**PROJECT BASED REPORT**

ON

**CHARACTER RECOGNISATION OF MNIST**

**DATASET USING DEEP LEARNING**

(CSE V SEMESTER)

2021-2022



**SUBMITTED BY -**

Ms. SANA AFREEN

University Roll No: 2014831

C.S.E – V – SEMESTER

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

**GRAPHIC ERA DEEMED TO BE UNIVERSITY, DEHRADUN**

**ABSTRACT:**

In this project, we have presented an implementation of Handwritten Digit Recognition. HCR or Handwritten Character Recognition is one of the most challenging tasks in machine learning as the handwriting of every individual on the Earth is unique. The **handwritten digit recognition** is the capability of computer applications to **recognize** the human **handwritten digits.** It is a hard task for the **machine** because **handwritten digits** are not perfect and can be made with many different shapes and sizes. The **handwritten digit recognition system** is a way to tackle this problem which uses the image of a **digit** and recognizes the **digit** present in the image. So it’s quite challenging to train a model that can predict handwritten text with high accuracy. We have developed a CNN (Convolutional Neural Network) model that can recognize digits from images with a 99.83% accuracy.

**CONTENTS:**

* Introduction
* Methodology

1. Description of the dataset
2. Convolutional Neural Network Architecture
3. Multi layered Preceptron
4. Visualization

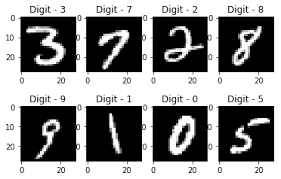
* Implementation
* Result and Discussion
* Conclusion and Future work

**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. 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 report provides deep learning algorithm for the purpose of handwritten digit recognition using Convolutional Neural Network(CNN). We have applied a different type of Convolutional Neural Network algorithm on Modified National Institute of Standards and Technology (MNIST) dataset using Tensorflow, a Neural Network library written in python. The main purpose is to analyze the variation of outcome results for using a different combination of hidden layers of Convolutional Neural Network. Stochastic gradient and backpropagation algorithm are used for training the network and the forward algorithm is used for testing.

**METHODOLOGY:**

1. **DESCRIPTION OF THE DATASET-**

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. 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 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.

*Fig. 1 Plotting of some random MNIST Handwritten digits*

1. **CONVOLUTIONAL NEURAL NETWORK ARCHITECTURE-**

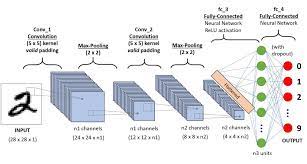
CNN is a deep learning algorithm that is widely used for image recognition and classification. It is a class of deep neural networks that require minimum pre-processing. It inputs the image in the form of small chunks rather than inputting a single pixel at a time, so the network can detect uncertain 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.

1. Input Layer- The input data is loaded and stored in the input layer. This layer describes the height, width and number of channels (RGB information) of the input image.
2. Hidden Layer- The hidden layers are the backbone of CNN architecture. They perform a feature extraction process where a series of convolution, pooling and activation functions are used. The distinguishable features of handwritten digits are detected at this stage.
3. Convolutional Layer- The convolutional layer is the first layer placed above the input image. It is used for extracting the features of an image. The n x n input neurons of the input layer are convoluted with an

m x m filter and in return deliver (n − m + 1) × (n – m + 1) as output.

The concept of padding is introduced in CNN architecture to get more accuracy. Padding is introduced to control the shrinking of the output of the convolutional layer. The output from the convolutional layer is a feature map, which is smaller than the input image.

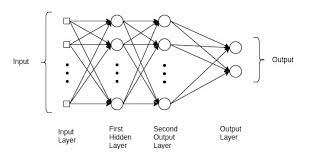
1. Pooling Layer- A pooling layer is added between two convolutional layers to reduce the input dimensionality and hence to reduce the computational complexity.Pooling allows the selected values to be passed to the next layer while leaving the unnecessary values behind. The pooling layer also helps in feature selection and in controlling overfitting. The pooling operation is done independently. It works by extracting only one output value from the tiled non-overlapping sub-regions of the input images.
2. Activation Layer- Just like regular neural network architecture, CNN architecture also contains the activation function to introduce the non-linearity in the system. The sigmoid function, rectified linear unit (ReLu) and Softmax are some famous choices among various activation functions exploited extensively in deep learning models. The activation function used in the present work is the non-linear rectified linear unit (ReLu) function, which has output 0 for input less than 0 and raw output otherwise. Some advantages of the ReLu activation function are its similarity with the human nerve system, simplicity in use and ability to perform faster training for larger networks.
3. Classification Layer- The classification layer is the last layer in CNN architecture. It is a fully connected feed forward network, mainly adopted as a classifier. The neurons in the fully connected layers are connected to all the neurons of the previous layer. This layer calculates predicted classes by identifying the input image, which is done by combining all the features learned by previous layers. The number of output classes depends on the number of classes present in the target dataset. In the present work, the classification layer uses the softmax activation function for classifying the generated features of the input image received from the previous layer into various classes based on the training data.



*Fig.2 The Architectural Design of CNN layers*

1. **MULTILAYERED PERCEPTRON-**

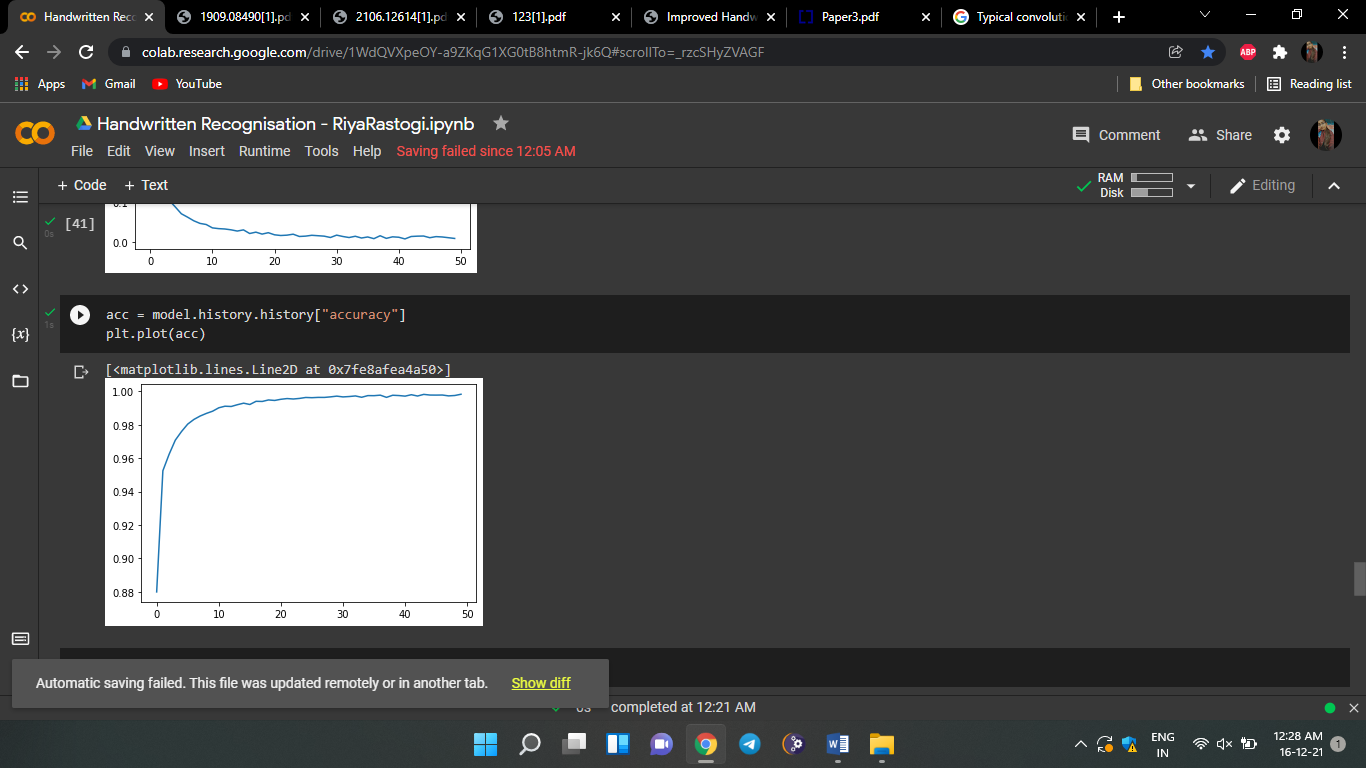
MLP is a class of feedforward artificial neural networks (ANN). It consists of three layers: input layer, hidden layer and output layer. Each layer consists of several nodes that are also formally referred to as neurons and each node is interconnected to every other node of the next layer. In basic MLP there are 3 layers but the number of hidden layers can increase to any number as per the problem with no restriction on the number of nodes. The number of nodes in the input and output layer depends on the number of attributes and apparent classes in the dataset respectively. The particular number of hidden layers or numbers of nodes in the hidden layer is difficult to determine due to the model erratic nature and therefore selected experimentally. Every hidden layer of the model can have different activation functions for processing. For learning purposes, it uses a supervised learning technique called backpropagation. In the MLP, the connection of the nodes consists of a weight that gets adjusted to synchronize with each connection in the training process of the model**.**

****

*Fig.3 The basic architecture of the Multilayer Perceptron*

1. **VISUALIZATION-**

We have used the MNIST dataset (i.e. handwritten digit dataset) to compare different level algorithm of deep and machine learning on the basis of execution time, complexity, accuracy rate, number of epochs and number of hidden layers (in the case of deep learning algorithms). To visualize the information obtained by the detailed analysis of algorithms we have used bar graphs and tabular format charts using module matplotlib, which gives us the most precise visuals of the step by step advances of the algorithms in recognizing the digit. The graphs are given at each vital part of the programs to give visuals of each part to bolster the outcome.



*Fig.4 Accuracy rate*

**IMPLEMENTATION:**

https://github.com/2286-SANA/Hand-Written-Digit-Recognition-/blob/main/HWDR.ipynb

**RESULT:**

In this project, the variations of accuracies for handwritten digit were observed for 50 epochs by varying the hidden layers. The accuracy curves were generated for the six cases for the different parameter using CNN MNIST digit dataset. CNN has been applied on the MNIST dataset in order to observe the variation of accuracies for handwritten digits. The accuracies are obtained using Tensorflow in python. Training and validation accuracy for 50 different epochs were observed exchanging the hidden layers for various combinations of convolution and hidden layers by taking the batch size 100 for all the cases. The layers were taken randomly in a periodic sequence so that each case behaves differently during the experiment. The maximum and minimum accuracies were observed for different hidden layers variation with a batch size of 100. Among all the observation, we achieved a recognition rate of 99.83% with the Adam optimizer for the MNIST database, which is better than all previously reported results. The effect of increasing the number of convolutional layers in CNN architecture on the performance of handwritten digit recognition is clearly presented through the experiments.

**CONCLUSION AND FUTUREWORK:**

In this project, we have implemented model for handwritten digit recognition using MNIST datasets, based on deep and machine learning algorithms. CNN gave the most accurate results for handwritten digit recognition. So, this makes us conclude that CNN is best suitable for any type of prediction problem including image data as an input. Next, by comparing execution time of the algorithms we have concluded that increasing the number of epochs without changing the configuration of the algorithm is useless because of the limitation of a certain model and we have noticed that after a certain number of epochs the model starts overfitting the dataset and give us the biased prediction.

The future development of the applications based on algorithms of deep and machine learning is practically boundless. In future, different architectures of CNN, namely, hybrid CNN, viz., CNN-RNN and CNN-HMM models, and domain-specific recognition systems, can be investigated. Evolutionary algorithms can be explored for optimizing CNN learning parameters, namely, the number of layers, learning rate and kernel sizes of convolutional filters.

THANK YOU