

# Robust Automatic Traffic Signs Recognition Using Fast Polygonal Approximation of Digital Curves and Neural Network

Abderrahim SALHI<sup>1</sup>, Brahim MINAOUI<sup>2</sup>, Mohamed FAKIR<sup>3</sup>  
Information Processing and Decision Laboratory, Sultan Moulay Slimane University,

**Abstract**—Traffic Sign Detection and Recognition (TSDR) has many features help the driver in improving the safety and comfort, today it is widely used in the automotive manufacturing sector, a robust detection and recognition system a good solution for driver assistance systems, it can warn the driver and control or prohibit certain actions which significantly increase driving safety and comfort. This paper presents a study to design, implement and test a method of detection and recognition of road signs based on computer vision. The approach adopted in this work consists of two main modules: a detection module which is based on color segmentation and edge detection to identify areas of the scene may contain road signs and a recognition module based on the multilayer perceptrons whose role is to match the patterns detected with road signs corresponding visual information. The development of these two modules is performed using the C/C++ language and the OpenCV library. The tests are performed on a set of real images of traffic to show the performance of the system being developed.

**Keywords**—Traffic sign; recognition; detection; pattern matching; image processing; Polygonal Approximation of digital curves

## I. INTRODUCTION

A driver assistance system is designed to help the driver to better control the vehicle in difficult circumstances (tired, stress, carelessness, poor vision ...) to help increase safety and the safety of other drivers, pedestrians etc. involved in the traffic on the roads.

Computer vision is a promising approach for addressing this problem. Automatic control of the braking system, automatic speed control, generation of alerts and notifications, corresponding to the various events encountered on the road etc., are some examples among many installations of driver assistance that could be developed, based on this approach.

The aim of this work is to design, implement and test a system of Traffic signs Recognition.

Traffic signs Recognition system developed in this work like [1-3] composed of two main Modules. The first Module is the detection Module which is designed to detect and extract zones may contain traffic signs in the image, based on the

particular color and geometry of these panels; and using image processing techniques such as color segmentation, threshold technique, Gaussian filter, canny edge detection, it allows us to greatly reduce the amount of information to be processed. This allows faster processing and facilitating the realization of a system operating in real time.

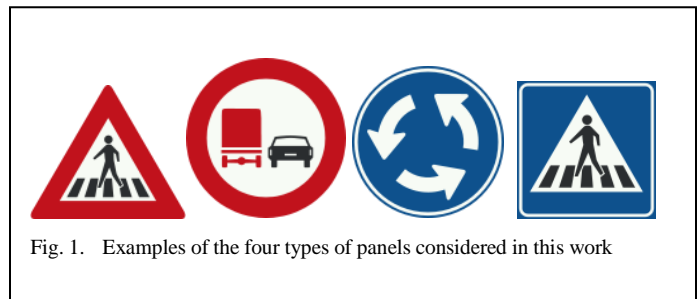


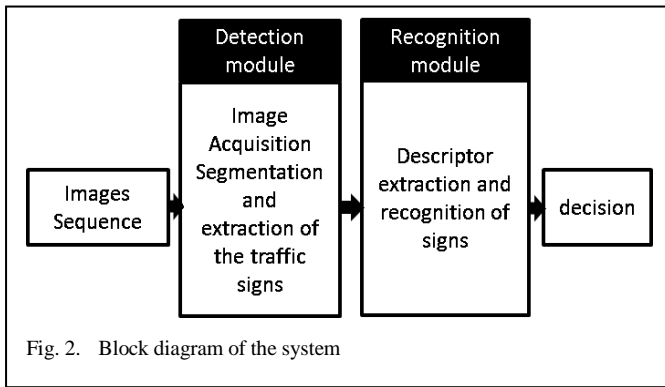
Fig. 1. Examples of the four types of panels considered in this work

The second Module is the recognition Module; it helps identify road signs by comparing information provided by the detection Module with models of road signs that have been learned beforehand, based multilayer perceptrons.

The algorithms are designed to detect and recognize red triangular and circular panels, and blue circular and rectangular panels. Making this choice of traffic signs to recognize, it covers a vast majority of common traffic signs and also the most important signals that advertise a danger on the road, a ban (speed limit) or require a driver to certain behavior (blue signals) figure (FIG 1). Shows some examples of signs used in this work.

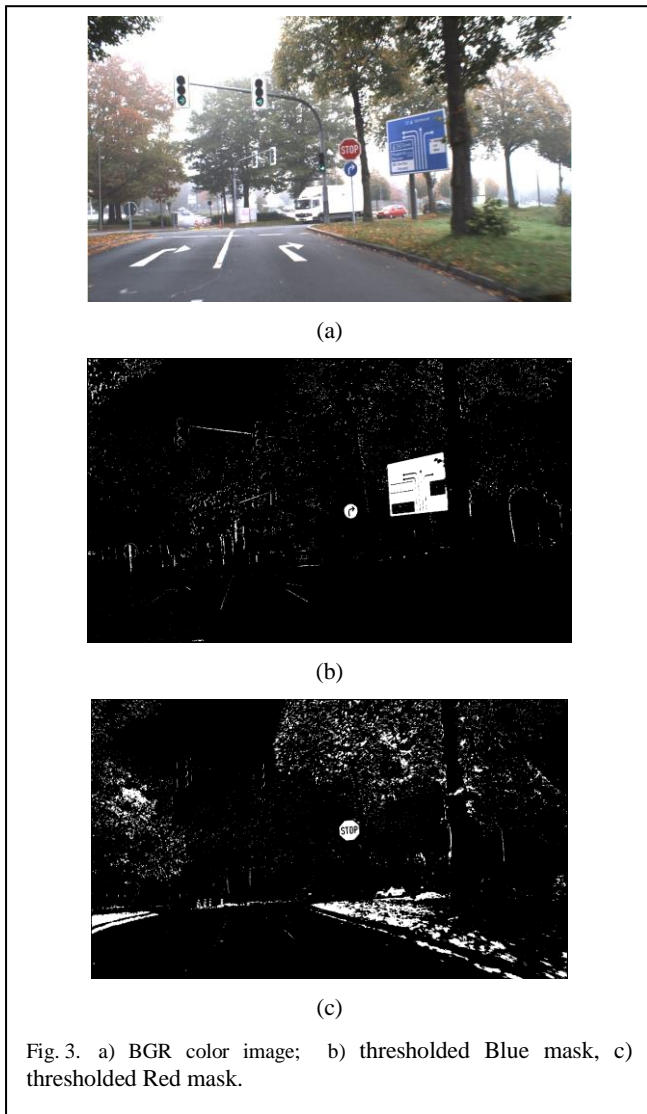
## II. SYSTEM DESCRIPTION

The first section in our system (FIG 2) is the detection module, in this part we read the RGB image from real images sequence and convert color to HSV space, to minimize the effects of changing brightness, followed by a color segmentation using threshold [5], two masks are generated, the first one, is designed to find zones which may be a red circles or a red triangles shapes, using shape recognition [8], the second mask, is to find zones which may be a bleu circles or a bleu rectangles shapes by the same way.



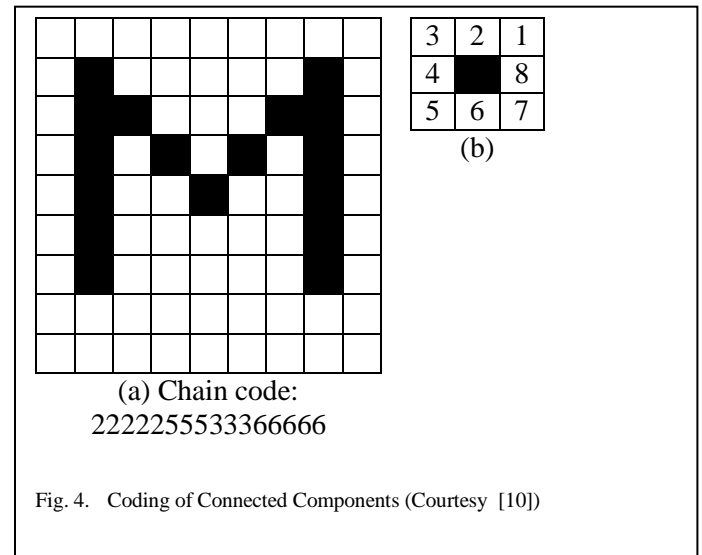
The second section is the Recognition module, in this part we extract the descriptors of zones (blobs) provided by the detection module, to be matched with templates in database [9] using Multilayer Perceptron (MLP) algorithm, and take the appropriate decision.

### III. TRAFFIC SIGNS DETECTION MODULE



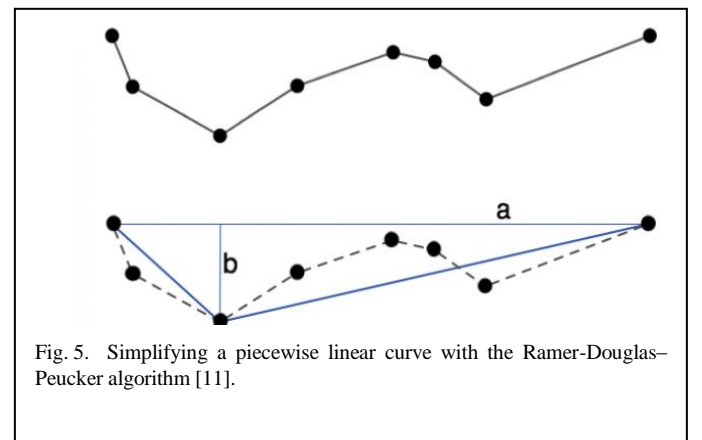
The image is read in RGB (Red Green Blue) color mode (FIG 3: a), and then converted to HSV (Hue Saturation Value) color space, then we use the thresholding technique to generate two red mask and blue (FIG 3: a & FIG 3: b), these masks will be used later to find the contours, and before that, we apply the technique of smoothing by using a Gaussian filter, and the technique of Canny edge detection to improve the image and easily get the desired region.

The obtained binary image for each mask is processed to retrieve contours by `findContours` function for binary image and the retrieved contours are returned and stocked in chain format.



To represent contours in the OpenCV library, we use the Freeman method or the chain code. For any pixel all its neighbors with numbers from 0 to 7 can be enumerated as Fig. 4(a). The 0-neighbor denotes the pixel on the right side, etc.

As a sequence of 8-connected points, the border can be stored as the coordinates of the initial point, followed by codes (from 0 to 7) that specify the location of the next point relative to the current one. (Fig. 4 (b) illustrates example of Freeman coding of connected components.)



The extracted contours are used to find the edges that can match the shapes of traffic signs, like triangular shapes, circular shapes and rectangular shapes. In the contours returned from the red mask, we will look for the triangular and circular shapes, and in those returned from blue mask, we will search for rectangular and circular shapes.

To find these shapes will be based on the algorithm of Ramer -Douglas – Peucker [11]. The idea is to simplify a polyline (n nodes) and replace it with a single line (two-node) if the distance of the farthest from the line formed by the ends of the polyline node is below a threshold, as shown in (FIG 5) The algorithm works recursively by the method of "divide and rule", to initialize the algorithm we select the first and last node (for a polyline), or any node (such as a polygon). These are the terminals. At each step, through all the nodes between the terminals and the farthest segment formed by the terminal node is selected.

- If there is no node between terminals algorithm ends.
- If this distance is less than a certain threshold are removed all the nodes between terminals.
- If it exceeds the polyline is cannot be directly simplified. Called recursively the algorithm on two sub-parts of the polyline: the first terminal to the remote node and the remote terminal to the final node.

The approxpolyDP function provided by the OpenCV library [10] is used to implement this algorithm, and takes as input the contours found by findContours function, and the threshold, it return a table of points forming the new polygon approached to the real contour.

Our algorithm tests the size of the returned table of points, and if it contains tree elements it means that the founded shape is a triangle, if it contains four elements, it means that the shape founded corresponding to a rectangle.

For the circular shapes we empirically take the size of the table from beyond six elements.

To close that founded shapes to those of traffic signs, we impose certain criteria, such as: triangles should be equilateral with an error near; their surfaces must be included in a well-defined interval, and the something for rectangles.

For circular shapes, we use the ellipse detection technique [1] provided by OpenCV library [10] and then the sizes of its two axes are taken as criteria. The shapes that satisfy the criteria imposed will be cropped and extracted from the color natural image; these blobs will be resized to 32x32 pixels to be used in the recognition phase.

This allows us to significantly reduce the number of shapes processed and computing time cost of the algorithm later; Figure (FIG 6) shows the flowchart of the detection module.

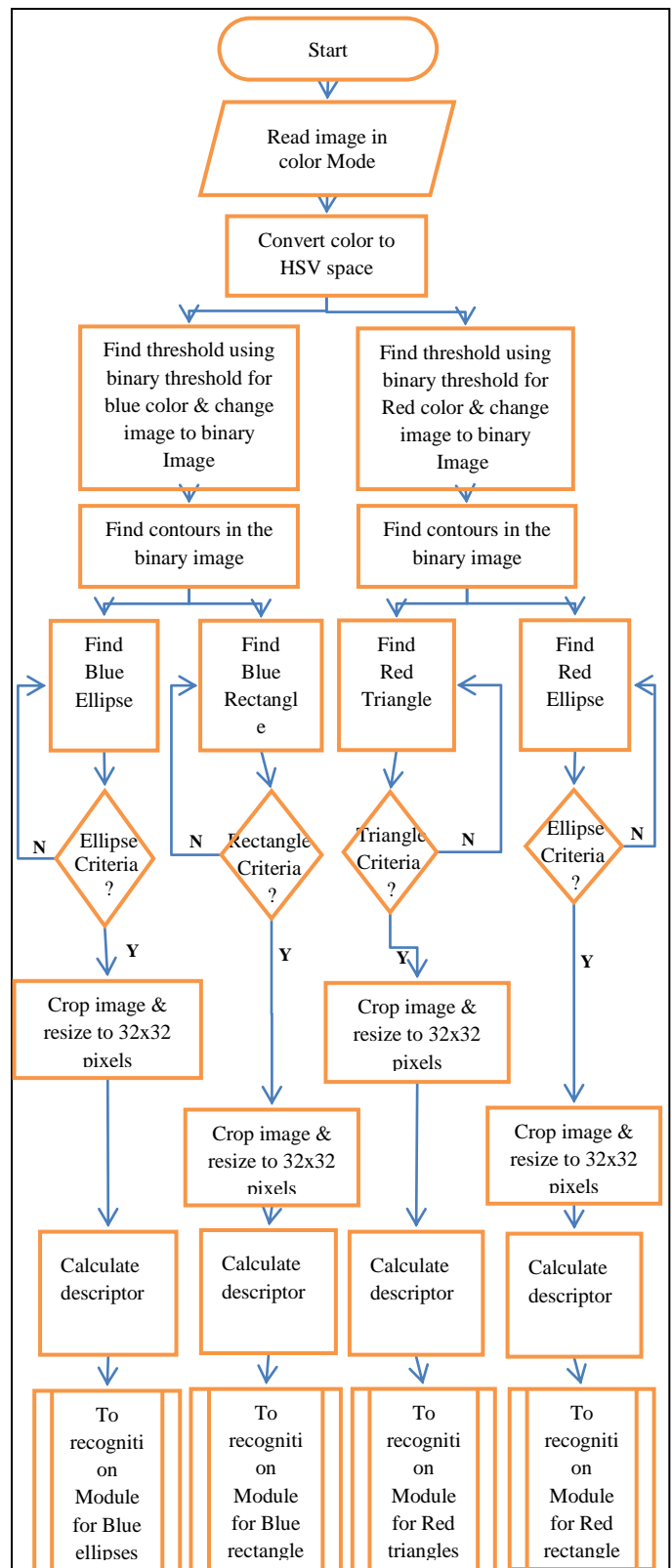


Fig. 6. flowchart of the detection module

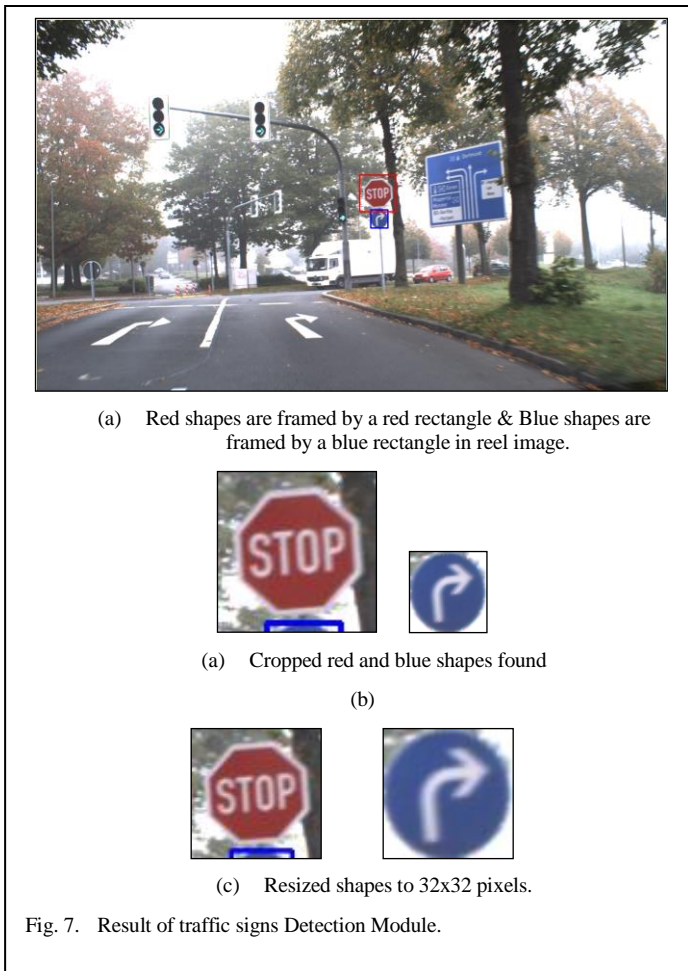


Fig. 7. Result of traffic signs Detection Module.

The traffic signs images database [9] used in this paper contains 300 color images with natural background, and with 1300x800 pixels size. Our program is tested on all these images, the figure (FIG 7) shows one result of traffic signs Detection Module.

The program may extract also some blobs which are not a traffic signs, but all these blobs will be presented as input to Traffic Signs Recognition Module to decide which the right traffic sign is, and which the bad is.

Figure (FIG 8 & 9) shows respectively the number of blobs extracted from the first 100 images and the time spent in processing each image by Traffic Signs Recognition Module.

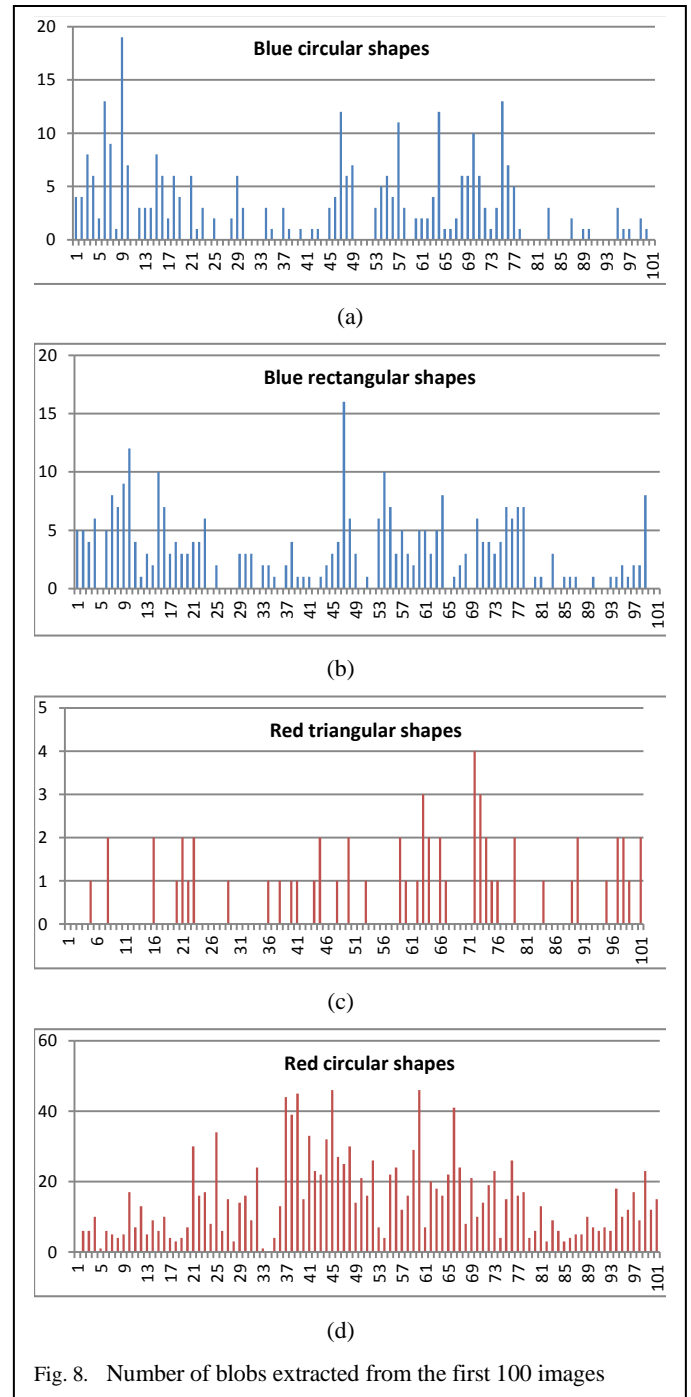
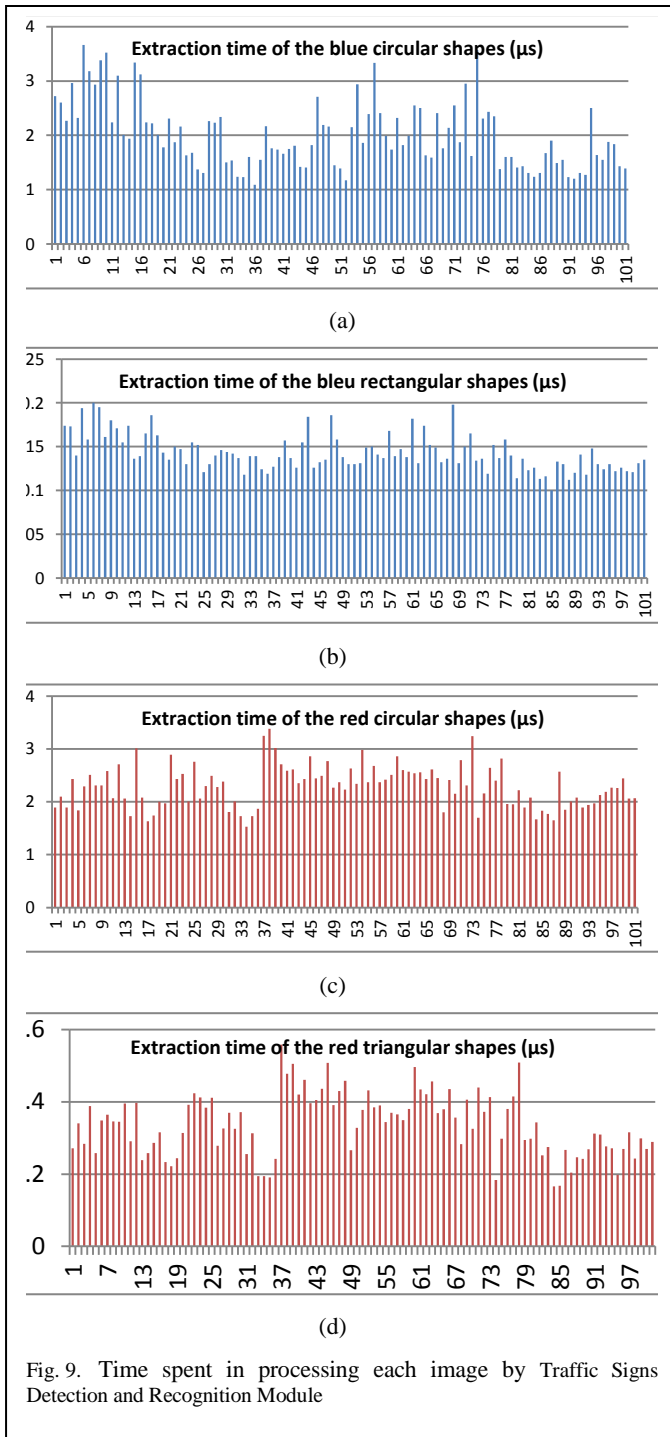


Fig. 8. Number of blobs extracted from the first 100 images



#### IV. TRAFFIC SIGNS RECOGNITION MODULE

##### A. Preprocessing Stage.

Before passing the extracted blobs to the Traffic Signs Recognition Module which is based on neural network, a preprocessing stage was required to reduce the amount of information to process, and reduce the computing time cost of the system thereafter. Through this stage, we proceed by extracting the descriptor of each blob.

Blobs are presented in the 32x32 size, which is 1024 pixels, and for the three layers R, G and B, the size of neural network input vector will be 3072, and this will delay the system. For each channel, a projection of the pixels is done on both vertical and horizontal axes.

So for each point  $CX_i$  on the horizontal axis:

$$CX_i = \frac{1}{255} * \frac{1}{32} \sum_{j=1}^{32} C_{i,j} \quad i=1, 2... 32 \quad (1)$$

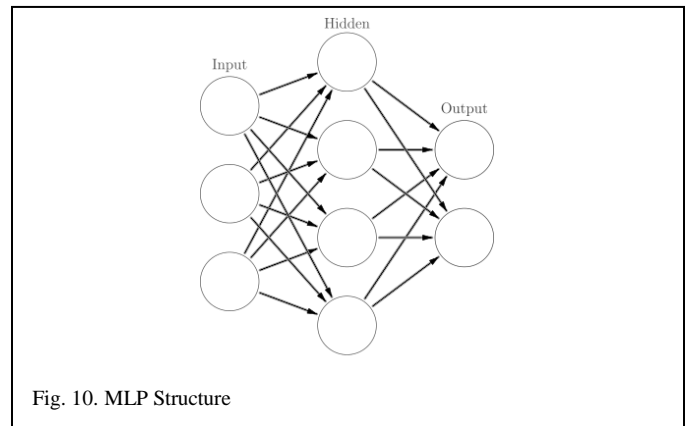
And for each point  $CY_i$  on the vertical axis:

$$CY_j = \frac{1}{255} * \frac{1}{32} \sum_{i=1}^{32} C_{i,j} \quad j=1, 2... 32 \quad (2)$$

- $C_{i,j}$  Is the intensity of the pixel whose coordinates are (i, j) of the layer C.
- C Is the red layer R, green layer G or blue layer B.
- $CX_i$  And  $CY_i$  values are between 0 and 1.

The new descriptor is composed of 192 elements, the 32 elements of  $CX_i$ , and 32 others of  $CY_i$  for the three layers RGB.

##### B. traffic signs recognition CORE



The Traffic Signs Recognition Module is implemented by using feed-forward artificial neural networks or, more particularly, multi-layer perceptrons (MLP), the most used type of Artificial neural networks [12]. These are a mathematical model, inspired by the brain that is often used in machine learning. It was initially proposed in the '40s and there was some interest initially, but it waned soon due to the inefficient training algorithms used and the lack of computing power. More recently however they have started to be used again, especially since the introduction of autoencoders, convolutional nets, dropout regularization and other techniques that improve their performance significantly.

The MLP includes at least 3 layers. The first one is the input layer; the last one is the output layer, and one or more hidden layers. Each layer of MLP contains one or more neurons directionally linked with the neurons from the previous and the next layer. The figure (FIG 10) 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 has several input links (it takes the output values from several neurons in the previous layer as input) and several output links (it passes the response to several neurons in the next layer).

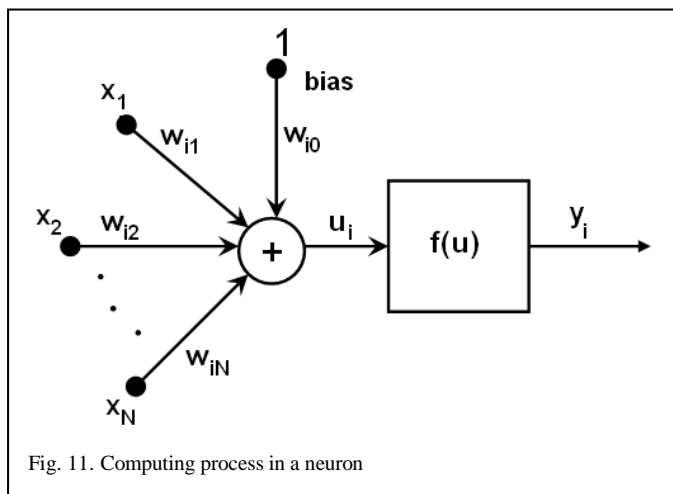


Fig. 11. Computing process in a neuron

The values retrieved from the previous layer are summed up with certain weights, individual for each neuron, plus the bias term. The sum is transformed using the activation function  $f$  that may be also different for different neurons (FIG 11) In other words, given the outputs  $x_j$  of the layer  $n$ , the outputs  $y_i$  of the layer  $n + 1$  are computed as:

$$u_i = \sum_j (w_{i,j}^{n+1} * x_j) + w_{i,bias}^{n+1} \quad (3)$$

$$y_j = f(u_i) \quad (4)$$

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

$$f(x) = \beta * \frac{1 - e^{-\alpha x}}{1 + e^{-\alpha x}} \quad (5)$$

With  $\alpha = 1$  et  $\beta = 1$ , it is shown like illustrate (FIG 12); the sigmoidal function basically has some very useful mathematical properties, monotonicity and continuity. Monotonicity means that the function  $f(x)$  either always increases or always decreases as  $x$  increases. Moreover, continuity means there are no breaks in the function, it is smooth.

These parameters are intrinsic properties eventually assist networks power to approximate and generalize on functions by learning from data.

The Traffic Signs module used in this paper is separated to four MLP functions; each one is used for one type of blobs, (blue circular blobs, blue rectangular blobs, red circular blobs and red triangular blobs).

The results of Traffic Signs Recognition Module are presented in table I, II and III below.

TABLE I. RESULTS OF RECOGNIZED RED TRIANGULAR TRAFFIC SIGNS

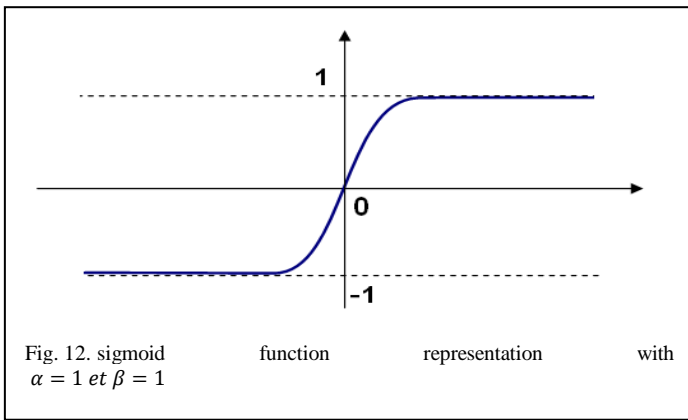
Number of tested images	30	5	3	7	12
Number of Recognized Signs	26	4	2	6	10
Recognition rate (%)	87	80	67	85	83

TABLE II. RESULTS OF RECOGNIZED RED CIRCULAR TRAFFIC SIGNS

Number of tested images	33	18	36	14	10	9	23	11
Number of recognized signs	30	15	32	10	10	8	20	11
Recognition rate (%)	90	83	89	71	100	89	87	100

TABLE III. RESULTS OF RECOGNIZED BLUE CIRCULAR TRAFFIC SIGNS

Number of images	29	5	6	5	7	11	18	5
Number of recognized sign	25	4	4	4	6	9	16	3
Recognition rate (%)	86	80	67	80	86	82	89	60



## V. PERSPECTIVES

Tests show that the Detection Module currently implemented is very fast and good enough (detection rate 99%), even if it's depending on size of images captured. by cons, the Recognition Module is much less efficient (71%). To improve performance of the proposed system, an improvement of the Recognition Module is required, by using or combining other recognition techniques.

We can also use the techniques of multithreading, which allows parallel execution of several programs at the same time, this will significantly improve the execution time of the system which will be more robust and efficient for a real time implementation.

## VI. CONCLUSION

Traffic Sign Recognition is discussed in this study, by using Neural Networks technique. In the first time the Traffic signs Module has processed the input images using image processing techniques, such as, threshold technique, Gaussian filter, canny edge detection, Contours, algorithm of Ramer -Douglas -Peucker and Fit Ellipse. Next the Traffic Signs Recognition based on the Neural Networks, were performed to recognize the traffic sign patterns. The main reason to select this method is to reduce the computational cost in order to facilitate the real time implementation. The strategy is to reduce number of MLP inputs by pre-processing blobs before giving them to Traffic Sign Recognition Module

The system was ended with a Traffic Sign Recognition Module based on Neural Networks, which were trained, and validated with a database containing more than 300 real images, with complex natural background, to find the best network architecture.

## REFERENCES

- [1] A. Lorsakul and J. Suthakorn, "Traffic Sign Recognition for Intelligent Vehicle/Driver Assistance System Using Neural Network on OpenCV", The 4th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI 2007)
- [2] R. Belaroussi — J-P. Tarel, "Détection des panneaux de signalisation routière par accumulation bivariée", *Traitement du Signal* 27, 3 (2010) pp 265-296- DOI : 10.3166/ts.27.265-296 .
- [3] F.A. Aly and A.E. Alaa. Detection, categorization and recognition of road signs for autonomous navigation. In *Proceeding of Advanced Concepts for Intelligent Vision System*, Brussels, Belgium, Aug 2004.
- [4] F.A. Aly and A.E. Alaa. Detection, categorization and recognition of road signs for autonomous navigation. In *Proceeding of Advanced Concepts for Intelligent Vision System*, Brussels, Belgium, Aug 2004.
- [5] H. X. Liu, and B. Ran, "Vision-Based Stop Sign Detection and Recognition System for Intelligent Vehicle", *Transportation Research Board (TRB) Annual Meeting 2001*, Washington, D.C., USA, January 11, 2001.
- [6] H. Fleyeh, and M. Dougherty, "Road And Traffic Sign Detection And Recognition", *Proceedings of the 16th Mini - EURO Conference and 10th Meeting of EWGT*, pp. 644-653.
- [7] [4] Arturo de la Escalera, Luis E. Moreno, Miguel Angel Salichs, and José Maria Armingol. Road traffic sign detection and classification. *IEEE Transactions on Industrial Electronics*, 44(6) :848-859, Dec 1997.
- [8] S. Belongie, "Shape Matching and Object Recognition Using Shape Contexts", *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, VOL 24, NO 24, APRIL 2002.
- [9] DATASET "THE GERMAN TRAFFIC SIGN DETECTION BENCHMARK", <http://benchmark.ini.rub.de/?section=gtsdb&subsection=dataset> 2013
- [10] Intel Corporation, "Open Source Computer Vision Library," Reference Manual, Copyright © 1999-2001, Available: [www.developer.intel.com](http://www.developer.intel.com)
- [11] Urs Ramer, "An iterative procedure for the polygonal approximation of plane curves", *Computer Graphics and Image Processing*, 1(3), 244-256 (1972)
- [12] Laurence Fausett, "Fundamentals of Neural Networks Architectures, Algorithms, and Applications", Prentice Hall Upper Saddle River, New Jersey 1994.