

Traffic Signs Recognition using HP and HOG Descriptors Combined to MLP and SVM Classifiers

A. Salhi, B. Minaoui, M. Fakir, H. Chakib, H. Grimech

Faculty of science and Technology
Sultan Moulay Slimane University
Beni Mellal, Morocco

Abstract—Detection and recognition of traffic signs in a video streams consist of two steps: the detection of signs in the road scene and the recognition of their type. We usually evaluate globally this process. This evaluated approach unfortunately does not allow to finely analyze the performance of each step. It is difficult to know what step needs to be improved to obtain a more efficient system. Our previous work focused on a real-time detection of road signs, by improving the performances of the detection step in real time. In this paper, we complete the work by focusing on recognition step, where we compare the performances between histogram projection (HP) descriptor, and the histogram-oriented gradient (HOG) descriptor combined with the Multi-Layer Perceptron (MLP) classifier, and the Support Vector Machine (SVM) classifier, to compute characteristics and descriptors of the objects extracted in the step of detection, and identify the kind of traffic signs. Experimental results present the performances of the four combinations of these methods “Descriptor-Classifer” to identify which of them could have high performance for traffic sign recognition.

Keywords—Traffic signs detection and recognition; Histogram of oriented gradient (HOG); Support Vector Machine (SVM); Histogram projection (HP); Multi-layer perceptron (MLP)

I. INTRODUCTION

As an important road safety facility, traffic signs allow to regulate road traffic, by indicating the conditions of the road, guiding the pedestrians, driving safely, etc. Therefore, research issue on traffic sign real time detection and recognition is very important for driving assistance. In recent years, traffic sign recognition becomes an important research direction due to the rapid development of intelligent transportation system. Generally, works in this issue like [4]-[6] adopt a two-step approach, a detection step and a recognition step. These works evaluate globally the process; this unfortunately does not allow analyzing the performance of each step separately.

Our objective is to analyze each step separately, to stop on its weaknesses and propose solutions to correct and improve the performance of each one, and subsequently improve the performance of the overall system.

In our works [1]-[3], we focused on the step of detection of traffic signs, by evaluating their performances on a set of images of traffic scene (offline mode), and in real-time by using a camera (online mode). In this paper we continue the work by focusing on the step recognition of the traffic signs to evaluate and improve its performances.

The work presented in this paper, focuses on recognition step, where we design and implement an efficient method to identify traffic signs. We compare the performances between histogram projection descriptor, and the histogram-oriented gradient descriptor combined with Multi-Layer Perceptron (MLP) classifier, and the Support Vector Machine (SVM) classifier, to compute characteristics and descriptors of the objects extracted in the step of detection, and identify the kind of traffic signs. Experimental results present the performances of the four combinations of these methods “Descriptor-Classifer” to identify which of them could be more efficient for traffic sign recognition.

II. HISTOGRAM PROJECTION TECHNIC AND HISTOGRAM-ORIENTED GRADIENT TECHNIC

A. Histogram Projection (HP)

Blobs which are extracted by the traffic signs detection system (TSDS) [3] are color images with 64x64 pixel size, and for the three layers R, G and B we will get 12288 pixels, this great amount of data will delay the recognition system if we use it as an input vector for the classifier. To reduce the amount of data processed by the classifier, we apply a simple smoothing on the blobs, by using the Gaussian filter, the use histogram projection (HP) technic [2] for each color channel, for both vertical and horizontal.

So, points CX_i on the horizontal axis are computed by:

$$CX_i = \frac{1}{255} * \frac{1}{64} \sum_{j=1}^{64} C_{i,j} \quad i = 1,2, \dots, 4 \quad (1)$$

Moreover, points CY_j on the vertical axis are computed by:

$$CY_j = \frac{1}{255} * \frac{1}{64} \sum_{i=1}^{64} C_{i,j} \quad j = 1,2, \dots, 4 \quad (2)$$

$C_{i,j}$ is the intensity of the pixel (i, j) in the layer C.

C is the red layer R, green layer G or blue layer B,

CX_i and CY_j values are between 0 and 1.

The new input vector has 384 normalized elements, the 64 elements of CX_i and 64 others of CY_j for the three layers RGB. This is 32 times lower than the amount of initial data.

B. Histogram Oriented Gradient (HOG) Technic

The original HOG technic introduced in 2005 [7], as a feature descriptor of image. The idea of HOG is based on gradient direction; it consists in calculating the histogram of

oriented gradient in local area of blob extracted from the detection step.

The algorithm steps are shown as follows:

Image preprocessing: firstly, converting the extracted color blobs into gray, then using the Gamma correction for normalization, reducing the illumination influence, and suppressing noise.

Gradient calculation: calculating the gradient of image in horizontal and vertical direction of blobs using Sobel edge operator. The formula for the calculation is:

$$G_x(x, y) = H(x + 1, y) - H(x - 1, y) \quad (3)$$

$$G_y(x, y) = H(x, y + 1) - H(x, y - 1) \quad (4)$$

Where $H(x, y)$ is value of the pixel,

$G_x(x, y)$, $G_y(x, y)$ are respectively gradients at the vertical and horizontal direction of pixel (x, y) .

Therefore, gradient magnitude $G(x, y)$ and gradient direction $a(x, y)$ of pixel (x, y) formulas are cited below:

$$G(x, y) = \sqrt{G_x^2(x, y) + G_y^2(x, y)} \quad (5)$$

$$a(x, y) = \tan^{-1} \left(\frac{G_y(x, y)}{G_x(x, y)} \right) \quad (6)$$

HOG does not need to extract the feature of the whole picture. We can divide the image into great number of cells, and then calculate the gradient or edge direction histogram of each pixel in all cells. Meanwhile, we could take several cells to form a block in order to increase the performance of the algorithm. To obtain the feature of gradient direction, several blocks could be taken to compose a connected graph, then we normalize the gradient of each cell in these blocks.

III. MLP AND SVM CLASSIFIERS

A. MLP Classifier

Multi-Layer Perceptron (MLP) is the most used type of Artificial neural networks (ANN) [9], these are the biologically inspired simulations implemented on the computer in order to perform certain specific tasks like, classification, pattern recognition etc. ANNs are often used in machine learning. It was introduced in the 1940s and it was abandoned because of the inefficient training algorithms used and the lack of computing power. Recently, with the development of computers, especially the computing power, and the storage capacity, they have started to be used again, and several technics are developed to improve the ANNs performances

The MLP includes at least three layers. One input layer; one or more hidden layers and one output layer. Each layer contains one or more neurons directionally linked with the neurons from the previous and the next layer. Fig. 1 represents an example of a 3-layer perceptron with three inputs, two outputs, and the hidden layer including four neurons.

All the neurons in MLP are similar. Each of them takes the output values from several neurons in the previous layer as input and passes his response to several neurons in the next layer.

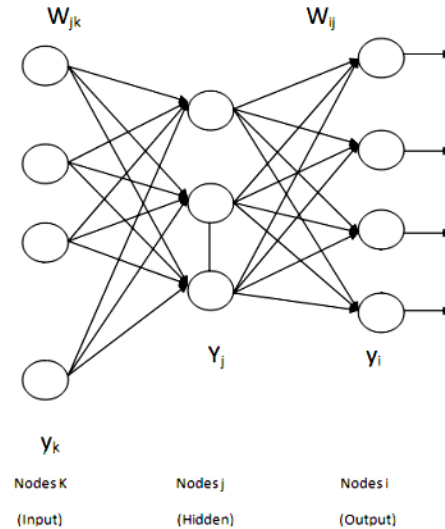


Fig. 1. Typical architecture of a MLP neural network.

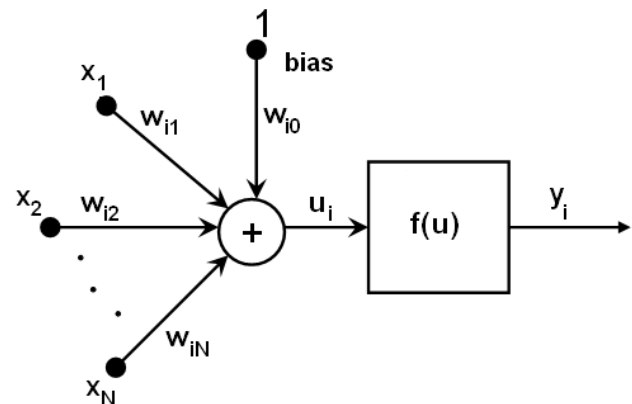


Fig. 2. Computing process in a neuron.

In each neuron the values, retrieved from the previous layer, are summed up with certain weights plus the bias term. The sum is transformed using the activation function f that may be also different for different neurons (Fig. 2). In other words, given the outputs x_j of the layer n , the outputs y_i of the layer $n+1$ are computed as:

$$u_i = \sum_j (w_{i,j}^{n+1} * x_j) + w_{i,bias}^{n+1} \quad (7)$$

$$y_j = f(u_i) \quad (8)$$

The activation function that used in this paper is binary sigmoid function, which is defined as:

$$f(x) = \beta * \frac{1 - e^{-\alpha x}}{1 + e^{-\alpha x}} \quad (9)$$

MLP learns iteratively by adjusting its weights and bias to yield desired output. Several learning algorithms are developed for this task, the most common of them are the gradient descent and the back-propagation, they use a gradient search technique to minimize the mean square error (MSE) between the actual and the desired net outputs.

$$MSE = \frac{1}{l} \sum_{j=1}^N (y_j - \hat{y}_j)^2 \quad (10)$$

Initially a small random weights and internal thresholds are selected for training the MLP. All training data are repeatedly given to the net that adjust their weights after every trial using Information distinguishing the right class until weights converges and the (MSE) is reduced to an acceptable minimal value.

B. SVM Classifier

SVM initially introduced by Cortes and Vapnik in [8], are conceived to solve the binary classification problems. For a given training sample set S , with n data training samples. For x_i the feature vector of training samples and y_i the label of training samples, where $y_i = 1$ and $y_i = -1$, determine the two types of training samples:

$$S = \{(x_i, y_i) | x_i \in R_d, y_i \in \{-1, 1\}\}_{i=1}^n \quad (11)$$

Finding an optimal hyperplane $h(x)$, is the basic idea of SVM, this hyperplane should separate two classes of labels of training data, and should be as far as possible from the members of the both classes. Data can be linearly separable in this case. The format of the hyperplane function is as follows:

$$h(x) = sgn(w \cdot x + b) \quad (12)$$

- w is the normal to the hyperplane,
- x is the input vector,
- b is the deviation.

when data cannot be separated by a linear function, the solution is to map the input vector x into a high dimensional space $\varphi(x)$, if there is a “kernel function” satisfied $K(x_i, x_j) = \langle \varphi(x_i), \varphi(x_j) \rangle$, we can calculate only the kernel function $K(x_i, x_j)$ instead to compute $\varphi(x_i)$. The most used kernel functions in SVM are the polynomial, the radial and the sigmoid kernel functions whose formulas are shown as follows:

- Polynomial kernel function:

$$K(x_i, x_j) = [\gamma(x_i, x_j) + bias]^t$$

- Radial basis kernel function:

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2)$$

- Sigmoid kernel:

$$K(x_i, x_j) = \tanh(\gamma(x_i, x_j) + bias)$$

- γ is the width of the kernel function,
- $bias$ is the bias coefficient,
- t is the order of polynomial.
- x is the given input vector.

The decision function is:

$$h(x) = sgn(\sum_{i=1}^{N_s} a_i y_i K(s_i, x) + b) \quad (13)$$

- $a_i \geq 0$
- N_s is the number of support vectors,
- s_i is the support vector,
- $K(s_i, x)$ denotes kernel function.

IV. EXPERIMENTAL AND ANALYSIS

A. Experimental Environment

In this work, we propose four methods for the recognition system:

- Multi-layer perceptron classifier with the histogram projection descriptor (HP-MLP).
- Multi-layer perceptron classifier with the histogram oriented gradient descriptor (HOG-MLP).
- Support-vector-machine classifier with the histogram projection descriptor HP-SVM.
- Support-vector-machine classifier with the histogram oriented gradient descriptor HOG-SVM.

To train and test these proposed methods, the traffic signs images database¹ used in this paper contains 300 color images with natural background in under variable conditions, and with 1300x800 pixels size. We use the traffic signs detection system (TSDS) presented in our previous work, to extract traffic signs, resized to 64x64 pixels, and after eliminating the insignificant blobs, we proceed to a manual classification of traffic signs blobs depending to shape and to color.

Four color-shape datasets are generated: red-circular signs dataset, red-triangular signs dataset, blue-circular signs dataset, and blue-quadrangular signs dataset.

These datasets are prepared to train and to test the methods HP-MLP, HOG-MLP, HP-SVM and HOG-SVM, which constitute the recognition system. The process for preparing datasets of traffic signs is presented in Fig. 3.

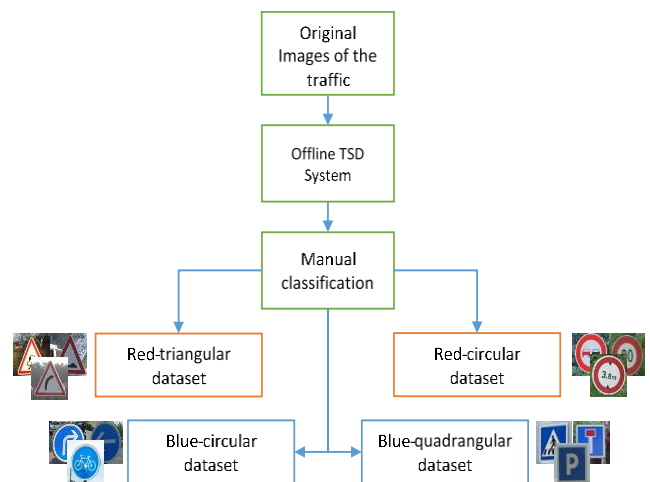


Fig. 3. Process for preparing the traffic signs dataset.

¹ DATASET, THE GERMAN TRAFFIC SIGN DETECTION BENCHMARK, <http://benchmark.ini.rub.de/?section=gtsdb&subsection=dataset> 2013

The traffic sign global dataset generated composed of 2535 blobs of traffic signs, classified in 49 classes. 1724 blobs are used for training our classifiers and, 811 blobs for the test.

To evaluate the performance of these methods, the implementing environment is based on a Desktop computer with Intel® Core™ i3 CPU M370 @ 2.40GHz processor, 4Go memory, and using Microsoft Visual Studio 2012 and OpenCV library 2.4.4 as the software platform.

B. Descriptor-Classifer Recognition Methods

In the stage of training, we use HP and HOG to extract de descriptor of each element in the training color-shape datasets, these descriptors are to train the MLP and the SVM classifiers, and we do same for each color-shape dataset.

In the stage of testing, we extract the HP and the HOG descriptors from elements of dataset of testing, and we use the correspondent trained MLP and SVM classifiers, to predict the class of the given traffic sign. The framework of training and testing this method is shown in Fig. 4.

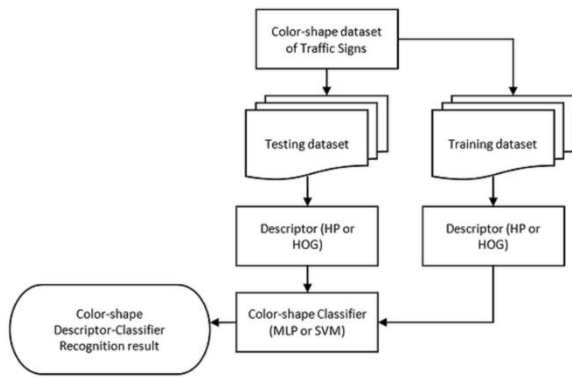


Fig. 4. Framework for training and testing descriptor-classifier recognition method.

C. Descriptor-Classifer Recognition Methods

Results of the four recognition methods are presented in the following figures:

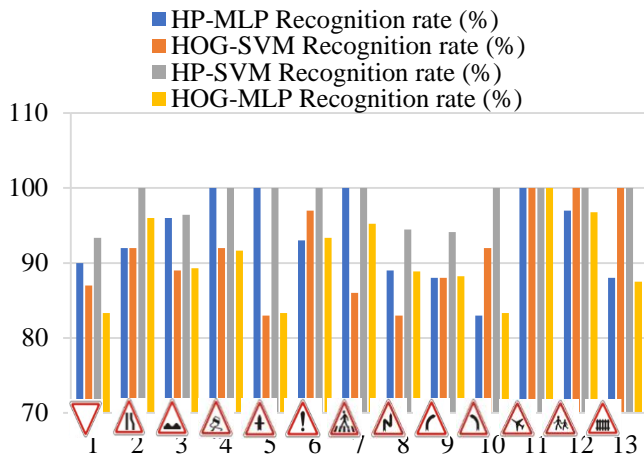


Fig. 5. The Performances of HP-MLP, HP-SVM, HOG-MLP and HOG-SVM using a red triangular Road Signs dataset.

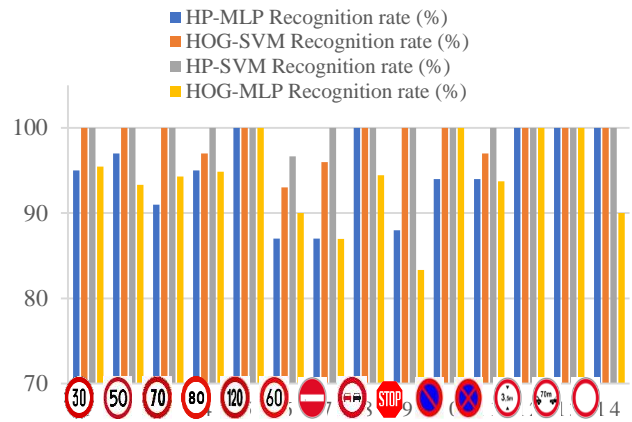


Fig. 6. The Performances of HP-MLP, HP-SVM, HOG-MLP and HOG-SVM using a red circular Road Signs dataset.

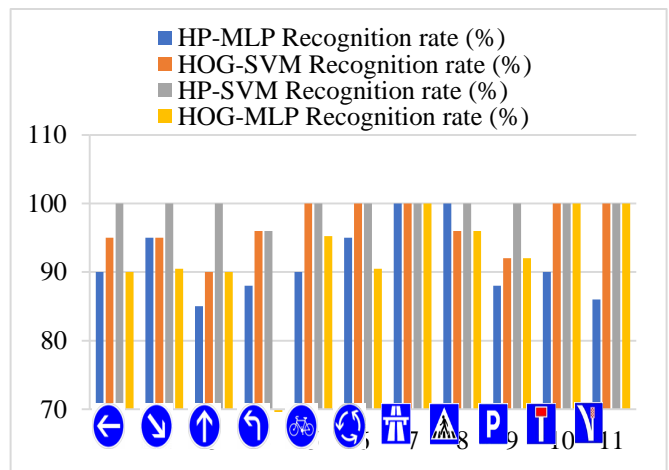


Fig. 7. The Performance of HP-MLP, HP-SVM, HOG-MLP and HOG-SVM using a blue circular and quadrangular Road Signs dataset.

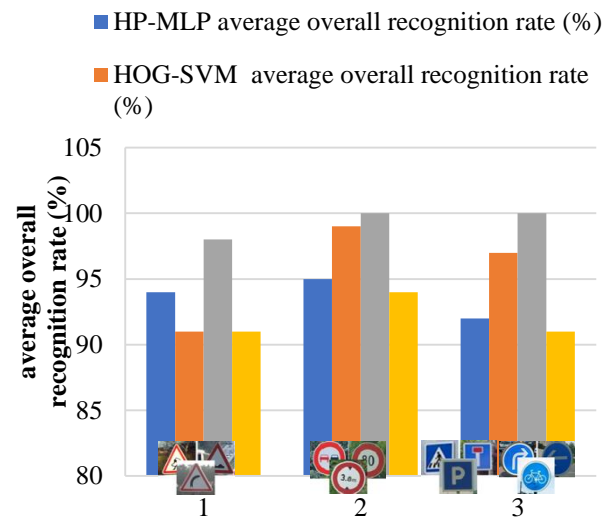


Fig. 8. The average overall recognition rate of HP-MLP, HP-SVM, HOG-MLP and HOG-SVM.

The results presented in Fig. 5, 6 and 7, shows that the HP-SVM method is more efficient than the other methods for all tested kind of traffic signs, with an average overall recognition rate of 99,33% as shown in Fig. 8.

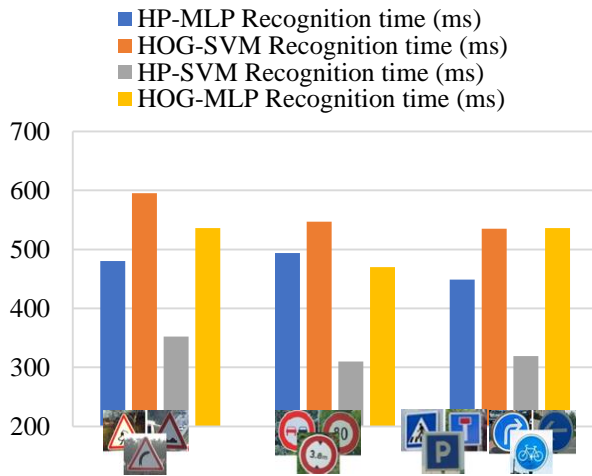


Fig. 9. The average overall recognition time of HP-MLP, HP-SVM, HOG-MLP and HOG-SVM.

The HP-SVM method, as is shown in Fig. 9 HP-SVM, in addition to its efficiency in recognizing traffic signs, has a competitive processing time compared to the other methods. This will facilitate a future real-time implementation of traffic sign recognition system.

The performance of HP-SVM is due to the right choice of the amount of data processed and delivered by the HP Descriptor. The compromise of the speed and the efficiency of recognition of the traffic signs are controlled by the size and representativeness of the information delivered by the chosen descriptor.

Above all, as is shown by the experimental results, HP-SVM method compared to the author methods, achieve the highest accuracy for recognition of all kind of traffic signs, it could be used as a core of recognition step, and combined with the detection system developed in the previous work [3] for the

detection step, to evaluate the performance of an entire system of detection and recognition of road signs.

V. CONCLUSION

This work presented a comparative study of four methods, designed and developed for a traffics sign recognition system. These methods are combinations of two descriptors HP and HOG and two classifiers MLP and SVM.

The study concluded that the HP-SVM method presents a competitive performance with respect to the accuracy of recognizing traffic signs and processing time, this makes it the appropriate method to be used in our traffic signs recognition system (TSRS), and to be combined with our TSD system, to build a robust real-time traffic signs detection and recognition system, which will be the aim objective of a future work.

REFERENCES

- [1] A. SALHI, B. MINAOUI & M. FAKIR (2015). Real-Time Detection of Road Signs. Journal of Electronic Commerce in Organisations (JECO), IGI, pages 36-46.
- [2] A. SALHI, B. MINAOUI & M. FAKIR (2014). Robust Automatic Traffic Signs Recognition Using Fast Polygonal Approximation of Digital Curves and Neural Network. Int. Journal of Adv Comp Sc and Applications(IJACSA), Special Issue on Advances in Vehicular Ad hoc Networking and Applications, IEEE, pages 433-437.
- [3] A. SALHI, B. MINAOUI & M. FAKIR (2014). Robust Automatic Traffic Signs Detection using fast polygonal approximation of digital curves. Multimedia Computing and Systems (ICMCS), 2014 International Conference on, IEEE, pages 14-16.
- [4] Y. Zeng, X. Xu, D. Shen & all (2016), Traffic Sign Recognition Using Kernel Extreme Learning Machines With Deep Perceptual Features, IEEE Transactions on Intelligent Transportation Systems, IEEE, pages 1-7.
- [5] M. Xing, M. Chunyang, W. Yan & all (2016), Traffic sign detection and recognition using color standardization and Zernike moments, Control and Decision Conference (CCDC), Chinese, IEEE, pages 28-30.
- [6] A. de la Escalera, J. Ma Armingol & M. Mata (2003), Traffic sign recognition and analysis for intelligent vehicles, Image and Vision Computing, Elsevier, pages 247-258,
- [7] N. Dalal & B. Triggs (2005), Histograms of oriented gradients for human detection, Computer Vision and Pattern Recognition. IEEE Computer Society Conference on. IEEE, pages: 886-893.
- [8] C. Cortes & V. Vapnik (1995), Support-vector Networks. Machine Learning, springer link pages 273-297
- [9] Schalkoff, R. J. (1997). Artificial neural networks (Vol. 1). New York: McGraw-Hill.