7liA2VWzCQA

https://www.youtube.com/watch?v=tPYj3fFJGjk&t=5794s

https://www.youtube.com/watch?v=aircAruvnKk

https://www.youtube.com/watch?v=IHZwWFHWa-w

https://www.youtube.com/watch?v=Ilg3gGewQ5U

https://www.youtube.com/watch?v=tIeHLnjs5U8

https://www.youtube.com/watch?v=uuuzvOEC0zw&t=3254s

https://www.youtube.com/watch?v=Qn5DDQn0fx0

https://www.youtube.com/watch?v=tpCFfeUEGs8&list=WL&index=129