

**CS401 Artificial Intelligence**

Programming Assignment No 4 (Sections E, F)

Spring 2020

Assigned on 21/04/2020

**HARD Deadline Part a)** 23/04/2020 on **slate and piazza** before 11:55 p.m.

[1 Mark]

**SOFT Deadline Part b- d)** 01/05/2018 on **Kaggle, slate and piazza** before 11:55 p.m.

[3 Marks]

**HARD Deadline Part e)** 05/05/2020 on **slate and piazza** before 11:55 p.m.

[2 Marks]

**Total Weight 6%****Digit Recognition Using Supervised Learning****Instructions:**

Following rules will be enforced for this assignment

- This assignment will be done independently or in groups of at most 2 students
- Recommended platform for this assignment is PYTHON 3.5 but you are free to choose a language of your choice like Java, MATLAB, and Octave etc.

---

**Introduction**

In this assignment we are going to participate in a digit recognition challenge available on Kaggle.

See <https://www.kaggle.com/c/digit-recognizer> for more details.Some of the test accuracies obtained for MINIST digit recognition dataset by using various ML based techniques are available at <http://yann.lecun.com/exdb/mnist/index.html>

The overall assignment has been divided in three steps consisting of

- i) Creating an account on Kaggle and creating a team (if you are working in group of two students).
- ii) Creating different classifiers for recognizing the handwritten digits **using existing libraries or implementations**. You are mainly required to use K-Nearest Neighbor classifier (**kNN**), Feed-Forward Neural Networks (**FFNN**), and an SVM based classifier along with HOG features to create a digit recognizer. Finally you must submit your entries, one for each of these classifiers, on Kaggle.
- iii) Create a detailed report explain the problem i.e. the digit recognition challenge, details of your methods, like the number of layers and number of neurons in each layer, details of your experimental settings and finally a comparison of test accuracies obtained using various techniques.

**CS401 Artificial Intelligence**

Programming Assignment No 4 (Sections E, F)

Spring 2020

Assigned on 21/04/2020

**HARD Deadline Part a)** 23/04/2020 on **slate and piazza** before 11:55 p.m.

[1 Mark]

**SOFT Deadline Part b- d)** 01/05/2018 on **Kaggle, slate and piazza** before 11:55 p.m.

[3 Marks]

**HARD Deadline Part e)** 05/05/2020 on **slate and piazza** before 11:55 p.m.

[2 Marks]

**Total Weight 6%****Grading****Part a [1 Mark]**

Create an account on Kaggle and team up. Please use an **exotic** name for your team.

You must submit, on Slate and Piazza, a document containing the details of your account information and team information on Kaggle. Deadline is 23<sup>rd</sup> of April before 11:55 pm.

**Part b) Using K-NN classifier to predict labels of an input image.**

Your first job in this assignment is to compare performance of K-Nearest Neighbors classifier for the MNIST digit recognition dataset. Remember that you are allowed to use existing implementation in Python or any other language. Finally submit your entry on Kaggle using your KNN classifier.

Please remember that the best value of K can be chosen empirically or using validation data (Read about validation data and hyper-parameter tuning online. The K in KNN is a hyper-parameter)

**Part c)**

Use an existing method of creating a Feed-Forward Neural Network and use it to create a network for recognizing digits. Try various choices for the number of hidden layers and number of neurons in each layer and submit the best results you can obtain using Neural Networks.

**Part d)**

Use SVM along with Histogram of Oriented Gradients (HOG features) as descriptors/features to classify the digits. (You can use any available implementation of the HOG based digit recognition using SVM)

**Part e)**

Create a detailed report (in the format of a research paper) explaining the problem i.e. the digit recognition challenge, details of your methods, like the number of layers and number of neurons in each layer, details of your experimental settings (amount of data used for training, testing etc.) and finally a comparison of test accuracies obtained using various techniques.

**A Sample Paper Format is given on the next pages.**

# Efficient Handwritten Digit Recognition based on Histogram of Oriented Gradients and SVM

Reza Ebrahimzadeh  
Islamic Azad University of Zahedan branch  
Zahedan, Iran

Mahdi Jampour  
Graz University of Technology  
Graz, Austria

## 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 Oriented Gradients (HOG). HOG is a very efficient feature descriptor for handwritten digits which is stable on illumination variation because it is a gradient-based descriptor. Moreover, linear SVM has been employed as classifier which has better responses than polynomial, RBF and sigmoid kernels. We have analyzed our model on MNIST dataset and 97.25% accuracy rate has been achieved which is comparable with the state of the art.

## General Terms

Image Processing, Computer Vision, Artificial Intelligence

## Keywords

Handwritten Digit Recognition, Number Recognition, Character Recognition, HOG, SVM.

## 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 [1]. 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 digits 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: Saxena et al. [2] proposed a neural network model for classification of handwritten digits; they enhanced their methods using ensemble classification. Das et al. [3] selected local features in handwritten digits using genetic algorithm and then classified features with SVM.

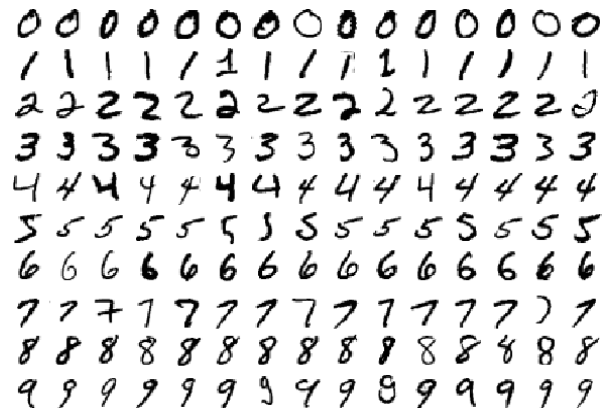


Fig. 1: Different samples of handwritten digits in MNIST

Cardoso and Wichert proposed a biologically inspired model for HDR [4] they also used a linear SVM. A hybrid model is proposed by Niu and Suen [16] where integrating the synergy of two superior classifiers: Convolutional Neural Network (CNN) and Support Vector Machine (SVM) that was the main contribution of authors for improving handwritten digit recognition.

## 2. DESCRIPTION ON OVERALL MODEL

Our model works in three steps: 1) Preprocessing, 2) HOG features extraction and 3) Support vector machines classification. In the preprocessing, we have some basic image processing to separate numbers from real samples or preparing data from dataset (which is reshaped from images to the vectors) and then in the second part, we extract HOG features which is very distinguishable descriptor for digits recognition where we divide an input image into 9x9 cells and compute then the histogram of gradient orientations thereby we represent each digit with a vector of 81 features. And finally in the third stage, a linear multiclass support vector machine has been employed to classify digits. The overall view of proposed approach has been illustrated in Figure 2. In general, the main contribution of our model is employing HOG features with SVM. HOG is a fast and reliable descriptor which can performs distinguishable features. And SVM is also a fast and powerful classifier that can be useful to classify HOG features. The most important benefit of above structure is that our model is fast and useful for real-time applications. The details are described in the following sections. We first describe HOG descriptor in section 2.1 and reintroduce SVM in section 2.2, evaluated dataset and validation is described in section 2.3, results and comparison discusses in section 3 and finally we described our model robustness in section 4.

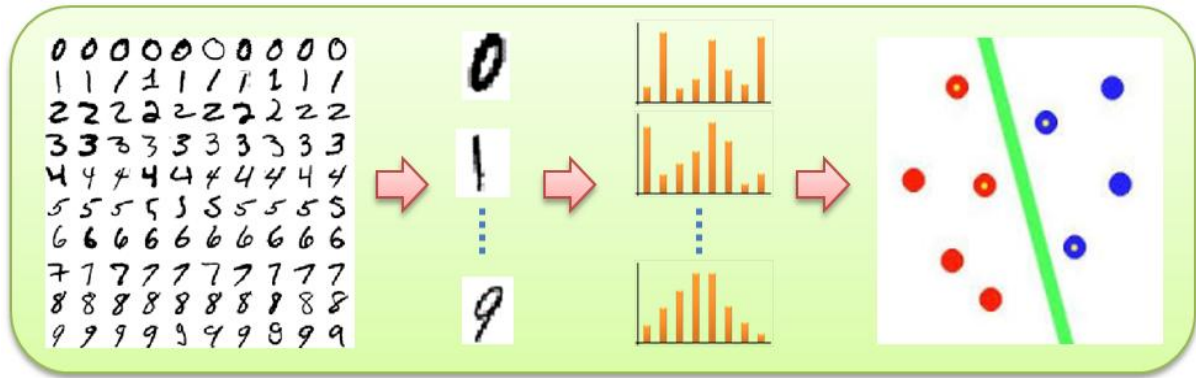


Fig. 2: The overall structure of our model, we first separate digits and then extract HOG features, finally, features are classified by multiclass SVM classification

## 2.1 Histogram of Oriented Gradients

Histogram of Oriented Gradient (HOG) was first proposed by Dalal and Triggs [5] for human body detection but it is now one of the successful and popular used descriptors in computer vision and pattern recognition. HOG counts occurrences of gradient orientation in part of an image hence it is an appearance descriptor. HOG divides the input image into small square cells (here we used  $9 \times 9$ ) and then computes the histogram of gradient directions or edge directions based on the central differences. For improve accuracy, the local histograms have been normalized based on the contrast and this is the reason that HOG is stable on illumination variation. It is a fast descriptor in compare to the SIFT and LBP due to the simple computations, it has been also shown that HOG features are successful descriptor for detection. HOG features on several numbers have been illustrated in figure 3.

## 2-2- Classification

Without any doubt, SVM is one of the successful and most popular supervised learning classifier in machine learning which constructs a hyper-plane in high order space which can be used as classification plane. SVM commonly used with linear, polynomial, RBF and sigmoid kernels. We have used a multiclass SVM classification (libsvm) [6] in our model with different kernels of 1) linear, 2) polynomial, 3) RBF, 4) sigmoid, and details are described in section 3 (results and comparison). As a standard validation, we separated data into two parts of Train and Test according to the baseline (see Table 1). Moreover, different kernels are employed to evaluate our model which is described in section 3.

## 2-3- Database and benchmark

MNIST Handwritten Digits database of New York University [7] has been used in our model validation. MNIST is one of the most famous and popular used database 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 shown in Table 1 ordered by classes and some of the samples are shown in figure 1.

Table 1: MNIST Handwritten Digits test and train info

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

## 3- RESULTS AND COMPARISON

In this part we have proposed the results and comparison of proposed model using HOG features and linear SVM. We have achieved 97.25% accuracy rate which is comparable with the state of the art where K-nearest neighbor [8] has 91.6% accuracy, K-means clustered modified k-nearest classifier (KMA-MKNN) [8] 93.64%, Relevance vector machine [9] with 94.9% accuracy, Convolutional network [10] reported 95.8% accuracy, Incremental tensor principal component analysis (ITPCA) [11] with 96.7% accuracy rate, Differential Chain Code Histogram (DCCH) 95.32% [12] Support vector machine [13] 96% accuracy, Invariant support vectors [14] 97% accuracy and Transformation distance [15] reported 97.5% accuracy. We also analyzed polynomial, RBF and sigmoid kernels that there was not any improvement than linear SVM; their accuracy has been shown in table 2.

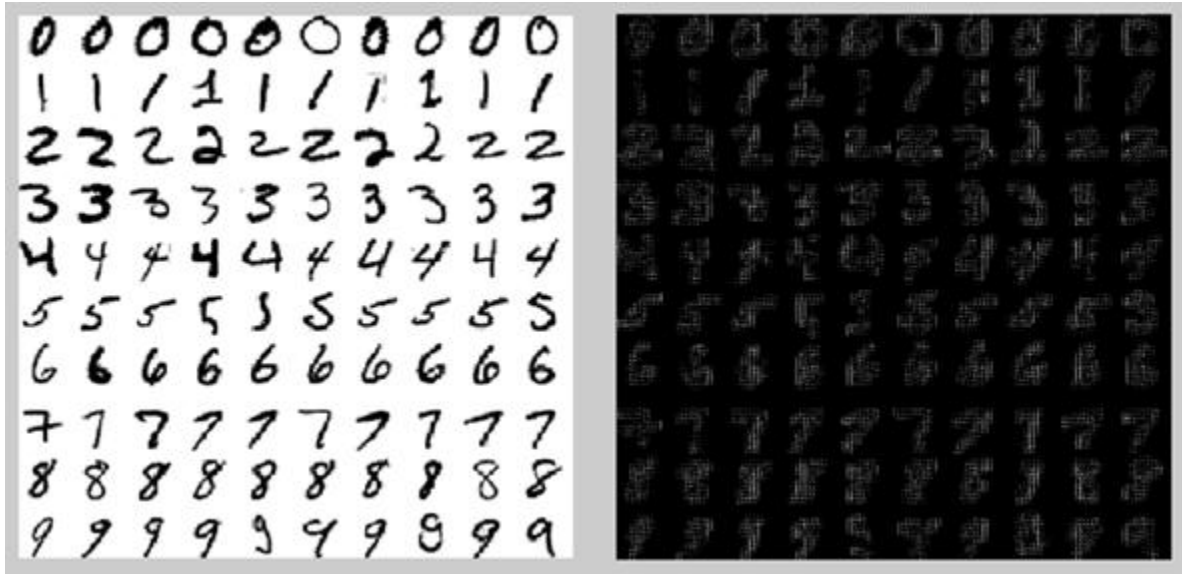


Fig. 3: HOG features for some handwritten numbers (zoom in to see details)

Table 2: Comparison between different kernels of SVM

SVM Kernels	Linear	Polynomial	RBF	Sigmoid
Accuracy Rate	97.25	85.91	95.75	94.97

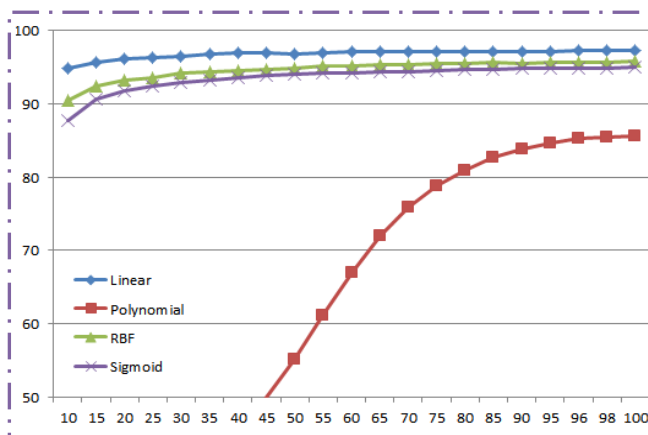


Fig. 4: Impact of training number of data on the accuracy rate (both of axes are based on percent), 100% of training data means 60,000 samples

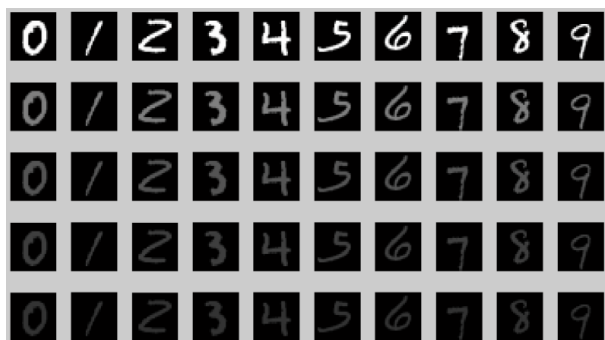
#### 4- ROBUSTNESS AND DISCUSSION

We analyzed affection of different size of training data on our approach with different kernels where we assumed 5% to 100% of training data in each trial.

We assumed 5% to 100% of training data in each trial. As shown in the figure 4, it is clear that our approach with linear SVM is stable on decreasing a big part of training data which means we don't need a lot of training data to learn our model but other kernels are strongly depend on the size of training data. Moreover, Table 3, shows the confusion matrix on handwritten digits where it is notable that the most confusions happened between classes of '2', '3' and '2', '7' and '2', '8' which are make sense because upper section of the number of '2' is similar to '3' and '8' and also the middle section of '2' is similar to '7' and '8'. Further, as we expected, our proposed approach is strongly stable on illumination variation where we darken digit images by 5%, 10%, 20%, 30% and 40% of the intensity and analyzed our method with new data, the results was surprisingly 97.25%, 97.25%, 97.24%, 97.23% and 97.22%, respectively. Figure 5 demonstrates some illuminated samples. In general, proposed approach used HOG which is a very efficient appearance-based descriptor with linear multiclass SVM classification for handwritten digits recognition. Results show, our model is not only efficient and comparable with the state of the art but also it is strongly stable on illumination variation even up to 40%.

**Table3: Handwritten digits recognition confusion matrix on MNIST database**

CM	'0'	'1'	'2'	'3'	'4'	'5'	'6'	'7'	'8'	'9'
'0'	<b>99.59</b>	0	0.10	0.10	0	0	0.10	0.10	0.10	0
'1'	0	<b>99.29</b>	0.17	0.08	0.08	0	0.08	0.17	0.17	0
'2'	0.48	0.09	<b>97.76</b>	0.58	0	0	0.09	0.77	0.29	0
'3'	0.29	0.09	1.28	<b>96.53</b>	0	0.69	0	0.49	0.59	0.09
'4'	0	0	0	0	<b>97.65</b>	0	0.40	0.30	0.30	1.42
'5'	0.11	0	0	1.45	0	<b>97.75</b>	0.22	0	0.44	0.11
'6'	0.52	0.10	0.20	0	0.31	1.04	<b>97.49</b>	0	0.41	0
'7'	0	0.19	1.55	0.09	0.38	0	0	<b>96.49</b>	0.58	0.77
'8'	0.51	0.10	1.23	0.61	0.61	0.30	0.51	0.82	<b>94.45</b>	0.92
'9'	0.19	0.39	0.19	0.29	0.99	0.09	0	1.38	0.39	<b>96.13</b>



**Fig. 5: Different illuminated samples are used for analyzing proposed approach**

## 5- REFERENCES

- [1] Hu D, Research and application of handwritten numeral recognition method, SM thesis, University of Nanchang, Nanchang, China. 2012
- [2] Neera Saxena, Qasima Abbas Kazmi, Chandra Pal and O.P. Vyas, Employing Neocognitron Neural Network Base Ensemble Classifiers To Enhance Efficiency of Classification In Handwritten Digit Datasets. D.C. Wyld, et al. (Eds): CCSEA 2011, CS & IT 02, pp. 408–416, 2011.
- [3] Nibaran Das, Ram Sarkar, Subhadip Basu, Mahantapas Kundu, Mita Nasipuri, Dipak Kumar Basu: A genetic algorithm based region sampling for selection of local features in handwritten digit recognition application. Appl. Soft Comput. (ASC) 12(5):1592-1606 (2012)
- [4] Ângelo Cardoso, Andreas Wichert: Handwritten digit recognition using biologically inspired features. Neurocomputing (IJON) 99:575-580 (2013)
- [5] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In CVPR, 2005
- [6] C.-C. Chang and C.-J. Lin. Libsvm: A library for support vector machines. ACM T. Intell. Syst. Technol., 2(3):1–27, 2011
- [7] MNIST Handwritten Digits database of New York University <http://www.cs.nyu.edu/~roweis/data.html>
- [8] M. Narasimha Murty, V. Susheela Devi. An Application: Handwritten Digit Recognition, ISBN: 978-0-85729-494-4, Springer, 2011
- [9] Michael E. Tipping, The relevance vector machine, Advances in Neural Information Processing Systems, vol.12, The MIT Press, 2000, pp.652–658
- [10] P. Simard, Y. LeCun, J. Denker, B. Victorri, Transformation invariance 400 in pattern recognition, tangent distance and tangent propagation, Neural Networks: Tricks of the Trade, Springer, 1998.
- [11] Chang Liu, Tao Yan, WeiDong Zhao, et al., Incremental Tensor Principal Component Analysis for Handwritten Digit Recognition, Mathematical Problems in Engineering, vol. 2014, Article ID 819758, 10 pages, 2014
- [12] You Qian, Wang Xichang, Zhang Huaying, Sun Zhen, Liu Jiang, Recognition Method for Handwritten Digits Based on Improved Chain Code Histogram Feature, 3rd Int. Conf. Multimedia Technology, 2013
- [13] B. Scholkopf, C. Burges, V. Vapnik, Extracting support data for a given task, First International Conf. Knowledge Discovery & Data Mining, AAAI Press, Menlo Park, CA, 1995
- [14] B. Scholkopf, P. Simard, A. Smola, V. Vapnik, Prior knowledge in support vector kernels, Advances in Neural Information Processing Systems, vol. 10, MIT Press, 1998, pp.640–646.
- [15] P. Simard, Y. Le Cun, J. S. Denker, Efficient pattern recognition using a new transformation distance, in: Advances in Neural Information Processing Systems, vol.5, Morgan Kaufmann, 1993, pp.50–58
- [16] Xiao-Xiao Niu, Ching Y. Suen: A novel hybrid CNN-SVM classifier for recognizing handwritten digits. Pattern Recognition (PR) 45(4):1318-1325 (2012)