Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

Traffic sign recognition application based on image processing techniques

Rubén Laguna*, Rubén Barrientos*, L. Felipe Blázquez*, Luis J. Miguel**

**Department of Electrical and Automatic Control and Systems Engineering, University of León, E.II.I.I., Campus de Vegazana s/n, 24071 León, Spain. (Tel: 34 987293471; e-mail: {rlagug00,rbarrm00,diefbq}@unileon.es).*

***Department of Automatic Control and Systems Engineering, University of Valladolid, E.I.I., Paseo del Cauce s/n, 47011 Valladolid, Spain. (Tel: 34 983423545; e-mail: ljmiguel@eii.uva.es).*

Abstract: This paper describes a software application for traffic sign recognition (TSR). The application works in four stages. First, an image preprocessing step and the detection of regions of interest (ROIs), which involves a series of steps that include transforming the image to grayscale and applying edge detection by the Laplacian of Gaussian (LOG) filter. Secondly, the potential traffic signs detection, where the ROIs are compared with each shape pattern. Thirdly, a recognition stage using a cross-correlation algorithm, where each potential traffic sign, if validated, is classified according to the data-base of traffic signs. Finally, the previous stages can be managed and controlled by a graphical user interface, which has been specially designed for this purpose. The results obtained show a good performance of the developed application, taking into account acceptable conditions of size and contrast of the input image.

Keywords: Traffic sign recognition; image processing; image segmentation; visual pattern recognition; boundary detection; cross correlation functions.

1. INTRODUCTION

Digital image processing uses algorithms to process digital images. Although it may seem recent technology, many of the techniques were developed in the 1960s at the Massachusetts Institute of Technology, University of Maryland and other research facilities (Chen *et al*., 1993). However, due to the high costs of computers at this time, the digital imaging process was too expensive for many to even consider. That changed in the 1970s, when digital image processing proliferated as cheaper computers and dedicated hardware became available. With the fast computers and signal processors available in the 2000s, images could be processed in real time and digital image processing has become the most common form of image processing (González and Woods, 2007). Computer vision is increasingly used in the field of intelligent transport (Mapanga and Ragavan, 2012), and traffic sign recognition is a very important part of this. These systems are typically based on detecting a region of interest (ROI), in which the traffic sign is located, using characteristics such as color and geometric form. In computer vision, the ROI defines the borders of an object under consideration and is commonly used in many application areas, such as medical imaging (Dougherty, 2009).

Video processing techniques for traffic applications have been an attractive field of research during the last two decades (Piccioli *et al*., 1996; Kastrinaki *et al*., 2003). Several techniques have been proposed to develop traffic sign recognition (TSR) systems (Ruta *et. al*., 2010). The main goal of these systems is the detection and recognition of every traffic sign present in the scene (Ritter *et. al*., 1995). In the last few years, the TSR systems have become important components of advanced driver assistance systems (ADAS) (Gil *et al*., 2008). The main difficulty that TSR systems face is the poor image quality due to low resolution, bad weather conditions or inadequate illumination.

The application developed in this paper determines the area of the original image where the potential traffic sign is located, and compares it with a database which contains normalized traffic signs. The process starts with a preprocessing stage for the input image, where the parameters of the image, such as resolution or contrast, are modified to guarantee that the filters and algorithms used later behave properly. When the parameters of the image have been adjusted, an edge detection algorithm is used in order to determine the potential areas of the image where a possible traffic sign may be located. The next step is to separate the regions of interest (ROIs) found in the image and obtain the potential traffic signs. Every potential traffic sign is submitted to a recognition process, using a cross-correlation algorithm that compares each one with a database, which contains patterns of traffic signs. This software includes a graphical user interface, which allows the user to control each stage of the application. The results obtained show a high success rate, dependent on the environmental conditions of the input image and its resolution.

This paper is organized as follows. In section 2, the preprocessing process and the ROI detection are described. Section 3 shows the potential traffic sign detection stage. Section 4 details the traffic sign recognition process. Section 5 describes the graphical user interface. In section 6 the experiments carried out are described and the results obtained are presented. Section 7 contains the conclusions of the work and finally some references are given.

2. IMAGE PREPROCESSING AND REGIONS OF INTEREST DETECTION

The first step of the preprocessing is to adjust the image size, reducing the space occupied when the resolution of the input image is too high. If the image size were too high, it would slow down the execution of the algorithms to the point that it would restrict the program execution. Next, a contrast limited adaptive histogram equalization (Sepasian *et al*., 2008; Reza, 2004) is performed to enhance the contrast of the image by transforming the values obtained in the intensity image. This method is chosen to prevent the over amplification of noise that adaptive histogram equalization can give rise to (Pizer *et al*., 1987). The contrast amplification in the vicinity of a pixel is proportional to the slope of the neighborhood cumulative distribution function and to the value of the histogram at that pixel value. The next step is to transform the input image into a gray scale image, so the edge detection algorithm can be applied (Pei and Horng, 2000; Wan *et al*., 2007). In Fig. 1, a typical traffic image taken as example and its corresponding gray-scale image, are shown.

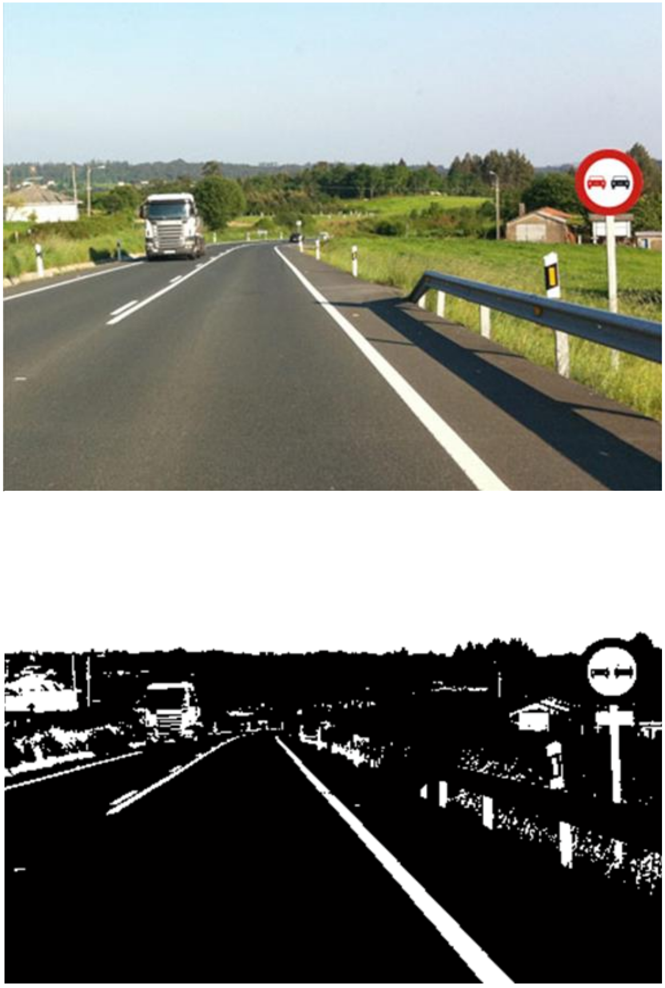


Fig. 1. Traffic image and its corresponding grayscale image.

The algorithm used to detect edges in the image is the Laplacian of Gaussian (*LOG*), which takes a grayscale image as input and produces another grayscale image as output, where only the edges are shown. The *LOG* is a hybrid filter that contains a Gaussian smoothing filter and a Laplacian filter, and then convolves it with the image to achieve the required result (Sharifi *et al*., 2002). The function used, centered on zero and with standard deviation σ, is given by (1) and has the form shown in Fig. 2, where (x,y) are the coordinates of each pixel.

$$LOG(x,y) = -\frac{1}{\pi\sigma^4}\left[1-\frac{x^2+y^2}{2\sigma^2}\right]e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{1}$$

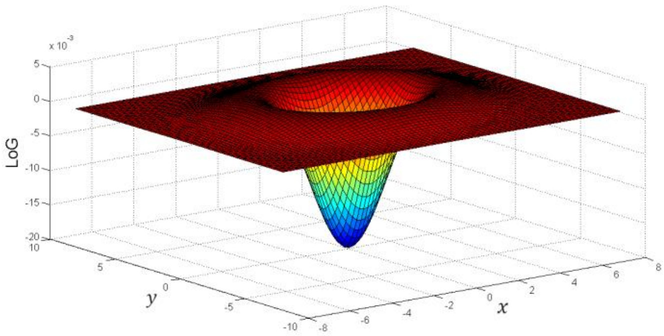


Fig. 2. Laplacian of Gaussian function.

The *LOG* operator calculates the second spatial derivative of an image. In areas where the image has a constant intensity, the gradient is zero and the response will be zero too. In the vicinity of a change in intensity, the response will be positive just to one side of the edge and negative just to the other side of the edge. In the case of the image used as an example, the result obtained is shown in Fig. 3.

Fig. 3. Edge detection.

Once the edge filter has been applied, the areas of the image that do not exceed a certain size are eliminated to avoid possible confusion with the environment. This is done because traffic signs are quite big objects so small edges correspond to objects that cannot be regions of interest. Figure 4 shows the ROIs for the image taken as example. In this figure, every ROI must be evaluated separately.

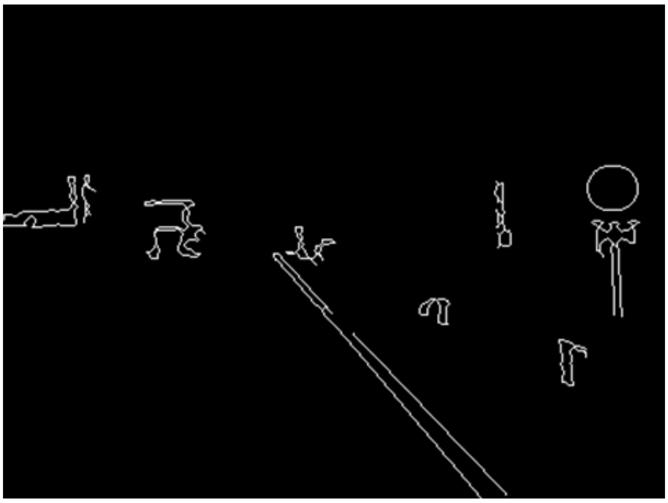


Fig. 4. Regions of interest.

Fig. 5. Contour of the ROI, its centroid and the starting point.

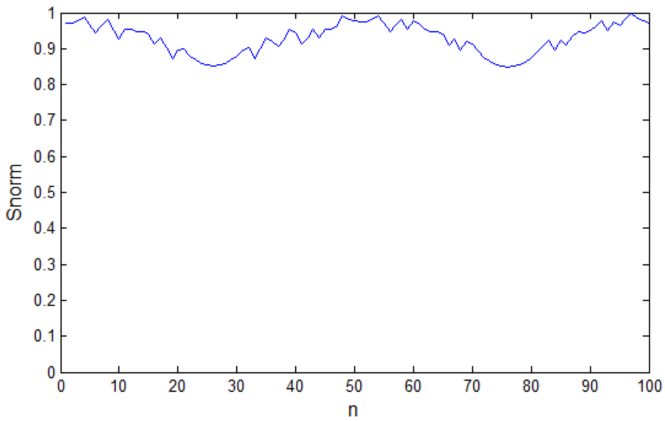


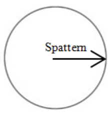
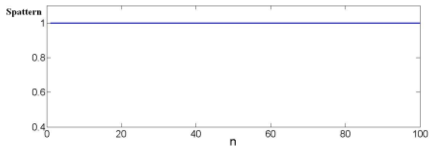
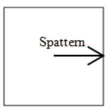
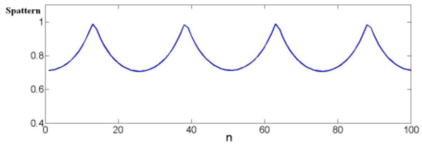
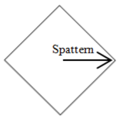
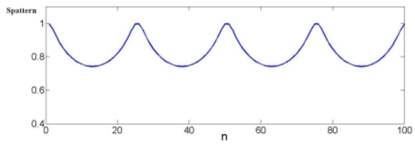
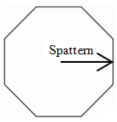
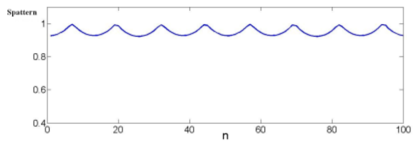
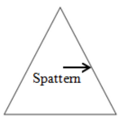
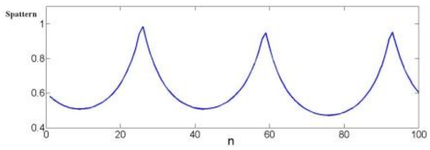
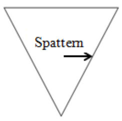
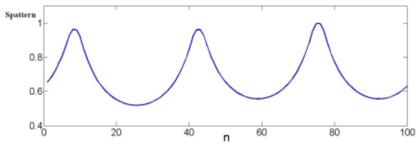
Fig. 6. Normalized signature of the ROI.

There are many types of traffic signs, but their shapes are very limited and characteristic, so knowing the shape allows us to determine if an ROI contains a potential traffic sign or not. In order to do so, the signature of every ROI must be calculated and normalized in order to be compared with the normalized signatures of the shape patterns, such as squares, circles, etc. To obtain the signature, the developed application calculates the contour of every ROI, its centroid and the closest point to the upper right corner of the image, as Fig. 5 shows. Then the distance between the centroid and the perimeter is calculated and the signature is obtained. This signature is normalized, i.e., to fit its value between 0 and 1, so it can be compared with the normalized signatures of the shape patterns. Each normalized signature obtained is stored in a vector of N=100 elements, where n=1,…,N. Using this method, no matter the size of the object evaluated, its signature will be the same. In Fig. 6, the normalized signature *Snorm* obtained for the ROI of Fig. 5 is presented.

3. POTENTIAL TRAFFIC SIGN DETECTION

Once the ROIs have been identified, the normalized signatures of the different shape patterns are loaded. In this application, N_s=6 shape patterns are used: circle, square, rhombus, octagon, triangle, and inverted triangle. The normalized signatures for these six shape patterns are shown in table 1 as *Spattern*. This application allows the user to add new patterns, so special regional traffic signs could also be recognized.

Table 1. Normalized signatures of the shape patterns

Traffic sign patterns	Normalized signature
 Circle	
 Square	
 Rhombus	
 Octagon	
 Triangle	
 Inverted triangle	

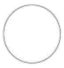
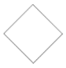
For each ROI, its normalized signature is compared with the normalized signature of each shape pattern by (2), where $Snorm_i$ is each value of the signature of the ROI, $Spattern_{ij}$ is each value of the normalized signature of the pattern j, i=1,...,N with N=100 and j=1,…,N_s with N_s=6.

$$D_j = \sum_{i=1}^{N} \left| Snorm_i - Spattern_{ij} \right| \tag{2}$$

If $min(D_j)<th_{shape}$, where th_{shape} is a threshold value, then this ROI is in fact a potential traffic sign, which has the same shape of the pattern j. In this case, the coordinates of this ROI are saved in a cell array, along with the type of shape that it contains. Then, the application continues with another ROI. In this application, th_{shape}=6 and has been calculated empirically to minimize the error. On the other hand, if $min(D_j)\geq th_{shape}$, then the ROI is not a potential traffic sign and the application continues with another ROI. When all ROIs have been checked and some potential traffic sign has been found, the application continues with the traffic sign recognition process.

Table 2 presents the values of D_j, obtained for the ROI of Fig. 5, and these values show that the ROI is in fact a potential traffic sign with a circular shape.

Table 2. Values of D_j

Spattern						
D_j	4.29	13.84	12.55	6.99	31.17	26.18

4. TRAFFIC SIGN RECOGNITION

Once the location and shape of all potential traffic signs in the image are known, each one of these potential traffic signs must be compared with a database. The database contains a certain number of traffic sign patterns, which are classified according to their shape. Due to the fact that the comparison process is iterative, the processing time of the application increases when more patterns of traffic signs are added. In this sense, the application allows the user to add new patterns. However, this would increase the processing time of the whole process significantly.

Each potential traffic sign is compared with each traffic sign pattern using the cross-correlation algorithm given by (3), where f is the cell array corresponding to the potential traffic sign and t_k is the cell array of the traffic sign pattern k (Lewis, 1995).

$$\gamma_k(u,v) = \frac{\sum_{x,y}[f(x,y)-\bar{f}_{u,v}][t_k(x-u,y-v)-\bar{t}_k]}{\left\{\sum_{x,y}[f(x,y)-\bar{f}_{u,v}]^2 \sum_{x,y}[t_k(x-u,y-v)-\bar{t}_k]^2\right\}^{0.5}} \tag{3}$$

The matrix γ_k obtained contains the correlation coefficients, whose values vary from -1.0 to 1.0. The peaks of the cross-correlation matrix occur where the images are best correlated. Both cell arrays are compared using the correlation coefficient R_k of γ_k given by (4), where A is the traffic sign image resized to the size of the pattern, B_k is the image of the pattern k, and $\bar{A}$, $\bar{B}_k$ are the mean of all the elements in A and B_k respectively.

$$R_k = \frac{\sum_m \sum_n (A_{mn}-\bar{A})(B_{k,mn}-\bar{B}_k)}{\sqrt{\left(\sum_m \sum_n (A_{mn}-\bar{A})^2\right)\left(\sum_m \sum_n \left(B_{k,mn}-\bar{B}_k\right)^2\right)}} \tag{4}$$

If $R_k>th_{sign}$, where th_{sign} is a threshold value, then the potential traffic sign can be considered very similar to the traffic sign pattern. So the application would give the traffic sign of the pattern k as the recognized sign. In this application th_{sign}=0.51 and has been calculated empirically.

Table 3. Cross-correlation matrix for different patterns of circular traffic signs

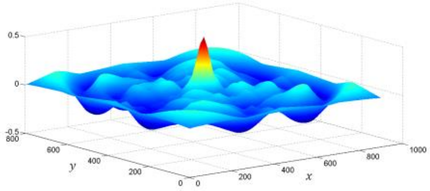
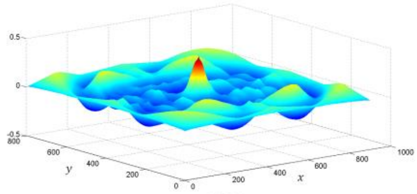
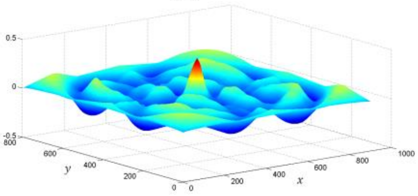
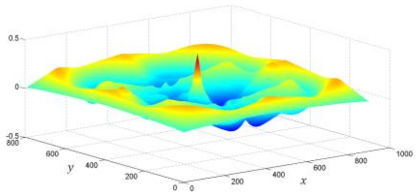
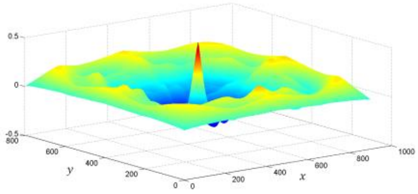
Database sign	γ_k	R_k
		0.5723
		0.3855
		0.3807
		0.4212
		0.5222

Table 3 presents the cross-correlation matrixes γ_k and their corresponding correlation coefficient R_k obtained for the potential traffic sign of Fig. 5 with k=1,…5. The values of the correlation coefficient of the table show that two values are higher than th_{sign}, and in these cases, the highest is taken for recognition purpose. So the application recognizes the potential traffic sign as the traffic sign pattern shown in Fig. 7

and we can see that the traffic sign has been successfully recognized.

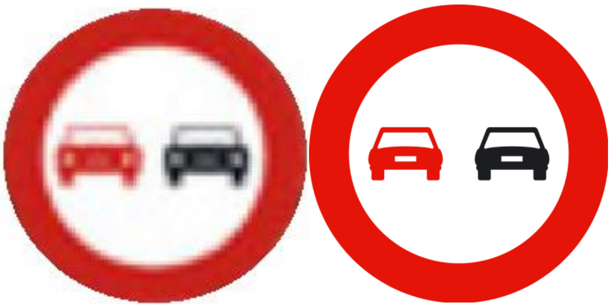


Fig. 7. Potential traffic signal and traffic signal pattern.

If the potential traffic sign does not match any of the traffic sign patterns, the object has the shape of a traffic sign, and therefore its signature, but is not a traffic sign. It should be noted that if the correlation coefficients of the cross-correlation matrix are lower than the threshold, it could be due to the fact that the signal pattern is not present in the pattern database.

5. GRAPHICAL USER INTERFACE DESIGN

In order to improve visualization of the application and make it interactive for a user, a graphical user interface has been designed. It allows the user to see graphically the whole process executed by the application and to modify the input parameters of the system. The code of the application is structured and separated in functions, so it can be handled by the dynamic interface. The graphical user interface actually consists of two stages. The first interface, as Fig. 8 shows, acts as a title page and allows the user to select dynamically the directory where the images are stored, as well as their extension. When the continue button is pressed, all the patterns, both traffic sign shapes and traffic signs, are loaded into the memory so they do not need to be loaded again, significantly reducing the processing time of the application.

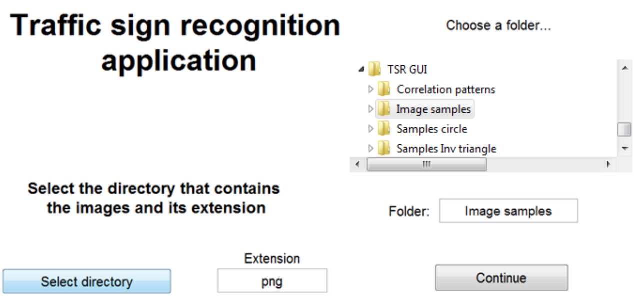


Fig. 8. First interface.

The second interface, as Fig. 9 shows, contains a series of buttons that allow the user to select the image to evaluate from all the images contained in the previously selected directory. The application allows the user to visualize the input image, its histogram and the ROIs that are to be evaluated. Through the button *detect signs,* the previously described algorithms are executed and the results are displayed. If more than one sign was detected, they would be shown at the bottom right in the *Signs recognized* section.

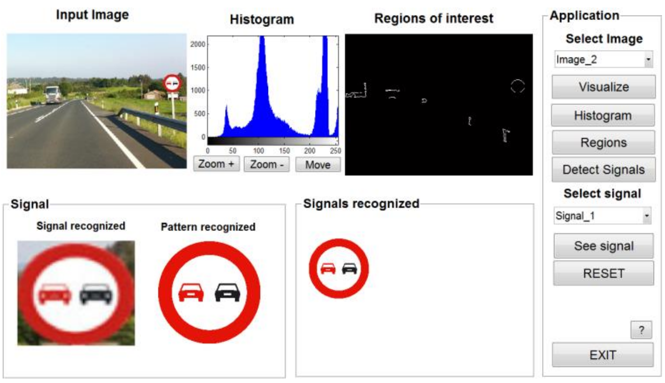


Fig. 9. Second interface.

6. EXPERIMENTAL RESULTS

The application has been tested with 200 different images, where 25 images do not contain any traffic sign, 141 images contain one traffic sign, 22 images contain two traffic signs and 12 images contain three traffic signs. The total number of shapes analyzed is 221 and the distribution is shown in Fig. 10.

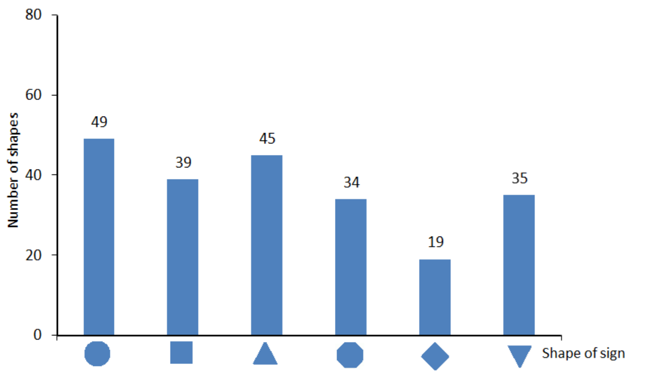


Fig. 10. Traffic sign shapes contained in the 200 test images.

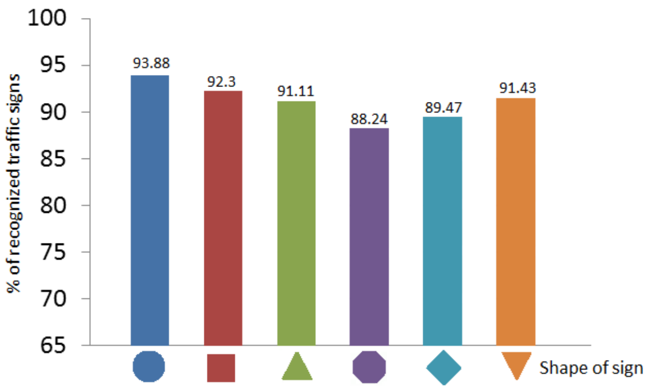


Fig. 11. Shape recognition hit rate in 200 test images.

The average success rate is 91.07%, the result being independent of the number of traffic signs that the image contains. The success rate for each type of shape is shown in Fig. 11. The lower rate of success in octagon shaped signals with 88.24%, is due to the fact that in certain images the signature of the octagon is recognized as a circle, so the detected sign is incorrect.

There are recognition problems in low-resolution images as, due to the low pixel rate, the recognized the areas are joined together, resulting in a detection of erroneous ROIs. Furthermore, very high resolution images, due to the weight in memory, must be previously treated, reducing the storage weight before loading them, because the memory space is problematic and cannot be properly processed by the program. On the other hand, in pictures showing cuts in their area or perimeter due to external agents such as vegetation, climate, etc., the application had difficulty recognizing the area and, in many cases, the area was ruled out due to the fact that it did not have a closed surface.

7. CONCLUSIONS

A new traffic sign recognition system has been presented in this paper. The application software developed in this work recognizes and classifies traffic signs from an input image. The image processing techniques used in this software include a preprocessing stage, regions of interest detection, potential traffic sign detection, according to the traffic sign shape patterns, and finally, the recognition and classification of these potential traffic signs according to a database of traffic sign patterns. The performance of this application depends on the quality of the input image, in relation to its size, contrast and the way the signs appear in the image. With this consideration, the percentages of recognized signs for this application are high.

As further work, a neural network could be implemented in order to obtain more accurately the empirical parameters used in the application. Furthermore, the application could be optimized by implementing an embedded hardware for use in active applications.

REFERENCES

Chen, C.H., Pau, L.F., and Wang, P.S.P. (1993). *Handbook of Pattern Recognition and Computer Vision*. World Scientific Publishing, Singapore.

Dougherty, G. (2009). *Digital Image Processing for Medical Applications.* Cambridge University Press, Cambridge.

Gil, P., Maldonado, S., Gómez, H., Lafuente, S., and López, F. (2008). Traffic sign shape classification and localization based on the normalized FFT of the signature of blobs and 2D homographies. *Signal Processing*, 88(12), 2943–2955.

González, R.C., and Woods, R.E. (2007). *Digital image processing, 3rd. Ed*. Prentice Hall, New Jersey.

Kastrinaki, V., Zervakis, and M., Kalaitzakis, K. (2003). A survey of video processing techniques for traffic applications. *Image and Vision Computing*, 21(4), 359–381.

Lewis, J.P. (1995). Fast Normalized Cross-Correlation. *Vision Interface*, 10, 120-123.

Mapanga, K., and Ragavan, V. (2012). Machine Vision for intelligent Semi-autonomous Transport (MV-iSAT). *Procedia Engineering*, 41, 395-404.

Pei, S.-C., and Horng, J.-H. (2002). Design of FIR bilevel Laplacian-of-Gaussian filter. *Signal Processing*, 82(4), 677-691.

Piccioli, G., De Micheli, E., Parodi, P., and Campani, M. (1996). Robust method for road sign detection and recognition. *Image and Vision Computing*, 14(3), 209-223.

Pizer, S.M., Amburn, E.P., Austin, J.D., Cromartie, R., Geselowitz, A., Greer, T., Romeny, B., Zimmerman, J.B. and Zuiderveld, K. (1987). Adaptive histogram equalization and its variations. *Computer Vision, Graphics, and Image Processing,* 39(3), 355-368.

Reza, A.M. (2004). Realization of the Contrast Limited Adaptive Histogram Equalization (CLAHE) for Real-Time Image Enhancement. *Journal of VLSI Signal Processing Systems for signal, image and video technology*, 38(1), 35-44.

Ritter, W., Stein, F., and Janssen, R. (1995). Traffic sign recognition using colour informaion. *Mathematical and Computer Modelling*, 22(4-7), 149-157, 159-161.

Ruta, A., Li, Y., and Liu, X. (2010). Real-time traffic sign recognition from video by class-specific discriminative features. *Pattern Recognition*, 43(1), 416-430.

Sepasian, M., Balachandran, W., and Mares, C. (2008). Image Enhancement for Fingerprint Minutiae-Based Algorithms Using CLAHE, Standard Deviation Analysis and Sliding Neighborhood. In International Association of Engineers (Ed.), *Proceedings of the World Congress on Engineering and Computer Science WCECS 2008*. 1199-1203. San Francisco. USA.

Sharifi, M., Fathy, M., and Tayefeh, M. (2002). A classified and comparative study of edge detection algorithms. In. *Proceedings of the International Conference on Information Technology: Coding and Computing*. 117-120.

Wan, J., He, X., and Shi, P. (2007). An Iris Image Quality Assessment Method Based on Laplacian of Gaussian Operation, In *Proceedings of the IAPR Conference on Machine Vision Applications IAPR MVA 2007*. 248-251. Tokio, Japan.