# Efficient Handwritten Digit Recognition based on kNN, FFNN and SVM using HOG Features


Hammad Maqsood
17L-4417
L174417@lhr.nu.edu.pk

M. Shazar
17L-4177
L174177@lhr.nu.edu.pk

Danish Ali
17L-6317
L176317@lhr.nu.edu.pk



***Abstract - Automatic Handwritten Digits Recognition (HDR) is the process of interpreting handwritten digits by machines. There are several approaches for handwritten digits recognition. In this paper, we have proposed an appearance feature-based approach which process data using Histogram of Gradient (HOG). Also, we use KKN and FFNN classifier to train our dataset to recognize handwritten dataset. HOG is a very efficient feature descriptor for handwritten digits which is stable on illumination variation because it is a gradient-based descriptor. KNN algorithms use data and classify new data points based on similarity measures. Whereas FFNN is also known as multi-layered network of neuron. We have analyzed our models on MNIST dataset and obtain 96.2 % accuracy in case of KKN, 94.7 % accuracy in case of FFNN and 98 % accuracy in the case of SVM using HOG.***



***Keywords - Handwritten Digit Recognition, Number Recognition, Character Recognition, HOG, SVM, FFNN, KNN***


## 1. INTRODUCTION

One of the very popular applications in computer vision is Handwritten Digits Recognition (HDR) in the field of character recognition. Digits like other universal symbols are widely used in technology, bank, OCR, analyzing of digits in engineering, postal service, numbers in plate recognition, etc. They are some of the famous applications on HDR. There are 10 classes corresponding to the handwritten digits from '0' to '9' which are very depend on the handwritten. The main difficulty in the handwritten digit's recognition is different handwritten style which is a very personal behavior where there are a lot of models for numbers based on the angles, length of the segments, stress on some parts of numbers, etc. **Figure 1** shows 15 different handwritten digits related to these issues. However, recognizing numbers is clear for human but it is not very easy for machines especially when there are some ambiguities on different classes (e.g. '1' and '7'). Recognizing digits is very important because it is related to the numbers thereby the recognition methods have to be very accurate. There are different kinds of HDR approaches reported by researchers.

**Fig. 1: Different samples of handwritten digits in MNIST**

## 2. DESCRIPTION OF OVERALL MODEL

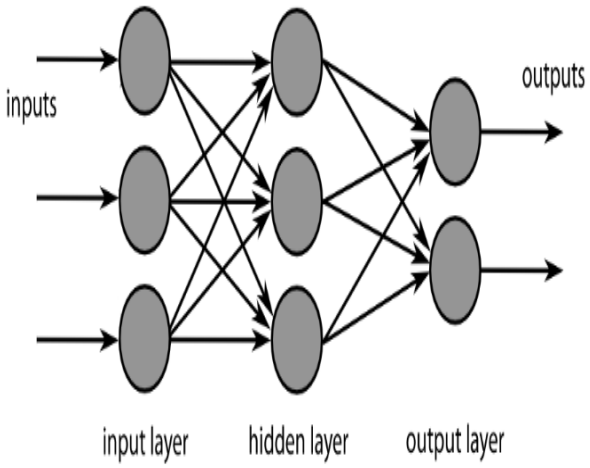
### 2.1 kNN:

In pattern recognition, the k-nearest neighbors' algorithm (k-NN) is a non-parametric method used for classification and regression. In both cases, the input consists of the k closest training examples in the feature space [1]. In k-NN classification, the output is a class membership. An object is classified by a plurality vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of that single nearest neighbor [2]. Different values of k were tried and tested to find the most accurate combination which came out to be k=3 for the given data on Kaggle. These values were tried using GridSerachCV and also by Cross Val Score. The accuracy for data tested in 96.3%. [3]

### 2.2 Feed Forward Neural Network:

In this pattern recognition, the feedforward neural network was the first and simplest type of artificial

neural network devised. In this network, the information moves in only one direction, forward, from the input nodes, through the hidden nodes (if any) and to the output nodes. There are no cycles or loops in the network. This class of networks consists of multiple layers of computational units, usually interconnected in a feed-forward way. Each neuron in one layer has directed connections to the neurons of the subsequent layer. In many applications the units of these networks apply a sigmoid function as an activation function. In this neural network, basically we make two hidden layer each have 64 neurons. Initially, we assign weight to each object randomly. Then, we train our neural network on MNIST dataset using sigmoid function as activation function. Loss function (**criterion**) decides how the output can be compared to a class, which determines how good or bad the neural network performs. And the **optimizer** chooses a way to update the weight in order to converge to find the best weights in this neural network. The accuracy we obtain from FFNN is approximately 95%. [4]

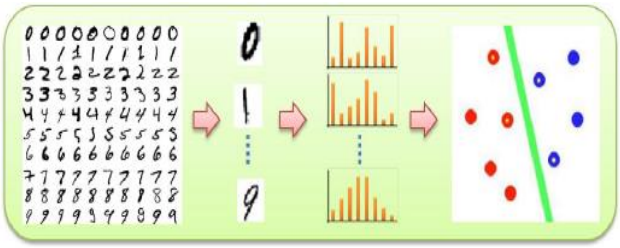



### 2.3 Support Vector Machine using HOG:

In machine learning, support-vector machines (SVMs, also support-vector networks) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. [5] Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier. Gradients of specific directions captures some notion of shape. To allow for some variability in shape, we'll use features known as Histogram of Oriented Gradients (HOG). [6]

The idea of HOG is instead of using each individual gradient direction of each individual pixel of an image, we group the pixels into small cells. For each cell, we compute all the gradient directions and group them into a number of orientation bins. We sum up the gradient magnitude in each sample. So stronger gradients contribute more weight to their bins, and effects of small random orientations due to noise is reduced. This histogram gives us a picture of the dominant orientation of that cell. Doing this for all cells gives us a representation of the structure of the image. The HOG features keep the representation of an object distinct but also allow for some variations in shape. [7]

An exhaustive search for different values of parameters was run for the algorithm and the best set of parameters gave an accuracy of 98%.



## 3. Database and Benchmark:

MNIST Handwritten Digits database of Kaggle has been used in our model validation. MNIST is one of the most famous and popular used databases for handwritten digits recognition which contains 70,000 samples included two parts of 60,000 and 10,000 samples corresponding to training and test data. The total numbers of instances are given below

| Class | Number of samples | | |
|---|---|---|---|
| | Train Data | Test Data | Total |
| '0' | 5923 | 980 | 6903 |
| '1' | 6742 | 1135 | 7877 |
| '2' | 5958 | 1032 | 6990 |
| '3' | 6131 | 1010 | 7141 |
| '4' | 5842 | 982 | 6824 |
| '5' | 5421 | 892 | 6313 |
| '6' | 5918 | 958 | 6876 |
| '7' | 6265 | 1028 | 7293 |
| '8' | 5851 | 974 | 6825 |
| '9' | 5949 | 1009 | 6958 |
| All | 60,000 | 10,000 | 70,000 |

**4. Result and Comparison:**

In this part we have proposed the results and comparison of proposed model like KNN, FFNN and SVM using HOG features. The accuracy we obtained in case of K nearest neighbor is 96.2%. The accuracy in case of feed forward neural network is 94.5% and in case of SVM using HOG features is 98%. All proposed model gives us almost accurate result as possible. Neural network is trained on 60,000 samples of MNIST digits and tested on 10,000 samples.

# References


[1] O. Surinta. [Online]. Available: https://medium.com/olarik/basic-character-recognition-tutorial-9f55ff030f1c.

[2] K. Wright. [Online]. Available: https://medium.com/district-data-labs/parameter-tuning-with-hyperopt-faa86acdfdce.

[3] "Data School's Machine Learning," [Online]. Available: https://www.ritchieng.com/machine-learning-efficiently-search-tuning-param/.

[4] C. Warke. [Online]. Available: https://medium.com/random-techpark/simple-feed-forward-neural-network-code-for-digital-handwritten-digit-recognition-a234955103d4.

[5] Mithi. [Online]. Available: https://medium.com/@mithi/vehicles-tracking-with-hog-and-linear-svm-c9f27eaf521a.

[6] dpetrini. [Online]. Available: https://github.com/dpetrini/mnist_compare.

[7] R. Gandhi. [Online]. Available: https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47.