Digit Recognition Using Supervised Learning Algorithms

Mehboob Ali
Computer Science
National University of Emerging and Computer Sciences
Lahore, Pakistan
l174316@lhr.nu.edu.pk

Hammad Hassan
Computer Science
National University of Emerging and Computer Sciences
Lahore, Pakistan
l174294@lhr.nu.edu.pk

Arsal Azhar
Computer Science
National University of Emerging and Computer Sciences
Lahore, Pakistan
l174314@lhr.nu.edu.pk

***Abstract*— Interpreting different forms of data through a computer requires different methods. Digit Recognition Challenge is the process where the digits are recognized by the computer. There are multiple methods and each has its pros and cons. In this study, we discuss K-NN, Feed-Forward Neural Network, and SVM through Histogram of Oriented Gradients (HOG). The computer is provided with an MNIST dataset and the accuracy of various models is checked. Each pixel on a picture is considered as a feature. This feature is processed by each of the three algorithms. The quickest approach is Feed-Forward Neural Network. It took 2-3 minutes to train and test the complete MNIST dataset. However, we observed that the most accurate approach is SVM through HOG which gave an accuracy of 98.23%.**

Keywords—KNN, SVM, HOG, FFNN, MNIST

I. Introduction

Digit Character Recognition is an important concept in Artificial Intelligence and Computer Vision. The data of handwritten digits is fed to the computer and the required output is the digit that was given as the data. The problem with digit recognition arises because a digit can have various styles. Handwritten characters can have different flavors which is why it is difficult for a computer to understand unless the right approach is used. It is an important benchmark in the field of computer vision. It reduced the need of retyping and allowed quicker text-editing, digital searches, and saved physical storage [1]. Moreover, countless recognition methods emerged from it. Following approaches are used in our research to complete the digit recognition challenge:

- K Nearest Neighbors (KNN) is an important concept in Artificial intelligence. It deals with problems related to classification and regression.
- Feed-Forward Neural Network is a concept derived from the actual working of neurons. It has layers of nodes which are called perceptrons. Each perceptron is given an input, in this case, in the form of a single pixel. This pixel is processed and then provided to the layer in succession. This layered model where input flows from input function x to intermediate layer and finally to output y is the reason why it is known as Feed-Forward [2]. This does not use a cyclic approach and data finally reaches the output layer.
- The third approach used in this study is SVM through Histogram of Oriented Gradients (HOG). Each image fed to the computer has several features. Processing each feature takes a considerable amount of time. HOG tries to reduce the features by getting the value of the gradient of multiple blocks of pixels in a picture. This list of features is then fed to SVM which supports Vector Machine Processing. The data is classified and conceptualized through 3-dimensional partition rather than a 2-dimensional partition. The time required to get the results by SVM through HOG is higher than the other two methods but it provides more accurate results.

II. Methodology

We have used three different supervised learning algorithms for the prediction of hand-written digits. These include: K-Nearest neighbors, Feed-Norward Neural Network and SVM with HOG features. Since this is a comparative study, the aforementioned classifiers will be discussed in detail in the following subsections.

A. K-Nearest Neighbors

KNN (K Nearest Neighbors) is a popular supervised machine learning algorithm used in artificial intelligence. Both classification and regression problems can be solved very simply using KNN. The process of classification using this algorithm is very easy to understand but its overall computation cost is pretty high. Hence, the results obtained are good but it takes a lot of time to train and test it. The value of K plays an essential role in it. A poorly chosen value for K can significantly deteriorate the overall accuracy. A depiction of how a target label is predicted by the classifier is shown in Fig. 1. The target label is predicted by finding a neighboring class which is the nearest. The distance is calculated using different formulas like the Euclidean distance and the Manhattan distance.

The value of K is external to the dataset, i.e. it cannot be evaluated using the data we have. Hence, the best way to find a good value for K is to perform input validation on the data. A portion of the training data is chosen for the input validation and different values of K are tested on this data. The value of K that produces the best accuracy is chosen for the training process.

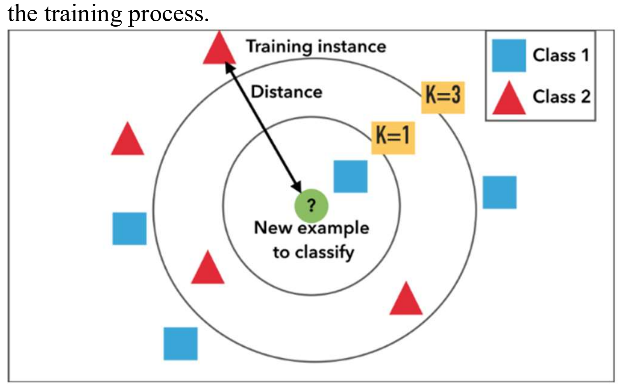

Figure 1: An intuitive representation of how a test distance is classified using K-NN. Source: Adapted from [3].

As the name suggests, KNN uses the distance of a new value from K nearest values (on which the algorithm is already trained), to find out its class. The class whose values are more abundant in the K nearest neighbors is assigned to the new value. Obviously, the cost of calculating the value for the nearest neighbors is very high. Especially when the size of the training dataset is large, it can take a lot of time to find the nearest neighbors.

B. Feed-Forward Neural Network

A feedforward neural network is a neural network in which the perceptrons (or neurons) are placed in different layers. It was the first neural network to be devised and is the simplest of all neural networks. A prominent property of this neural network is that it has no cycles between the nodes. A simple architecture of the feed-forward neural network is shown in Fig. 2. There are three major layers in a feed-forward neural network: Input layer, Hidden layer, and output layer. The input layer is the first layer and consists of all the input features, where each feature is represented as an individual neuron. There can be multiple layers in the hidden layer but the simplest form of this neural network has only one layer. The output layer is the last layer and has the output of the neural network. The number of neurons in the output layer is equal to the total number of classes that the neural network is classifying.

In a feed-forward neural network, each layer passes or feeds the output to the next layer. The input layer has all the features and passes them to the hidden layer. Then the hidden layer passes its output to the output layer. The output layer tells which class the object belongs to.

The working of a feed-forward neural network depends a lot on the activation function used. An activation function activates (or assigns output) for a neuron based on the input weights. The most common activation function is a sigmoid function as it can easily classify multiple classes. Another efficient and popular function is the relu function. It is also very frequently used in most of the neural networks.

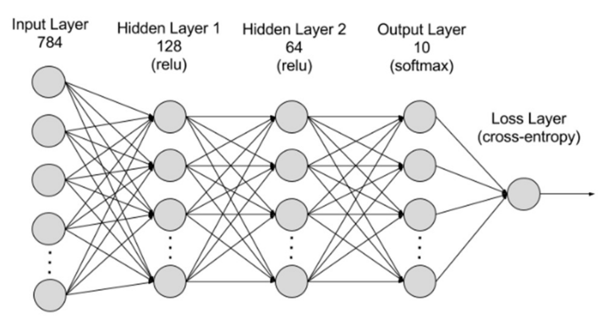

Figure 2: A simple Feed-Forward Neural Network with 2 hidden layers. Source: Adapted from [4].

C. SVM with Histogram of Oriented Gradients (HOG)

The histogram of oriented gradients (HOG) is used in computer vision and image processing as a feature descriptor. It is used for object detection. Gradient orientations are counted by this technique in a localized portion of the image. Using HOG, the process of classification is significantly enhanced.

Support Vector Machine (SVM) is a very successful and efficient supervised learning model in machine learning. It is a method based on statistical learning [5]. It converts the representation of the training data space to a higher dimensional space and creates a hyperplane that can be used to classify the objects. New examples are then mapped into this higher-order space and classified accordingly [6].

Using HOG to extract the features from the image before using the SVM classification model is very helpful. The whole image is first divided into different blocks and the HOG features can be extracted from these blocks of pixels. Once these features are extracted, they can be fed to SVM for the classification. Using HOG before SVM also increases the speed of the algorithm, as the number of features for the SVM algorithm is significantly reduced.

III. DATASET

We will be using MNIST Handwritten Digits dataset to train, test and evaluate the three classifiers. The dataset contains around 70000 samples of handwritten digits. Each sample consists of a grayscale image of size 28x28 (784 total pixels). The MNIST dataset has been partitioned into two parts. 60,000 images are used for training and the remaining 10,000 images are used for testing purposes. Different images of each digit from the MNIST dataset is shown in Fig. 3.

Figure 3: Samples images from the MNIST dataset. Source: Adapted from [7].

We calculated the number of distinct images for each digit. There are approximately 7000 sample images for each digit in the MNIST dataset as shown in Table. 1.

Table 1: Count of Samples for each digit in the MNIST dataset

Digit	Count
0	6903
1	7877
2	6990
3	7141
4	6824
5	6313
6	6876
7	7293
8	6825
9	6958

IV. EVALUATION

All the three classifiers were trained, tested and evaluated using the MNIST Handwritten digits dataset described in the previous section. This section describes the evaluations of the three proposed classifiers.

A. K-Nearest Neighbors

For the K-Nearest Neighbors classifier, the training dataset containing 60,000 handwritten images was split on a 75:25 ratio initially. The 75% is used for training purposes and the remaining 25% is used for testing. The split is done to prevent overfitting of the model. The 75% training dataset is further split on a 65:10 ratio. The 10% is used for tuning the value of "k" as it is a hyper-parameter. The exact values for training, testing and evaluation are shown in Table. 2.

Table 2: Data Split representation

Training Data	Testing Data	Evaluation Data
40500	15000	4500

To determine the best value of "k", the model was trained using different values of "k" ranging from 1 to 8. After training, the model was tested using the evaluation data. The best value of "k" turned out to be 3 as it has the highest accuracy as shown in Table. 3. The model was then trained using the best determined value of "k". It was then tested on the test dataset prepared by MNIST. It gave an accuracy of 96.79% on the unseen dataset which is quite encouraging.

Table 3: Model accuracy with different k values

K value	Accuracy (%)
1	96.87
2	96.51
3	97.02
4	96.76
5	97.02
6	96.62
7	96.71

B. Feed-Forward Neural Network

For the Feed-Forward Neural Network, slightly different accuracies were obtained by varying the number of hidden layers, number of neurons in the hidden layers, number of epochs and usage of different activation functions like relu and sigmoid. The training dataset of 60,000 samples was split on 80:20 ratio. 80% of the dataset was used for training and the remaining 20% was used for testing. The exact values for training, testing and evaluation are shown in Table. 4. After that, the model was tested on unseen MNIST test dataset comprising of 10,000 samples.

Table 4: Data split experimental setting for Feed-Forward Neural Network

Training Data	Testing Data
48000	12000

There were few experimental settings which were kept same throughout the evaluation process. These include: number of neurons in the input layer were kept to 784, number of neurons in the output layer were kept to 10, the activation function used for the output layer was "softmax" and the optimizer used was "adam". The accuracies obtained after varying the parameters are shown in Table. 5. The accuracy column in the table is based on portion of training data which was partitioned for testing.

Table 5: Accuracies with different set of parameters

Activation function	Number of hidden layers, # neurons	Training Data	Testing Data	Accuracy (%)
relu	1, 784	48000	12000	97.55
relu	1, 784	42000	18000	97.87
relu	2, 784, 512	48000	12000	97.75
relu	2, 784, 512	42000	18000	97.60
relu	3, 784, 512, 256	48000	12000	98
relu	3, 784, 512, 256	42000	18000	97.95
Sigmoid	1, 784	48000	12000	97.85
Sigmoid	1, 784	42000	18000	97.85
Sigmoid	2, 784, 512	48000	12000	98
Sigmoid	2, 784, 512	42000	18000	
Sigmoid	3, 784, 512, 256	48000	12000	97.99
Sigmoid	3, 784, 512, 256	42000	18000	97.62

C. SVM with HOG features

Linear and non-linear versions of the Support Vector Machine (SVM) classifier were used with HOG. For the non-linear versions, two different kernels were used on default settings, e.g. poly kernel and rbf kernel. The MNIST training dataset of 60,000 samples was split on 80:20 ratio as well.

The HOG features vector is calculated for all the training samples. The vector is then passed to the SVM classifier for training purposes. HOG features are extracted as well for testing the model on unseen data instead of using raw test images. The parameters set for calculating HOG features are shown in Table. 6.

Table 6: Experimental setting for calculating HOG features from images

Number of orientation	Pixels per cell	Cells per block
4	4	4

The SVM classifier with different kernels was training, tested and evaluated with HOG features as input vector instead of actual raw images. The accuracy of different kernels are shown in Table. 7.

Table 7: Accuracies obtained using different SVM kernels

Kernel	HOG features used	Accuracy (%)	# of training samples	# of testing samples
Linear	Yes	11.1	12000	48000
Linear	No	91.57	12000	48000
Non-linear (poly)	Yes	98.38	12000	48000

Non-linear (rbf)	Yes	98.23	12000	48000

V. DISCUSSION

Three different digit recognition algorithms were used for training and testing the data on the MNIST dataset of handwritten digits. For each algorithm, the dataset was divided into a training set and a testing set. The training set for Feed Forward Neural Network was 80% and the testing set was 20% of the original data. For SVM using HOG and KNN 75% of the data was used for testing and 25% for testing. Different ratios were tested before using these ratios and it was found that this yielded the best results.

KNN was used with input validation for finding the ideal value of K. 10% of the data in the training set was used for the validation and it was found that K=3 yields the best accuracy. Once the value of K is identified, the algorithm maps the training dataset. After the training process, each value in the testing set is classified by finding its K nearest neighbors. Even though the process is very costly and takes a lot of time, the classification is very accurate.

Feed-Forward Neural Network was tested thoroughly using a different number of neurons for each test. There are 784 neurons in the input layer and 10 neurons in the output layer. It was found that using 784 neurons in the hidden layer gives the best result. It was tested using both sigmoid function and relu function as the activation function. It was found that the sigmoid function works better.

The SVM model was implemented using HOG for extracting the features. In this approach, we divide the input image into a 9 x 9 grid and calculate oriented gradients for each of these blocks of pixels. These gradients are used as the features for the SVM algorithm. As the image is divided into a 9 x 9 grid, there are a total of 81 features that are fed to SVM. Then the SVM maps the features obtained through HOG onto a higher-dimensional space. Whenever a new value appears, it is mapped on the same plane and classified accordingly.

VI. RESULTS AND COMPARISON

Each model provided a different result. The accuracy and the time difference is evident. After initial training and testing on the 60,000 sample training dataset of MNIST, the models were tested on the unseen 10,000 MNIST testing dataset. 97.83% accuracy was achieved using relu as activation function and 97.88% was achieved using sigmoid as activation function with Feed-Forward Neural Network which is the lowest out of all three. However, it was the fastest approach as the result was provided in 2-3 minutes. The accuracy achieved with K-NN was 97.97% and it took 2 hours to provide the answer. Finally, the most efficient approach was SVM through HOG. The accuracy achieved with it was 98.38%. K-NN took the most time as it took over 2 hours for it to process the data. The reason why KNN is the slowest out of the three methods is that it requires finding the value for K using input validation and then finds K nearest neighbors, which is a very costly process. The results of the three classifiers are shown in Table. 8.

Table 8: Accuracy and time consumption of three different classifiers

Classifier	Accuracy (%)	Time (minutes)
K-NN (k = 3)	97.97	120
FFNN (sigmoid activation function)	97.88	2-3
SVM (non-linear poly kernel)	98.38	90

VII. CONCLUSION

Three different supervised learning algorithms were discussed in this paper to solve the famous handwritten digit recognition problem. K-Nearest Neighbors classifier gave us an accuracy of about 97.97% on unseen data but the computational time was exceptionally high. Support Vector Machine (SVM) classifier was used with HOG features. The non-linear poly SVM kernel gave us the highest prediction accuracy of 98.38%. It faces the same problem as K-NN, the computational is high for it as well. To overcome this problem, a simple Feed-Forward Neural Network was used as well. It predicted handwritten digits with an accuracy of 97.88% but within 2-3 minutes. It could be seen that the accuracy and computational time trade-off is quite significant in this scenario. Moreover, using Feed-Forward Neural Network, we were able to obtain Top 50% rank in the MNIST Digit Recognition competition.

REFERENCES

[1] Sunanda, Optical Recognition of Digital Characters Using Machine Learning. Dayananda Sagar College of Engineering, Bengaluru, India 2018.J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[2] Tushar Gupta, Deep Learning: Feedforward Neural Network. 2017.

[3] Medium. 2020. *K-Nearest Neighbours (KNN) Algorithm*. [online] Available at: <https://medium.com/@sonish.sivarajkumar/k-nearest-neighbours-knn-algorithm-9900c1427726> [Accessed 12 June 2020].

[4] Medium. 2020. *A Must Read Intro To Neural Networks Using Pytorch—Handwritten Digit Recognition*. [online] Available at: <https://towardsdatascience.com/handwritten-digit-mnist-pytorch-977b5338e627> [Accessed 12 June 2020].

[5] Ying, Zhao, and Wan Fuyong. "An Algorithm to Incremental Learning with Support Vector Machine and Its Application in Multi-Class Classification." *2006 Chinese Control Conference*, 2006, doi:10.1109/chicc.2006.280868. M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

[6] Ebrahimzadeh, Reza, and Mahdi Jampour. "Efficient Handwritten Digit Recognition Based on Histogram of Oriented Gradients and SVM." *International Journal of Computer Applications*, vol. 104, no. 9, 2014, pp. 10–13., doi:10.5120/18229

[7] Medium. 2020. *A Must Read Intro To Neural Networks Using Pytorch—Handwritten Digit Recognition*. [online] Available at: <https://towardsdatascience.com/handwritten-digit-mnist-pytorch-977b5338e627> [Accessed 12 June 2020].