IJARSCT

ISSN (Online) 2581-9429

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

Impact Factor: 6.252

Volume 2, Issue 7, May 2022

Digit Recognition using Deep Learning Algorithms

Shasshank Rana[1], Vishal Yadav[2], Parichit Kukreti[3]
Department of Computer Science and Engineering
Dronacharya Group of Institutions Greater Noida, India

Abstract: *The reliance of human over machine were never this high. Nowadays machines are used in everyone's daily life as a part of it. From recognition objects in photographs to adding sounds automatically. Likewise, handwritten recognition is a vast area for research and development with unlimited no of possibilities that could be attained. It is the ability of computer to recognise different input taken by it. In this paper we have performed recognition with the help of MNIST dataset. Our main job is to compare the accuracy with the execution time to get best possible result.*

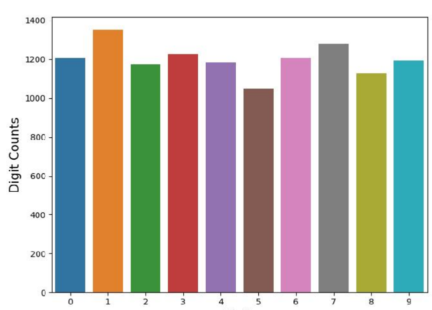
Keywords: Sign Language, Deep Learning , MNIST dataset.

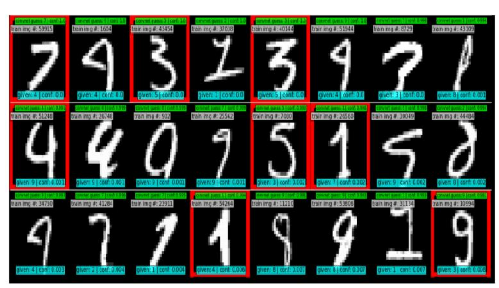
I. INTRODUCTION

Handwritten digit recognition is the ability of a computer to recognise the human handwritten digits from different sources like paper, images, touch screen etc. This has been a topic of a boundless research area in the field of deep learning. It has many applications like no plate recognition, face recognition in Cameras and many more. Identification of digit from where best discriminating features can be extracted is one of the major tasks in the area of digit recognition system. Handwritten digit dataset are vague in nature because there may not always be sharp and perfectly straight lines. The main goal in digit recognition is feature extraction is to remove the redundancy from the data and gain a more effective embodiment of the word image through a set of numerical attributes.

1.1 Dataset.

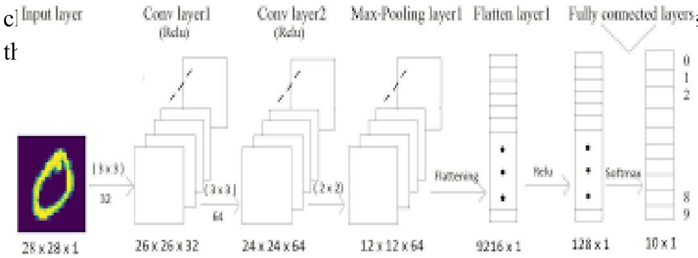
Handwritten character recognition is an expansive research area that already contains detailed ways of implementation which include major learning datasets, popular algorithms, features scaling and feature extraction methods. MNIST dataset (Modified National Institute of Standards and Technology database) is the subset of the NIST dataset which is a combination of two of NIST's databases: Special Database 1 and Special Database 3. Special Database 1 and Special Database 3 consist of digits written by high school students and employees of the United States Census Bureau, respectively. MNIST contains a total of 70,000 handwritten digit images (60,000 - training set and 10,000 - test set) in 28x28 pixel bounding box and anti-aliased All these images have corresponding Y values which apprises what the digit is.

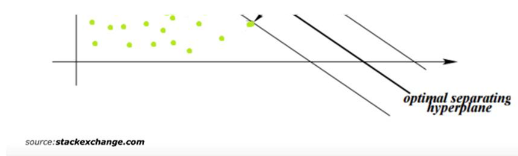




1.2 Support Vector Machine

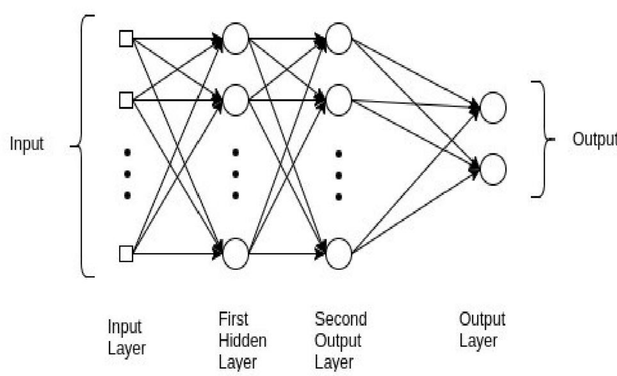
Support Vector Machine (SVM) is a supervised machine learning algorithm. In this, we generally plot data items in n-dimensional space where n is the number of features, a particular coordinate represents the value of a feature, we perform the classification by finding the hyperplane that distinguishes the two classes. It will choose the hyperplane that separates the classes correctly. SVM c se extreme cases are called support vectors, and hence th





1.3 Multi-layered Perceptron

A multilayer 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



II. CONVOLUTIONAL NEURAL NETWORK

The implementation of handwritten digit recognition by Convolutional Neural Network [15] is done using Keras. It is an open-source neural network library that is used to design and implement deep learning models. From Keras, we have used a Sequential class which allowed us to create model layer-by-layer. This layer uses a matrix to convolve around the input data across its height and width and extract features from it. This matrix is called a Filter or Kernel. The values in the filter matrix are weights. We have used 32 filters each of the dimensions (3,3) with a stride of 1. Stride determines the number of pixels shifts. Convolution of filter over the input data gives us activation maps whose dimension is given by the formula: $((N + 2P - F)/S) + 1$ where N= dimension of input image, P= padding, F= filter dimension and S=stride. In this layer, Depth (number of channels) of the output image is equal to the number of filters used. To increase the non-linearity, we have used an activation function that is Relu [21]. Next, another convolutional layer is used in which we have applied 64 filters of the same dimensions (3,3) with a stride of 1 and the Relu function.

III. IMPLEMENTATION

To compare the algorithms based on working accuracy, execution time, complexity, and the number of epochs (in deep learning algorithms) we have used three different classifiers: • Support Vector Machine Classifier • ANN - Multilayer Perceptron Classifier • Convolutional Neural Network Classifier We have discussed in detail about the implementation of each algorithm explicitly below to create a flow of this analysis to create a fluent and accurate comparison.

3.1 Pre-Processing

Pre-processing is an initial step in the machine and deep learning which focuses on improving the input data by reducing unwanted impurities and redundancy. To simplify and break down the input data we reshaped all the images present in the dataset in 2-dimensional images i.e (28,28,1). Each pixel value of the images lies between 0 to 255 , we Normalized these pixel values by converting the dataset into 'float32' and then dividing by 255.0 so that the input features will range between 0.0 to 1.0. Next, we performed one-hot encoding to convert the y values into zeros and ones, making each number categorical, for example, an output value 4 will be converted into an array of zero and one which is [0,0,0,0,1,0,0,0,0,0].

3.2 Support Vector Machine

The SVM in scikit-learn [16] supports both dense (numpy.ndarray and convertible to that by numpy.asarray) and sparse (any scipy.sparse) sample vectors as input. In scikit-learn, SVC, NuSVC and LinearSVC are classes capable of performing multi-class classification on a dataset. In this paper we have used Linear-SVC for classification of MNIST datasets that make use of a Linear kernel implemented with the help of LIBLINEAR.

After this, plotting of some samples as well as converting into matrix followed by normalization and scaling of features have been done. Finally, we have created a linear SVM model and confusion matrix that is used to measure the accuracy of the mode.

3.3 Multi-layered perception -

The implementation of Handwritten digits recognition by Multilayer perceptron [18] which is also known as feedforward artificial neural network is done with the help of Keras module to create an MLP model of Sequential class and add respective hidden layers with different activation function to take an image of 28x28 pixel size as input. After creating a sequential model, we added a Dense layer of different specifications and Drop out layers as shown in the image below. The block diagram is given here for reference. Once you have the training and test data, you can follow these steps to train a neural network in Keras.

3.4 Convolutional Neural Network

The implementation of handwritten digit recognition by Convolutional Neural Network [15] is done using Keras. It is an open-source neural network library that is used to design and implement deep learning models. From Keras, we have used a Sequential class which allowed us to create model layer-by-layer. The dimension of the input image is set to 28(Height) 28(Width), 1(Number of channels). Next, we created the model whose first layer is a Conv layer [20]. This layer uses a matrix to convolve around the input data across its height and width and extract features from it. This matrix is called a Filter or Kernel. The values in the filter matrix are weights. We have used 32 filters each of the dimensions (3,3) with a stride of 1. Stride determines the number of pixels shifts. Convolution of filter over the input data gives us activation maps whose dimension is given by the formula: $((N + 2P - F)/S) + 1$ where N= dimension of input image, P= padding, F= filter

IJARSCT

ISSN (Online) 2581-9429

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

IJARSCT
Impact Factor: 6.252

Volume 2, Issue 7, May 2022

dimension and S=stride. In this layer, Depth (number of channels) of the output image is equal to the number of filters used. To increase the non-linearity, we have used an activation function that is Relu [21]. Next, another convolutional layer is used in which we have applied 64 filters of the same dimensions (3,3) with a stride of 1 and the Relu function

IV. RESULT

After implementing all the three algorithms that are SVM, MLP and CNN we have compared their accuracies and execution time with the help of experimental graphs for perspicuous understanding. We have taken into account the Training and Testing Accuracy of all the models stated above. After executing all the models, we found that SVM has the highest accuracy on training data while on testing dataset CNN accomplishes the utmost accuracy. Additionally, we have compared the execution time to gain more insight into the working of the algorithms. Generally, the running time of an algorithm depends on the number of operations it has performed. So, we have trained our deep learning model up to 30 epochs and SVM models according to norms to get the apt outcome. SVM took the minimum time for execution while CNN accounts for the maximum running time.

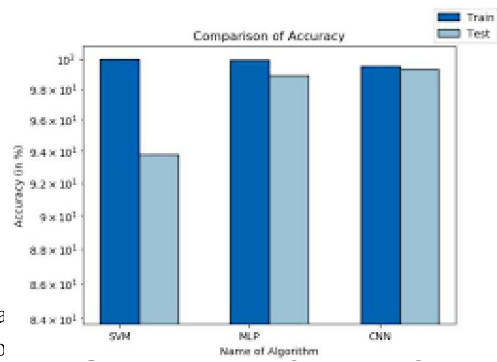
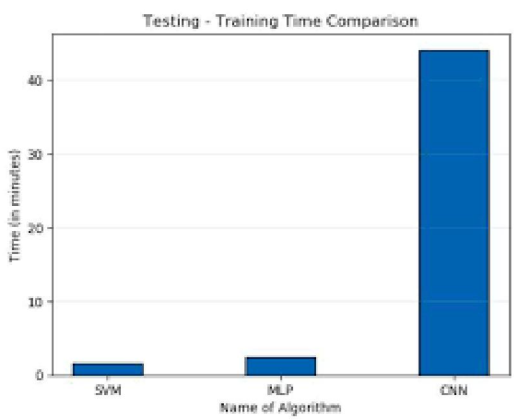


Figure - This table represents the overa ... ns 5 columns, the 2nd column represents model name, 3rd and 4th co ... cy of models, and 5th column represents execution time of models.



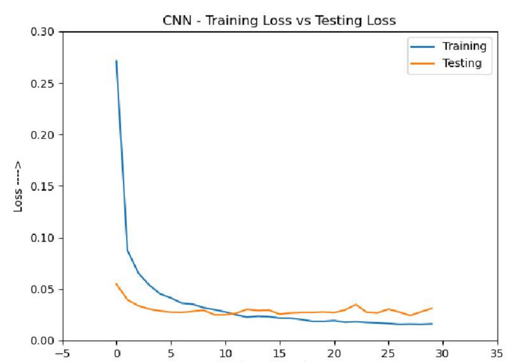


Fig- ... in: 99 ... .85%), CNN

Fig - Bar graph showing execution time comparison of SVM, MLP and CNN (SVM: 1.58 mins, MLP: 2.53 mins, CNN: 44.05 mins). we visualized the performance measure of deep learning models and how they

Model	Training Rate	Testing Rate	Execution Time
SVM	99.98%	94.005%	1:35 min

IJARSCT

ISSN (Online) 2581-9429

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

Impact Factor: 6.252

Volume 2, Issue 7, May 2022

MLP	99.92%	98.85%	2:32 min
CNN	99.53%	99.31%	44:.02 min

"This table represents the overall performance for each model. The table contains 5 columns, the 2nd column represents model name, 3rd and 4th column represents the training and testing accuracy of models, and 5th column represents execution time of models."

ameliorated their accuracy and reduced the error rate concerning the number of epochs.

Fig - Graph illustrating the transition of training loss with increasing number of epochs in Multilayer Perceptron (Loss rate v/s Number of epochs). Fig - Graph illustrating the transition of training loss of CNN with increasing number of epochs (Loss rate v/s Number of epochs).

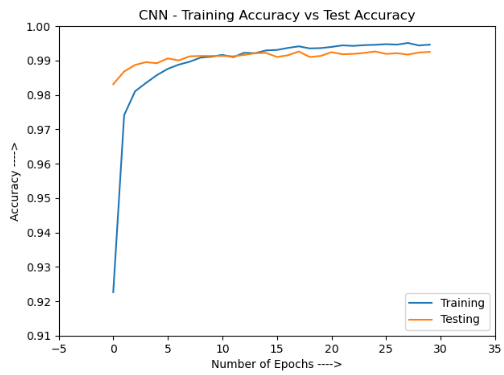


Fig - Graph illustrating the transition of training accuracy of CNN with increasing number of epochs (Accuracy v/s Number of epochs).

V. CONCLUSION

In this research, we have implemented three models for handwritten digit recognition using MNIST datasets, based on deep and machine learning algorithms. We compared them based on their characteristics to appraise the most accurate model among them. Support vector machines are one of the basic classifiers that's why it's faster than most algorithms and in this case, gives the maximum training accuracy rate but due to its simplicity, it's not possible to classify complex and ambiguous images as accurately as achieved with MLP and CNN algorithms. We have found that 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.

REFERENCES

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

[2]. "Handwritten Digit Recognition using Machine Learning Algorithms", S M Shamim, Mohammad Badrul Alam Miah, Angona Sarker, Masud Rana & Abdullah Al Jobair "Advancements in Image Classification using Convolutional Neural Network" by Farhana Sultana, Abu Sufian & Paramartha Dutta.

[3]. "How Do Convolutional Layers Work in Deep Learning Neural Networks": https: //machinelearningmastery .com/convolutional −layers − f or − deep − learning − neural − networks/