Project Report

on

Handwritten Digit Recognition

Natural Language Processing

By

**J Keerthana(190330093)**

**P Akhila(190330012)**

under the supervision of

# **Mrs. Anuradha Nandula**



Department of Computer Science and Engineering

K L University Hyderabad,

Aziz Nagar, Moinabad Road, Hyderabad – 500 075, Telangana, India.

May, 2022

**Declaration**

The Project Report entitled “**Handwritten Digit Recognition**“is a record of bonafide work of **P Akhila(190330012) J Keerthana(190330093)** submitted as a requirement for the completion of the course **Natural Language Processing** in the Department of Computer Science and Engineering to the K L University, Hyderabad. The results embodied in this report have not been copied from any other Departments/University/Institute.

(P Akhila)

(J Keerthana)

## Certificate

This is to certify that the Project Report entitled “**HandWritten Digit Recognition**” is being submitted by **J Keerthana (190330093), P Akhila(190330012)** as a requirement for the completion of the course **Natural Language Processing** in the Department of Computer Science and Engineering, K L University Hyderabad is a record of bonafide work carried out under our guidance and supervision.

The results embodied in this report have not been copied from any other departments/University/Institute.

## Signature of theSupervisor

(**Mrs. Anuradha Nandula)**

## Signature oftheHOD Signature of the Examiner

**ACKNOWLEDGEMENT**

First and foremost, we thank the lord almighty for all his grace & mercy showered upon us, for completing this project successfully.

We take grateful opportunity to thank our beloved Founder and Chairman who has given constant encouragement during our course and motivated us to do this project. We are grateful to our Principal **Dr. L. Koteswara Rao** who has been constantly bearing the torch for all the curricular activities undertaken by us.

We pay our grateful acknowledgement & sincere thanks to our Head of the Department **Dr. Chiranjeevi Manike** for his exemplary guidance, monitoring and constant encouragement throughout the course of the project. We thank **Mrs. Anuradha Nandula of** our department who has supported throughout this project holding a position of supervisor.

We whole heartedly thank all the teaching and non-teaching staff of our department without whom we won’t have made this project a reality. We would like to extend our sincere thanks especially to our parents, our family members and friends who have supported us to make this project a grand success.

#### **ABSTRACT**

The handwritten digit recognition problem becomes one of the most famous problems in machine learning and computer vision applications. Many machine learning techniques have been employed to solve the handwritten digit recognition problem. This paper focuses on Neural Network (NN) approaches. The most three famous NN approaches are deep neural network (DNN), deep belief network (DBN) and convolutional neural network (CNN). In this paper, the three NN approaches are compared and evaluated in terms of many factors such as accuracy and performance. Recognition accuracy rate and performance, however, is not the only criterion in the evaluation process, but there are interesting criteria such as execution time. Random and standard dataset of handwritten digit have been used for conducting the experiments. The results show that among the three NN approaches, DNN is the most accurate algorithm; it has 98.08% accuracy rate. However, the execution time of DNN is comparable with the other two algorithms. On the other hand, each algorithm has an error rate of 1-2% because of the similarity in digit shapes, specially, with the digits (1,7), (3,5), (3,8), (8,5) and (6,9).

**TABLE OF CONTENTS**

|  |  |
| --- | --- |
| S.No | Title |
| 1 | Introduction |
| 2 | Terminology |
| 3 | Related Work |
| 4 | Objective |
| 5 | Requirements and Datasets |
| 6 | Modules |
| 7 | Proposed Solution ( Implementation) |
| 8 | Results discussion |
| 9 | Conclusion |
| 10 | References |

# **INTRODUCTION**

Handwritten digit recognition is the ability of a computer to recognize the human handwritten digits from different sources like images, papers, touch screens, etc, and classify them into 10 predefined classes (0-9). This has been a topic of boundless-research in the field of deep learning. Digit recognition has many applications like number plate recognition, postal mail sorting, bank check processing, etc [2]. In Handwritten digit recognition, we face many challenges because of different styles of writing of different peoples as it is not an Optical character recognition. This research provides a comprehensive comparison between different machine learning and deep learning algorithms for the purpose of handwritten digit recognition. For this, we have used Support Vector Machine, Multilayer Perceptron, and Convolutional Neural Network. The comparison between these algorithms is carried out on the basis of their accuracy, errors, and testing-training time corroborated by plots and charts that have been constructed using matplotlib for visualization.

The accuracy of any model is paramount as more accurate models make better decisions. The models with low accuracy are not suitable for real-world applications. Ex- For an automated bank cheque processing system where the system recognizes the amount and date on the check, high accuracy is very critical. If the system incorrectly recognizes a digit, it can lead to major damage which is not desirable. That’s why an algorithm with high accuracy is required in these realworld applications. Hence, we are providing a comparison of different algorithms based on their accuracy so that the most accurate algorithm with the least chances of errors can be employed in various applications of handwritten digit recognition.

# **METHODOLOGY**

**FEATURE EXTRACTION :** In machine learning , pattern recognition, and image processing , feature extraction starts from an initial set of measured data and builds derived values (features) intended to be informative and non-redundant, facilitating the subsequent learning and generalization steps, and in some cases leading to better human interpretations. Feature extraction is related to dimensionality reduction. When the input data to an algorithm  is too large to be processed and it is suspected to be redundant (e.g. the same measurement in both feet and meters, or the repetitiveness of images presented as pixels then it can be transformed into a reduced set of features (also named a feature vector Determining a subset of the initial features is called feature selection. The selected features are expected to contain the relevant information from the input data, so that the desired task can be performed by using this reduced representation instead of the complete initial data.

CNN is a deep learning algorithm that is widely used for image recognition and classification. It is a class of deep neural networks that require minimum pre-processing. It inputs the image in the form of small chunks rather than inputting a single pixel at a time, so the network can detect uncertain patterns (edges) in the image more efficiently. CNN contains 3 layers namely, an input layer, an output layer, and multiple hidden layers which include Convolutional layers, Pooling layers(Max and Average pooling), Fully connected layers (FC), and normalization layers . CNN uses a filter (kernel) which is an array of weights to extract features from the input image. CNN employs different activation functions at each layer to add some non-linearity. As we move into the CNN, we observe the height and width decrease while the number of channels increases. Finally, the generated column matrix is used to predict the output

# 

# 

# 

# **DATASETS**

We are going to use the MNIST dataset for the implementation of a handwritten digit recognition app. To implement this we will use a special type of deep neural network called Convolutional Neural Networks. In the end, we will also build a Graphical user interface(GUI) where you can directly draw the digit and recognize it straight away.

The MNIST database of handwritten digits, available from this page, has a training set of 60,000 examples, and a test set of 10,000 examples. It is a subset of a larger set available from NIST. The digits have been size-normalized and centered in a fixed-size image.

The MNIST database was constructed from NIST's Special Database 3 and Special Database 1 which contain binary images of handwritten digits. NIST originally designated SD-3 as their training set and SD-1 as their test set. However, SD-3 is much cleaner and easier to recognize than SD-1. The reason for this can be found on the fact that SD-3 was collected among Census Bureau employees, while SD-1 was collected among high-school students. Drawing sensible conclusions from learning experiments requires that the result be independent of the choice of training set and test among the complete set of samples. Therefore it was necessary to build a new database by mixing NIST's datasets.

The MNIST training set is composed of 30,000 patterns from SD-3 and 30,000 patterns from SD-1. Our test set was composed of 5,000 patterns from SD-3 and 5,000 patterns from SD-1. The 60,000 pattern training set contained examples from approximately 250 writers. We made sure that the sets of writers of the training set and test set were disjoint.



# 

# 

# **IMPORT LIBRARIES AND DATASET**

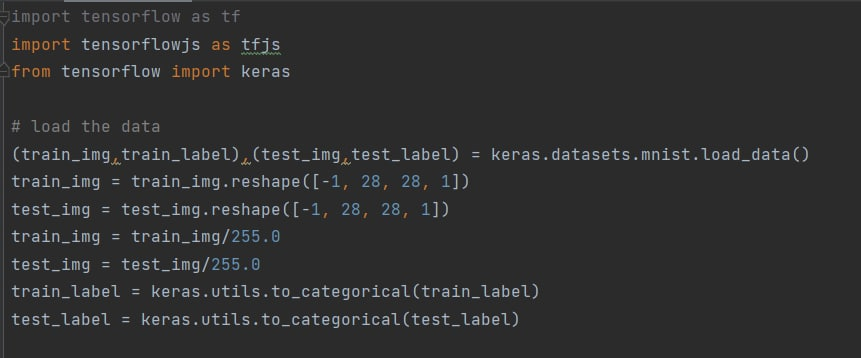
# Libraries required

1. Keras
2. Tensorflow
3. OpenCV
4. Sklearn
5. Numpy

# 

We import all the needed modules for training our model. We can easily import the dataset and start working on that because the Keras library already contains many datasets and MNIST is one of them. We call mnist.load\_data() function to get training data with its labels and also the testing data with its labels.

Model cannot take the image data directly so we need to perform some basic operations and process the data to make it ready for our neural network. The dimension of the training data is (60000\*28\*28). One more dimension is needed for the CNN model so we reshape the matrix to shape (60000\*28\*28\*1).



# **MODEL CREATION**

# We use CNN (convolution neural network) for this project

# We will use the Adadelta optimizer for the model compilation.

# We use model architecture as keras sequential

# Cross entropy is used as loss function

# .

# A**Sequencial** model is appropriate for**a plain stack of layers**where each layer has**exactly one input tensor and one output tensor**.

# 

# 

# **TRAINING MODEL**

# To start the training of the model we can simply call the model.fit() function of Keras. It takes the training data, validation data, epochs, and batch size as the parameter.

# The training of model takes some time. After succesful model training, we can save the weights and model definition in the ‘mnist.h5’ file.

# 

## OUTPUT

## 

# 

References

[1] “Handwriting recognition”: https : //en.wikipedia.org/wiki/Handwritingrecognition

[2] “What can a digit recognizer be used for?”: https : //www.quora.com/W hat−can−a−digit−recognizer− be − used − f or

[3] ”Handwritten Digit Recognition using Machine Learning Algorithms”, S M Shamim, Mohammad Badrul Alam Miah, Angona Sarker, Masud Rana & Abdullah Al Jobair.

[4] ”Handwritten Digit Recognition Using Deep Learning”, Anuj Dutt and Aashi Dutt.

[5] ”Handwritten recognition using SVM, KNN, and Neural networks”, Norhidayu binti Abdul Hamid, Nilam Nur Binti Amir Sharif.

[6] ”Recognition of Handwritten Digit using Convolutional Neural Network in Python with Tensorflow and Comparison of Performance for Various Hidden Layers”, Fathma Siddique, Shadman Sakib, Md. Abu Bakr Siddique.

[7] ”Advancements in Image Classification using Convolutional Neural Network” by Farhana Sultana, Abu Sufian & Paramartha Dutta.

[8] ”Comparison of machine learning methods for classifying mediastinal lymph node metastasis of non-small cell lung cancer from 18 FFDG PET/CT images” by Hongkai Wang, Zongwei Zhou, Yingci Li, Zhonghua Chen, Peiou Lu, Wenzhi Wang, Wanyu Liu, and Lijuan Yu.

[9] https : //github.com/rishikakushwah16/SV M digit recognition using MNIST dataset

[10] “Support Vector Machine Algorithm”: https : //www.javatpoint.com/machine−learning−support− vector − machine − algorithm

[11] <https://machinelearningmastery.com/neural-networks-crash-course/>

[12] “An Introduction to Convolutional Neural Networks”: researchgate.net/publication/285164623 An Introduction to Convolutional Neural Networks