**Journal of Mines, Metals and Fuels,** 71(5): 683-686; 2023. DOI: 10.18311/jmmf/2023/

Print ISSN : 0022-2755

Journal of Mines, Metals and Fuels

Contents available at: www.informaticsjournals.com/index.php/jmmf

# Handwritten Digit Classification and Performance Analysis of Different Machine Learning Algorithms and CNN

**Ayan De[1]*, Saswati Rakshit[2] and Debayan Bhattacharya[1]**

*[1]Computer Science and Engineering, Swami Vivekananda University, West Bengal, India.*
*E-mail: ayande1998@gmail.com / bdebayan@gmail.com*
*Computer Science and Engineering, IIE, Kalyani, West Bengal, India. E-mail: saswatirakshit@gmail.com*

## Abstract

*This work deals with the method of hand written image classification by using machine learning and deep learning. Python along with Tensor Flow framework is used here for developing the classification model. Image classification can be considered as a supervised learning approach which works on a labelled dataset for classification. Though many research works are going on this machine learning based Image classification, yet it is challenging for accurate classification. This paper focuses on handwritten image data classification task using machine learning algorithms namely K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Random Forest classifier (RF). In this paper a comparative analysis is performed on the dataset based on various parameters using different machine learning algorithms. Results are discussed in terms of classification accuracy of different ML based algorithms. From our experiment, it has been noticed that RF Classifier exhibits the highest accuracy of 100% which is followed by SVM having the accuracy of 98% and finally a KNN model having a correctness of 97%. The novelty of this paper is that along with various ML algorithms (RF, KNN, SVM and Naive Bayes) we applied deep learning algorithms (CNN), with three different activation functions and their combinations (RELU, TANH and SIGMOID) along with variation of optimizers (Adam and SGD). Finally performance of the combined optimization method is found to be better than the performance of the individual optimization method.*

***Keywords:*** *Image Classification, K-Nearest Neighbor, Support Vector Machine, Random Forest classifier, CNN, Gaussian Naïve Bayes.*

## 1.0 Introduction

Recently, image classification is in high demand amongst generation builders particularly with the blast of records in distinct elements of enterprise together with e-commerce, automotive, healthcare, and gaming. Handwritten number recognition is a vital hassle in optic individual recognition. With the help of different machine learning algorithms, classification of handwritten images become popular and widely used in various fields for automatic character recognition. One trouble is that despite the fact that unique classifier strategies can classify the images, it's far tough to recognize which of those strategies will excess correct outcomes. In this paper, four generally used classifier strategies specifically Random Forest Classifier, K-Nearest Neighbor (KNN) and Support Vector Machine (SVM) and Naïve Bayes are used. The correctness of each of the models have been analyzed and finally decided the best suited algorithm for our dataset.

*Author for correspondence

## 2.0 Dataset Preparation

The MNIST dataset[1] is used which includes over 70000 images (60000 for training records and 10000 for test records). All these black and white digits are size normalized and centered in a fixed-size image[2]. The dimensionality of each image sample vector is 28×28 = 784, where each element is binary. MNIST handwritten number database is collected from the web page of Yann LeCun (Yann.lecun.com). It has ended up a general for fast- trying out hypotheses of sample character and machine studying algorithms. It carries handwritten number prints for the classifier training and handwritten number prints for the classifier testing, each drawn from the equal distribution. All those black and white handwritten numbers are targeted in a fixed-length image. The dimensionality is 28×28 = 784 for each image wherein every detail is binary. In this paper, first in the stage of preprocessing, segmentation and normalization are performed on handwritten figures to ease the classification process[3].

Figure 1: MNIST dataset where each grayscale image is 28×28-pixels[4].

## 3.0 Algorithms used for Classification

In this section, the used ML approaches such as Random Forest Classification, K- Nearest Neighbor and Support Vector Machine are discussed.

### A. Random Forest Classifier

A random forest is a machine learning algorithm, where many decision trees are drawn which fit on dataset. The output from all decision trees have been considered and finally the class is decided by majority votes as the final output of the RF classifier. The Ginni Index is used for feature selection by the classifier, as it measures the contamination of a feature with respect to the classes.

In the mentioned diagram, n-number of trees can be drawn from a specific instance and then concluded into different classes as final output.

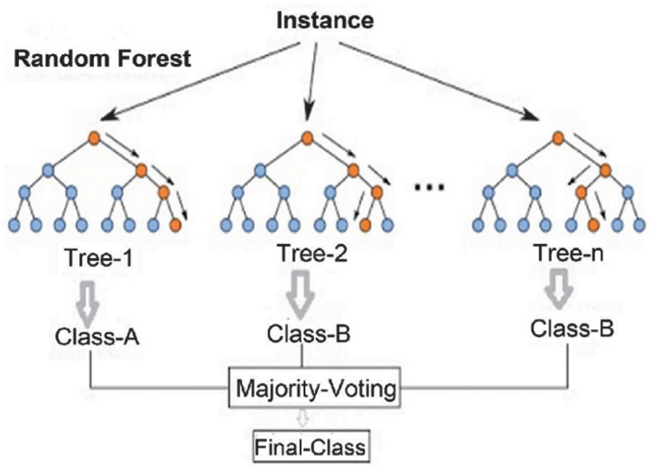

Figure 2: Example images of random forest classifier[5]

### B. K-Nearest Neighbor

The K-Nearest Neighbor or KNN[6,10,11] is a supervised learning approach where both regression as well as classification problems are solved. For industry specific classification KNN is widely used. The algorithm has trust on assumption to be true for further use. Distance, adjacency or closeness is calculated by measuring the distance between two points on a graph. Though there are many methods for distance calculation but still now, Euclidean distance is widely used.

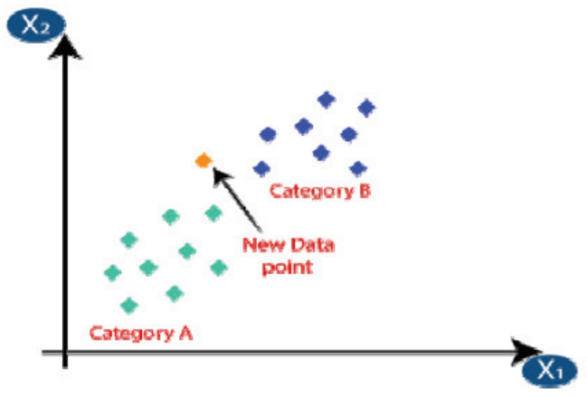

Figure 3: Example image of KNN classifier[7]

### C. Support Vector Machine

Support vector machines (SVM) are supervised ML approach[8]. A vital benefit of SVM over other classification algorithms is increased degree of accuracy. SVM uses hyperplane for classification. This hyperactive-plane is sustained by the use of support vectors. Periphery of the hyperactive-plane is insured by those vectors as much as possible.

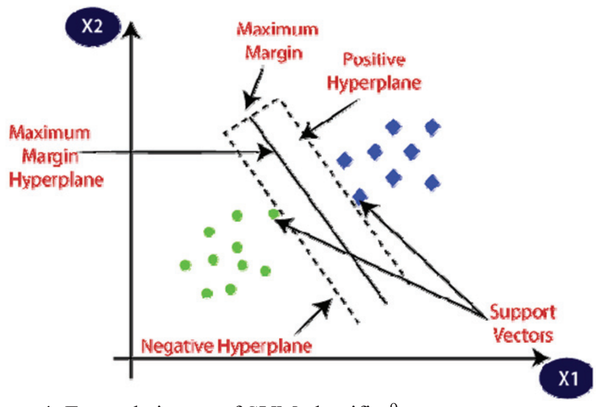

Figure 4: Example image of SVM classifier[9]

## D. Deep Learning Convolution Neural Networks

An artificial neural network represents the structure of an earthborn human brain modelled on the computer vision. It consists of neurons and synapses, weights, impulses, and functions organized into class. In this project, the layers and the functions were used are convolutional layer, flatten, dense, activation function and fully connected layers.

1. Convolutional layer: A convolutional layer includes a series of clarifiers that needs to be tutored about the parameters. The filters height and weight are minor than the size of the input[8]. To cipher an activation map made of neurons, each clarifier is altered with the volume of inputs. In computer vision, one of the difficulties is that images can be really large and therefore computationally expensive to work on. For practical uses, we need quicker and computationally cheaper algorithms. A simple neural network that's fully connected won't help.
2. Flatten: Flatten is used to flatten the input. Kernel uses flatten parameter to apply flatten layer. Flattening occurs when a reduction of all layers to one background layer is demanded. Layers will increase the size of trains, while also tying up available resources for processing. Anyone can choose to combine some layers or indeed flatten the whole image to one background layer to keep down the train size.
3. Dense: A regular deeply connected neural network layer is a thick layer. The thick layer performs the input operation below and returns the output. The thick layer is also known as a completely connected layer (FC). Completely connected or thick layers are those layers in a neural network where all inputs from one layer are connected to each activation unit of the coming layer. The last many layers are completely connected layers in utmost common machine learning models that collect the data uprooted from former layers to form the final output.
4. Activation function: In neural networks, a node's activation function determines the node's output given an input or set of inputs. Depending on the input, activation functions can be' ON'(1) or' OFF'(0). One direct function enables each layer. In turn, this activation goes as an input into the coming step and the alternate layer calculates the weighted sum on that input and, in turn, fires grounded on another point of direct activation. There some common activation functions similar as ReLU, Tanh, and Sigmoid. Thick layer can also be used as activation function but since the MNIST figure classification is a non-linear process, we need another activation function.

ReLU activation function works by allowing only positive inputs to pass through unchanged and blocked any others. The exact structure of ReLU function is described in Equation (1).

$$\text{Relu}(x) = \max(0, x) \qquad \text{... (1)}$$

A common activation function is the sigmoid activation function and the exact expression in shown in Equation (2)

$$\text{sigmoid}(x) = 1/(1-e^{-x}) \qquad \text{... (2)}$$

## 4.0 Proposed Method

First dataset is loaded and various (RF, SVM, KNN) statistical classifiers are used and cconsequently performances of different algorithms are calculated. The MNIST dataset for hand-written figures is fed into neural networks. A heap of CNN-ReLU Maxpooling model is established to test the network for the same dataset. It's a combination of a 3-layer 2D convolutional layer. It has a Max Pooling 2D layer and a dropout layer. A dataset of 28×28 grayscale image is fed into the network. In connected network, Adam is used as an optimizer. The loss function of this classifier is set as categorical cross entropy. The model was trained with 10 periods primarily. In this network, the training process is a bit slower than the Network I. It takes a standard of 4 seconds per time. In these network settings, the results of a training accuracy were 92.59% and a test accuracy of 92.8%. A modification of the Network II was done to get better performance the parameters were changed similar to batch size, kernel size, pool size, clarifiers, and dropout. Also, the optimizer SGD (stochastic gradient descent) (SGD Classifier) was changed to Adam. After several trial and errors, results an accuracy of 99.82% during the training set and 99.25% for the testing set. Though it took near about 50 seconds per time during the training sessions, optimizer 'Adam' gave us a good result according to testing.

## 5.0 Results and Discussion

For testing the performance of different algorithms, the same dataset was trained with different network settings. For the first network, the model was trained using 10 times originally. The confusion matrix for the network is presented in Fig.5 for both training and testing. From the confusion matrix, we can fluently understand how the classifier predicts. We can also see there

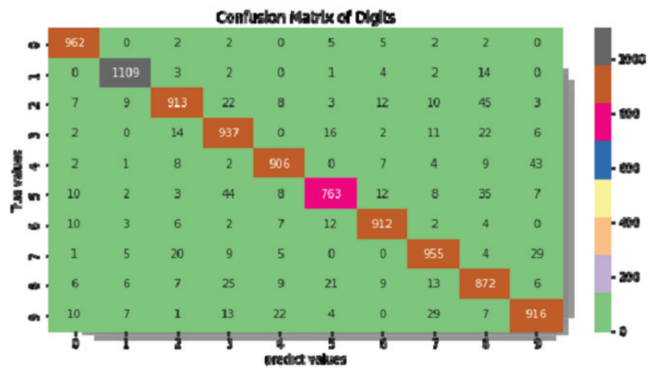

Figure 5: Confusion Matrix for the network

are some miss-classification while predicting.

We also tried stochastic gradient descent (SGD), which didn't perform well comparing to the Adam optimizer

After the training (60000 images included) and testing 10000 images in it, implementation time, precision, recall and F1 scores were also considered to examine the time complexity of the models. In Table 1, our best model proved to be RF classifier with the accuracy topmost training score of 100 and test score of 97 and Naïve Bayes have the lowest training score of 56 and test score of 55. We've witnessed that Random Forest Classifier have the topmost accuracy in all of used models.

In Table 2 we have seen that the CNN model (activation function-Sigmoid) provides a really stable accuracy of 92.54%. The ANN model (activation functions - Relu, Tanh, and Sigmoid) also provides an actually stable 92.80% accuracy in 5 times which gives the topmost accurate model in CNN.

**Table 1: Accuracy of Machine Learning Models**

| Accuracy of Machine Learning Models | | |
|---|---|---|
| Random Forest | 1.0 | 0.9704 |
| KNN | 0.9763333333333334 | 0.9659 |
| SVM | 0.9796666666666667 | 0.931 |
| Naïve Bayes | 0.5649 | 0.5558 |

**Table 2: The Results of CNN Models**

| Models | Loss | Accuracy of test data | Accuracy of train data |
|---|---|---|---|
| Sigmoid | 0.9259 | 0.2676 | 0.9245 |
| RELU,Sigmoid | 0.9305 | 0.2653 | 0.9254 |
| RELU,Tanh,Sigmoid | 0.9286 | 0.2638 | 0.9280 |

## 6.0 Conclusion and Future Scope

This paper mainly focused on measuring accuracy of various classifiers like Random Forest, SVM, KNN and Naïve Bayes while performing classification of handwritten digits on MNIST Dataset. A clear confusion matrix is drawn to compare the result which shows true predicted values for each digit after classification. Deep learning model CNN is also used for classification of hand written numbers. Adam and SGD optimizers are used for optimization part where Adam shows better performance compared to SGD. Different activation functions (RELU, TANH, SIGMOID) are applied individually and in combined approach to measure the performance of CNN[11]. The results of this present study revealed that out of the four models(Random Forest, KNN[12], SVM and Naïve Bayes) applied on our dataset, Random Forest Classifier is the best performer showing 100% accuracy in classification result. In future different architectures for DL[13] approach will be applied on that specific dataset to check performance accuracy as CNN method through ML has not achieved a good accuracy for this dataset.

## 7.0 References

1. Y. LeCun. "The MNIST database of handwritten digits." http://yann.lecun.com/exdb/mnist/ (accessed on 22nd june,22)
2. Image Classification using Feedforward Neural Network in Keras. (2017, October 23) (accessed on 23rd june, 2022).
3. E. Kussul and T. Baidyk, (2004): "Improved method of handwritten digit recognition tested on MNIST database," *Image and Vision Computing*, vol. 22, pp. 971-981, 10/01 2004, doi:10.1016/j.imavis.2004.03.008.
4. B. R. Atienza, "Advanced Deep Learning with Tensor Flow 2 and Keras - Second Edition," second edition ed: Packt Publishing Limited.
5. https://en.wikipedia.org/wiki/Random_forest (accessed on 22nd dec, 2022)
6. Deng, L., (2012): The MNIST database of handwritten digit images for machine learning research best of the web. *IEEE Signal Processing Magazine,* 29(6), pp.141-142.
7. https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning
8. Gedeon, T.D. and Harris, D., (1992): June. Progressive image compression. In Neural Networks, 1992. IJCNN, International Joint Conference on (Vol. 4, pp. 403-407). IEEE.
9. https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm
10. Rui Wang, Wei Li, Runnan Qin and JinZhong Wu (2017): "Blur Image Classification based on Deep Learning", IEEE, ISBN 978-1-5386-1621-5 pp. 1-6.
11. Harrison, O. (2019, July 14): Machine Learning Basics with the K-Nearest Neighbors Algorithm. Retrieved October 1, 2019, from https://towardsdatascience.com/machine-learning-basics-with-the- knearest-neighbors-algorithm-6a6e71d01761.
12. Goswami, A. (2018, August 9): Intro to image classification with KNN. Retrieved September 28, 2019, from https://medium.com/@YearsOfNoLight/intro-to-image-classificationwith-knn-987bc112f0c2.
13. Alzubaidi, L., Zhang, J., Humaidi, A.J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M.A., Al-Amidie, M. and Farhan, L., 2021. Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *Journal of big Data*, 8(1), pp.1-74.