New image processing algorithms for traffic signs recognition

Jarnea Alexandru Daniel, Florea Gheorghe and Dobrescu Radu

Abstract— The paper presents two novel algorithms for visual information procesing in an advanced driver assistance system of intelligent vehicles. Our approach focus on the two most important issues in these research area, the detection and recognition of traffic signs and the lane detection. The experimental results were made on images captured in real traffic and and demonstrate that the algorithms can successfully reach their goals with close to real time accuracy and with performance.

Keywords— image processing, parallel threading, particle analysis, region of interest, traffic sign recognition, lane detection

I. INTRODUCTION

The task to develop intelligent vehicles is not a new one and can be seen in any assistance systems available today. With the latest evolution of the computing power we can process the data faster. We can develop new, more reliable algorithms that can be based as much as possible on visual information.

Driving is based mostly on processing the useful visual information. The drivers take the appropriate actions from the data acquired by recognizing traffic signs, road lanes, virtual distances etc. Artificial systems still have some unresolved subjects when straight view to the useful information is obstructed. By respecting all the indicators' information the driver can adapt to any type of road even if he travels on this route for the first time.

An intelligent vehicle is based on advanced driver assistance systems (ADAS) that can autonomously intervene if a possible dangerous situation is detected. The researchers try to develop a system that can analyze and understand the visual based information with different approaches.

An ADAS can improve driving safety and can enhance the driver's comfort. For instance, it can inform the driver about the recommended actions or it can prevent him from doing inappropriate maneuvers.

Ing Jarnea Alexandru Daniel is PhD. student, Faculty of Automatic Control and Computers, University POLITEHNICA of Bucharest, Romania, e-mail: jarrnea@yahoo.com,

Ing Gheorghe Florea is PhD. student, Faculty of Automatic Control and Computers, University POLITEHNICA of Bucharest, Romania, e-mail: gelu.florea@sis.ro

Prof. dr. ing. Dobrescu Radu, Faculty of Automatic Control and Computers, University POLITEHNICA of Bucharest, Romania, e-mail: rd_dobrescu@yahoo.com

In usual atmospheric conditions road lanes or traffic signs are easily distinguished, but during harsh conditions they become difficult to recognize as outlined in [1]. A traffic sign recognition system monitors a complex and ever changing environment and must do so accurately and continuously. Here in lies the challenges for this type of system: complex environment, expectation of accuracy, and short response time. Essentially it must identify the road signs that are in view in real time.

The paper is structured as follows. In section II is presented a critical analysis of recent achievements of scientific research in the field of traffic signs detection, tracking and recognition. Section III discusses the challenges encountered for building traffic sign detection and recognition systems and briefly describes the procedure adopted in our work to improve sign recognition and classification. Section IV describes the characteristics of the proposed software architecture which support the main image processing algoritms developed for traffic sign recognition and lane detection, respectively. In section V we present the results obtained form the developed algorithms. In the final section VI we analyze the research and provide future directions from this range.

II. RELATED WORK

Recent work has been focused on detection, classification and tracking of relevant features, as it is presented in [2]. The article also talks about an ADAS that can map the important features from the environment using a CCD camera and vertical laser sensors.

Paper [3] outline the ADAS that are implemented today as platooning, stop and go, blind angle perception, maneuvers and they present a vehicle detection based on a geometrical model.

In [4] a road lane detection algorithm is explained, which is based on inverse perspective mapping. Also summarized in [5] is a simplified Markov model which has been made to operate even in cases where markings are obstructed or absent altogether.

A traffic lights recognition algorithm is proposed in [6]. A robust detection and tracking algorithm is achieved to avoid false positive results from brake lights.

A non-rigid objects detection as a pedestrian algorithm is presented in [7] where they use an infrared image and good results are obtained for a single frame performance. A deformable contour model for pedestrian detection with very good result is presented in [8] where frames from a stereo vision camera are used.

The approach that earned an important part of the researcher's attention is the object detection application developed by Paul Viola and Michael Jones [9]. Their approach, helped by the learning technique, uses cascade detectors to discriminate objects from image areas where every detector is a set of efficient computable Haar features.

A machine learning algorithm is presented by Yoav Freund and Robert Schafire in [10], named short Adaboost, is embedded in [11] as a faster approach and with better results extension of the original Haar Wavelets over large feature sets.

In [12] is presented a traffic sign recognition approach that is based on Adaboost and Forest Error-Correcting Output Code and they obtain a very high percentage at detecting circular traffic signs. The approach of color segmentation is studied in [13] where they too obtain a good segmentation method, but they don't achieve as good of a recognition percentage.

A procedure for analysis of topological structure and spatial characteristics of a whole road section of an urban transport network which uses real time video information is discussed in [14]. With the same goal to control competing flows of urban traffic in [15] is discussed a method in controlling the traffic light system placed on a certain junction, capable of counting the total number of vehicles entering and exiting from this junction on real time basis.

The development of an image-based lane departure warning (LDW) system to detect when the driver is in danger of departing the road and to alarm the driver early enough to take corrective action is presented in [16]. This system uses the Lucas-Kanade (L-K) optical flow for lane tracking and the Hough transform method for lane detection.

III. Improved Procedure for Sign Recognition

Road signs are an inherent part of the traffic environment. They are designed to regulate low of the vehicles, give specific information, or warn against unexpected road circumstances. For that reason, perception and fast interpretation of signs is critical for the driver's safety. Therefore, when video processing be came attainable on a computer machine, automation of the road sign detection and recognition process was found a natural direction to follow.

Recognition of traffic signs is a hard multi-class problem with an additional difficulty caused by the fact of certain signs being very similar to one another, but also other major problems exist in the whole detection process. Road signs are frequently occluded partially by other vehicles. Many objects are present in traffic scenes which make the sign detection hard (pedestrians, other vehicles, buildings and billboards may confuse the detection system by patterns similar to that of road signs). Colour information from traffic scene images is affected by varying illumination caused by weather conditions, time (day/night) and shadowing (buildings).

Traffic sign recognition is part of the general case of Pattern Recognition. Major problem in pattern recognition is the difficulty of constructing characteristic patterns (templates). Unlike the general case when the detection of certain pattern in the image is hampered by the large variety of the features to be searched in the images, traffic signs on the contrary: 1) are made with vivid and specific colors so as to attract the driver's attention and to be distinguished from the environment; 2) are of specific geometrical shapes (triangle, rectangle, circle-ellipse) and 3) for each sign there is a specific template. It is therefore rather easy to develop an algorithm in such a way that the computer has "*a priori*" knowledge of the objects being searched in the image.

The problem of identifying traffic signs within an image can be broken into the two sub-problems of detection and recognition. Detection presents the challenge of analyzing the image to identify portions of the image that could contain a traffic sign. Recognition is the challenge of determining if these candidates are indeed traffic signs and if so which one. The developed algorithm (detailed in section IV) is divided in two basic phases, each one composed of a certain number of steps. In the first phase the detection of the location of the sign in the image, based on its geometrical characteristics and of his colour information. The second phase is the sign recognition with the matching between the search image and the template images, already stored in a database, according to different categories and classes.

The approach we have adopted assumes the existence of a single separator between each class and all other classes. It is implemented using a winner-takes-all strategy that associates a real-valued score with each class. An example belongs to the class which assigns it the highest score.

Formally the classifier, $F(x)$ recognizes only two classes, namely "same" (S) and "different" (D), and is trained using pairs of images, i.e. $\boldsymbol{x} = (i_1, i_2)$. The pairs representing the same sign type are labeled $y = 1$ (positive), and the pairs representing two different types are labeled $y = -1$ (negative). We define F as a sum of image features f_j :

$$F(i_1, i_2) = \sum_{j=1}^{N} f_j(i_1, i_2) \tag{1}$$

Each feature evaluates to:

$$f_j(i_1, i_2) = \begin{cases} \alpha & if \quad d(\varphi_j(i_1), \varphi_j(i_2)) < t_j \\ & \beta \quad otherwise \end{cases} \tag{2}$$

where φ_j is a filter defined over a chosen class of image descriptors, d is a generic distance metric that makes sense for such descriptors, and t_j is a feature threshold.

The classification decision is given by the equation:

$$l(i_1, i_2) = \operatorname{sgn} F(i_1, i_2) \tag{3}$$

By omitting sign, value of the right-hand-side term can be treated as a degree of similarity of the input images. If one of those images, say i_1, is a prototype of known class k ($i_1 = p_k$), our road sign classifier assigns such a label to the other, unknown image, that satisfies:

$$l(i) = \arg\max F(p_k, i) \qquad (4)$$

In other words, *l(i)* is determined from the prototype to which the tested image is the most similar.

IV. New Visual Information Processing Algorithms For Intelligent Vehicles

Fig.1. illustrates the architecture of the driving assistance system, with mention of the main routines used for visual information processing, which will be presented in detail in this section.

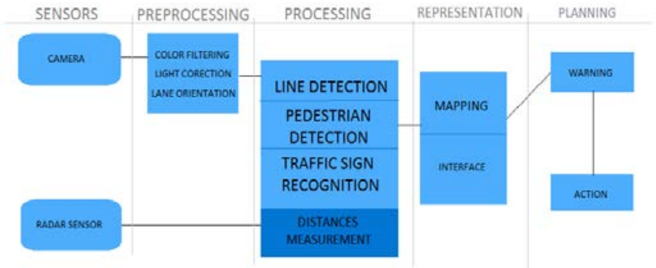


Figure.1 Architecture of our driving assistance system

Such a system based on image processing and important features detection should be based on traffic sings recognition, lane detection and obstacles detection. We present in the following our approach for traffic sign recognition and lane detection algorithms.

A. Traffic sign recognition algorithm

The program that we developed is written as a windows forms application in C# enviroment.

At this stage of development the program can run on a computer running the Windows operating system. The application may run on iPhone, Android, Linux or other operating systems if it is compiled with a cross platform application.

The Emgu CV library, which is described in [17], is used for image processing. This library adds some facilities to the application such as image class with generic color and depth, generic operations on image pixels. The most important feature of this library is that we can use video hardware processing to improve the timings of our algorithms.

The images that we use are captured, while driving, by a Galaxy S3 phone at HD resolution. Every frame is resized to a 640x480 pixels resolution before starting any analysis.

The library signs are acquired automatically from a folder placed next to the application executable file. We used a library of 8 restriction signs, however, any number of templates can be added if the computer platform can process them in a small amount of time.

In Fig.2 we can see the restriction and alert signs that are considered as library for these algorithms.

Figure.2 Signs from the library taken into consideration

The algorithm that we present here is based on common traffic sign recognition that usually has tree steps: segmentation, detection and recognition, as you can see from Fig.3. We chose to make the segmentation step permissive, with wider windows color then usual, so we can get all regions of interest where a sign could be found. We added an additional step, after the detection, where according to the light average of the image the initial ROI's are transformed into black and white ones by adapting the filter with corresponding variables.

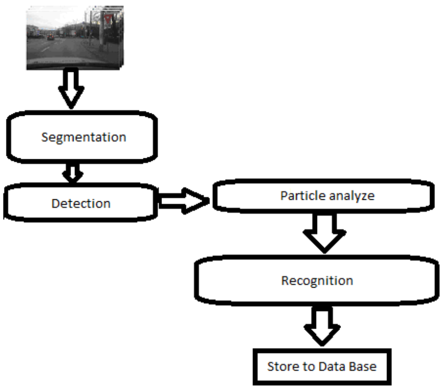


Figure.3 Structure of the traffic sign recognition algorithm

The application has two operating modes, the first one is a very detailed one which shows the correlation vlaues for the regions of interest with all the library signs. The second one has the same base algorithm but is faster, because it is implemented with parallel threrads.

a. Segmentation

The segmentation step starts with converting from RGB to HSL the query frame extracted from video. From the HSL matrix we extract the areas with red, blue, and yelow window color between set values.

To extract from the HSL matrix a black white image, which highlights the exact dimension and position where a sign could be. In the application it is used for example for red color the window that can be seen in equ. (1). The small white portions from inside the sign perimeter are filled in. You can see in Fig.5a the result obtained by applying to Fig.4 the procedure described by equ. (5).

$$\operatorname{Re} d(i,j) = \begin{cases} true, \quad if \quad ((0 < Frame(:,:,1) < 8) or (112 < Frame(:,:,1) < 5) \\ and \quad (25 < Frame(:,:,3) < 230)) \\ \qquad\qquad\qquad false, else \end{cases} \qquad (5)$$

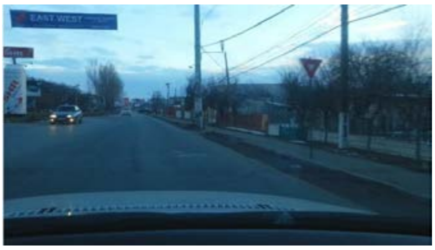
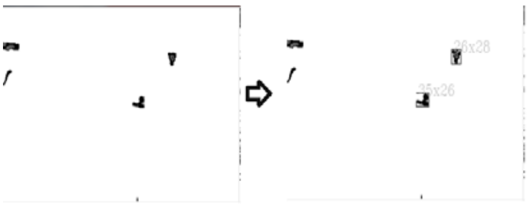


Figure.4 Video frame



a. b.

Figure.5 (a) Extracted posible areas of interest from Fig.3, for red color. (b) Outlined regions of interest, with measured square dimensions.

b. Detection of all the regions of interest

On the image from Fig.5a that we extracted at the first step we draw a rectangle as a contour for every black area. We measure the exact dimension of all distinct regions as can be seen in Fig.5b where we have just one outlined area, which in this case has the size of 26x28 pixels.

The regions that have the width and height higher than 20 pixels and less than 95 pixels are marked as regions of interest and are further anlyzed as posible regions where a sign could be present. These tresholds are also setted from measurements and experimental results, but can be easily modiffied from the application interface.

c. Particle replacement and reanalysis

At this step we calculate the average light in the initial frame. For every detected region of interest like the one from Fig.5b we overlap on the black areas the corespondent pixels from initial frame Fig.3. We convert to black and white the region of interest considering the initial light intensity.

d. Recognition

For this step we compare the regions of interest to the signs from the library after resizing the library sign to the dimensions of the region of interest.

If the correlation between a region of interest and a library sign is greater then 0.70, the region is found as a sign from the library. If the correlation is greater than 0.5 and smaller than 0.70 the region of interest is considered as posible presence of the sign from library sign.

We can see that the correlation between Fig.6a and Fig.6b, as one can see in Fig.6c, is 0.8, where 1 is perfect correlation, so we have a higher correlation value than the treshold and the sign is considered found.

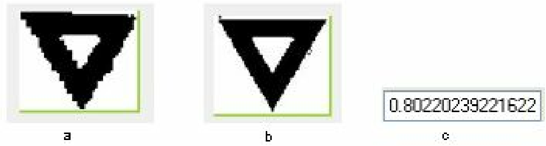


Figure.6 (a) Region of interest filtered in black and white. (b) Library sign filtered in black and white. (c) The correlation after a and b comparison

e. Save to database

Every region that passes the restrictions is compared with the library signs from this color cathegory and gets a correlation percentage for every library sign. If one of the region have the correlation higer than 0.7 for a library sign it is labeled as a match. For the regions with correlation higher than 40% we store in a data base the image name, the extracted region of interest, the name of the sign from the library and the GPS location of the sign, aquiered from the image data.

By analyzing the HSL matrix we avoid most of the problems that occur when filtering the RGB matrix after setting the window limits. With the light analysis, the color window limits will detect the sign even in an environment with shadows on the area where the sign is positioned.

During night time with public lighting we can detect signs without problems, as shown with the “give way” sign in Fig.7. We have a problem when the roads do not have public illumination. In this situation the colors of the sign are visible but the correlation is not high as we would want. This problem could be easily surpassed by using a camera with the new generation of high dynamic rage sensors.

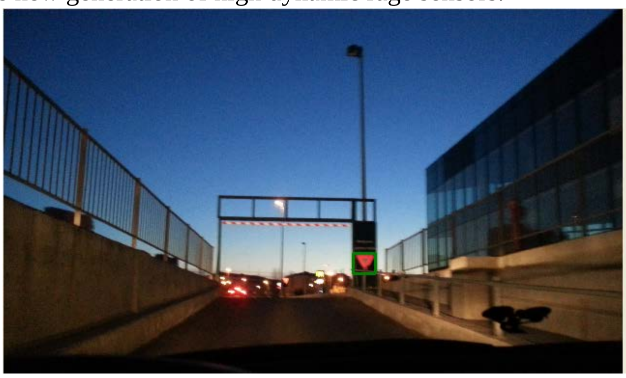


Figure.7 Night frame analyzed by the software

Traffic sign recognition is one of the systems that will help drivers, as covered in [18], and with which they will be more confident in themselves. Therefore these systems need s strong planning and reliability and finally should be one of the basic systems that will sustain self drive vehicles.

B. Lane detection

The algorithm that we have developed does not rely only on the marked lines like in [19], we also took into consideration the roadsides, to achieve better reliability. For example we can warn the driver if he is going to cross the roadside.

In the Fig.8 the structure of the lane detection algorithm can be seen.

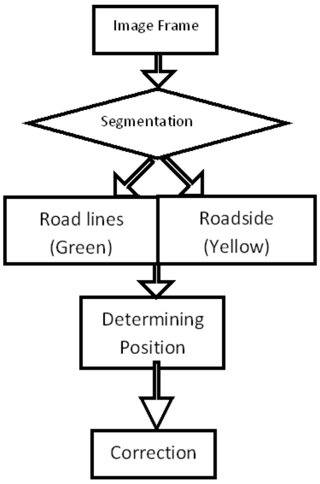


Figure.8 Structure of the lane detection algorithm

We calculate the average of every pixel's neighbors and if the value is grater then a step we mark then we make that pixel white, else black. The step we use is between 3 and 10. We use the minimum step because we also want to extract the road boundaries.

The initial video frame it is presented in Fig.9.

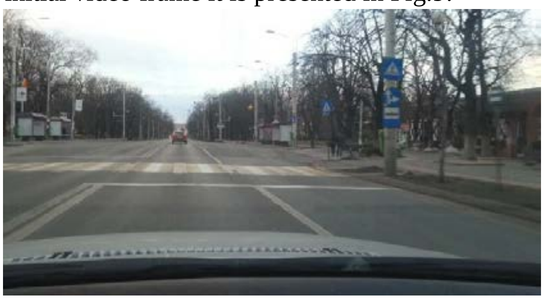


Figure.9 Initial video Frame

The result of this algorithm is a black image with road lanes, or road borders extracted on white color as can be seen in Fig 10.

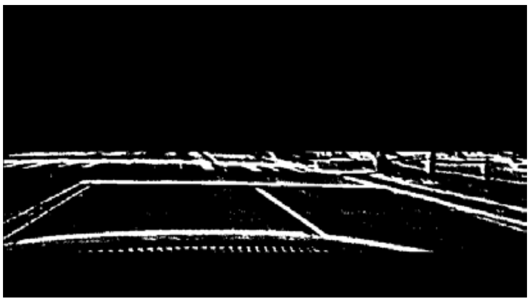


Figure.10 Segmentation of the road lanes

On Fig.10 we apply a Hough transform as presented in [20], and we extract two types of lanes after the angle incidence: first type is colored with green and could be the road lane and the second type of lane is colored with yellow and could represent the road borders and could be used when the road lanes are not visible.

The green lanes are chosen with the angle between 29 and 40 or -29 and -40 radians, and the yellow ones are selected with the angle between 14 and 29 or 40 and 54 radians or -14 and -29 or -40 and -54 radians, as can be seen in Fig 11.

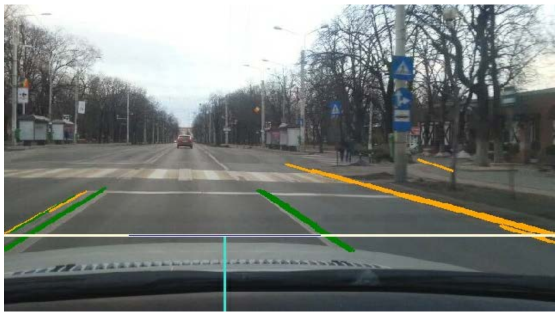


Figure.11 Drowned virtual road lane

We calculate the intersections of the virtual road lanes with the gray lane that you see in Fig 10. We get the closer green or yellow lane from the center of the vehicle from each side of the road.

The steering command is calculated in pixels by making the difference between the most restrictive left and right lanes green or if necessary using the yellow ones. This dimension could be converted in angles to be steered according to a range of parameters like the current speed, camera resolution, vehicle dimensions, road conditions etc.

The vertical red lanes that estimate the generated future vehicle dimension and position can be seen in Fig.12. The actual direction without correction is drawn with green vertical lanes.

Figure.12 Generated future position of the vehicle based on road lane middle

V. RESULTS

A. Traffic sign recognition

With the fast approach we can process from 15 to 30 frames per second according to the complexity of the image and the CPU of the computer that processes the data. If we compare with [21] where they used an i7-920 processor, 12GB of ram and 4 GTX 580 graphic cards for 25 frames per second this approach is faster because we used an AMD X2 250 with 4GB of ram and without video hardware we can process a frame within an average time of 45 milliseconds. We have to take into consideration that they used 43 library signs and in our approach only 8 signs are analyzed.

The result of the percentage of detected and recognized signs for each different category of color is presented in table I.

TABLE I. PERCENT OF DETECTION

	Red signs	Blue signs	Yellow signs
Frames	50	50	50
Detected	98%	98%	96%
Recognized	96%	96%	94%
Average processing time	44ms	46 ms	32ms

In Fig.13 one can see the representative frame of each set of signs.

Figure.13 Frames from each testing set

In Fig.14 one can see a chart which represents the results from table I.

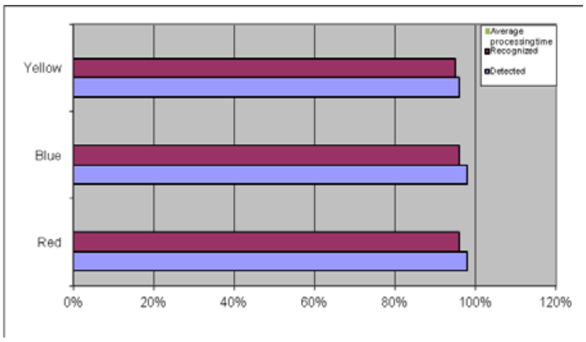


Figure.14 Frames from each testing set

The results from table II are obtained after analyzing different category of signs in foggy conditions with the same system variables set on normal weather conditions.

TABLE II. TRAFFIC SIGN RECOGNITION ON FOGGY CONDITIONS

	Red signs	Blue signs	Yellow signs
Frames	20	20	20
Detected	75%	81%	65%
Recognized	70%	75%	61%
Average processing time	35ms	35 ms	24ms

In Fig.15 can be seen a chart which represents the results from table II.

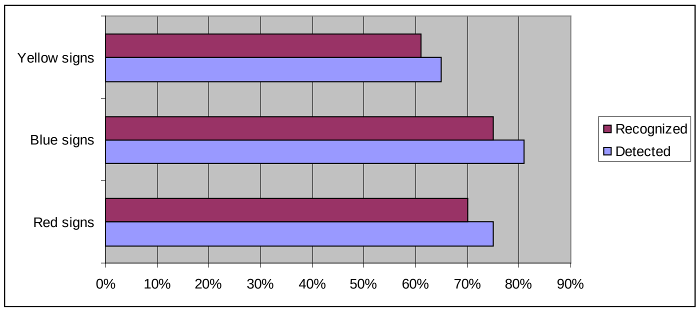


Figure.15 Percent of detection in foggy conditions

The results from table III represents an analysis of blue signs that have close comparison percentage.

TABLE III. SIGNS THAT GAVE HIGH COMPARISON RESULT

	-	0.5	0.5	0.4	0.2	0.5	0.4	0.6	0.6
	0.5	-	0.5	0.5	0.4	0.5	0.4	0.6	0.5
	0.5	0.5	-	0.5	0.2	0.4	0.5	0.5	0.4
	0.4	0.5	0.5	-	0.3	0.5	0.5	0.6	0.4
	0.2	0.4	0.2	0.3	-	0.2	0.2	0.2	0.4
	0.5	0.5	0.4	0.5	0.2	-	0.4	0.6	0.4
40	0.4	0.4	0.5	0.5	0.2	0.4	-	0.5	0.5
	0.6	0.6	0.5	0.6	0.2	0.6	0.5	-	0.4
	0.6	0.5	0.4	0.4	0.4	0.4	0.5	0.4	-

B. Lane detection

The lane detection method looks for the lane markers painted on each side of the road and if available, the system calculates the distance to it and compares that to a past value to determine where the vehicle is in the lane and in which direction it's headed.

In table III you can see the results that we had on 4 different types of road during the day time while detecting road lanes.

TABLE IV. Lane detection on day light at 640x480 pixels

	Frames	Correct	Incorrect	Accuracy
No lanes	34	32	2	93.75%
One lane	25	25	0	100%
Multi lanes	37	36	1	97.29%
Highway	15	15	0	100%

In Fig.16 you can see the direction correction of lane detection algorithm on different types of roads.

Figure.16 Frames with virtual lanes generated

In Fig.17 can be seen the data results from a road line analysis in night conditions.

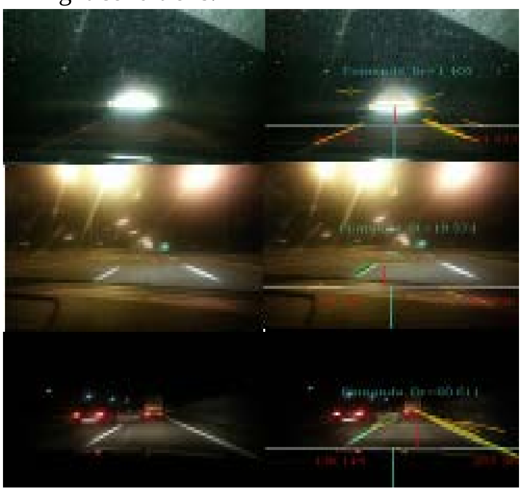


Figure.17 Night frames with virtual lanes generated

In Fig.18 can be seen the data results from a road line analysis in rainy day conditions.

Figure.18 Rainy day frames with virtual lanes generated

In table V one can see the results that we had on different road conditions.

TABLE V. Lane detection on different conditions with one or two lines

	Frames	Correct	Incorrect	Accuracy
Day normal conditions	25	25	0	100%
Night normal conditions	15	15	0	100%
Day rainy conditions	15	14	1	93%

In Fig.19 one can see a chart which represents the result from table V.

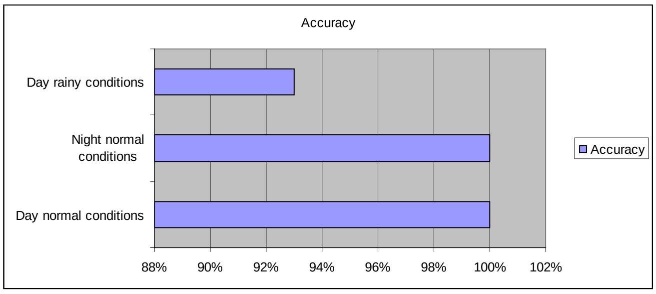


Figure.19 Graphic of line analyse results on different road contitions

VI. Conclusions

In this paper, we described our algorithms for visual information support for driving an intelligent vehicle. We aimed to have a very good processing time while detecting all the goals. A detected sign candidate is classified by maximizing its similarity to the class prototype image. This similarity is estimated by a linear combination of local image similarities. Robustness of the traffic sign detector, tracker and classifier is demonstrated in the static and dynamic recognition experiments. In the latter, our prototype implementation is shown to capture and correctly classify most road signs in real time.

We studied different methods of segmentation and some are implemented in our algorithm, though has passed to various modifications. Best traffic sign recognition programs Today reach up to 99% correct detection, but in harsh weather conditions, not uncommon the visual recognition is not able to do the task. The database mapping and validation of the traffic signs is probably the best solution that can be implemented today because in difficult environmental situations the signs will not be recognized.

The biggest problem encountered is that the public sign database is it very poor for Romanian traffic signs. We have the results from a new data set created for our traffic sign recognition algorithm. This aspect is making the comparison between papers very difficult.

The experimental results were made on images captured from actual Romanian roads and demonstrate that the algorithm can successfully reach its goal with close to real time accuracy and with performance.

As future work we strive to improve the correlation percentage and add tracking steps to our algorithms to optimize the processing time. We also want to add an interpretation step and detailed description for the developed algorithms to approach ADAS as much as possible.

REFERENCES

[1] S. Waite, and E. Oruklu, “FPGA-Based Traffic Sign Recognition for Advanced Driver Assistance Systems”, *Journal of Transportation Technologies*, 3, 2013, pp.1-16
[2] B. Hoferlin, K. Zimmermann, “Towards reliable Traffic Sign Recognition”, *IEEE Intelligent Vehicles Symposium*, 2009, pp.324-329
[3] J. M. Collado, C. Hilario, A. de la Escalera and J. M. Armingol, “Model Based Vehicle Detection for Intelligent Vehicles”, *IEEE Intelligent Vehicles Symposium*, 2004, pp.572-577
[4] S. Sehestedt, S. Kodagoda, A. Alempijevic, and G. Dissanayake, “Efficient Lane Detection and Tracking in Urban Environments”, *Proc. European Conf. Mobile Robots*, 2007, pp.126-131
[5] S. Sehestedt, S. Kodagoda, A. Alempijevic, G. Dissanayake, “Robust Lane Detection in Urban Environments”, *Intelligent Robots and Systems*, 2007, pp.123-128
[6] J. Gong, Y. Jiang, G. Xiong, C. Guan, G.Tao, and H. Chen, “The Recognition and Tracking of Traffic Lights Based on Color Segmentation and CAMSHIFT for Intelligent Vehicles”, *IEEE Intelligent Vehicles Symposium*, 2010, pp.431-435
[7] F. Suard, A. Rakotomamonjy, A. Bensrhair, and A. Broggi, “Pedestrian Detection using Infrared images and Histograms of Oriented Gradients”, *IEEE Intelligent Vehicles Symposium*, 2006, pp.206-212
[8] C. Hilario, J. M. Collado, J. M. Armingol, and A. de la Escalera, “Pedestrian Detection for Intelligent Vehicles based on Active Contour Models and Stereo Vision”, *Computer Aided Systems Theory*, EUROCAST, 2005, 537-542
[9] P. Viola, and M. Jones, “Rapid Object Detection using a Boosted Cascade of Simple Features”, *Conference on Computer Vision and Pattern Recognition*, 2001, pp. 511-518
[10] Y. Freund, and R. E. Schapire. „A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting”, *Journal of Computer and System Sciences*, 1997, pp.119-139
[11] R. Lienhart, and J. Maydt, “An extended set of Haar-like features for rapid object detection”, *IEEE Image Processing*, 2002, pp. 900-903
[12] X. Baró, S. Escalera, J. Vitrià, O. Pujol, P. Radeva, “Traffic Sign Recognition Using Evolutionary Adaboost Detection and Forest-ECOC Classification”, *IEEE Transactions on Intelligent Transportation Systems*, 10 (1), 2009, pp.113-126
[13] H. Gomez-Moreno, S. Maldonado-Bascon, Gill-Jamirez and S. Lafumente-Arroyo, “Goal Evaluation of Segmentation Algorithms for Traffic Sign Recognition”, *IEEE Transactions on Intelligent Transportation Systems*, 11 (4), pp. 917-930
[14] X.-H. Yang, and Y.Z. Zhang, “A Bus Transport Network Model Based On Directed Random Walk”, *WSEAS Transactions on Systems*, vol.13, 2014, pp.252-262
[15] S. K. Subramaniam, M. Esro and F. L. Aw, “Self-Algorithm Traffic Light Controllers for Heavily Congested Urban Route”, *WSEAS Transactions on Systems*, Iss.4, Vol.14, 2012, pp.115-124
[16] G.Taubel, R. Sharma, and J-S. Yang, “An Experimental Study of a Lane Departure Warning System Based onthe Optical Flow and Hough Transform Methods”, *WSEAS Transactions on Systems*, Vol.13, 2014, pp.105-115
[17] S. Shi, *Emgu CV Essentials*, Packt Publishing, 2013
[18] J. Stallkamp, M. Schlipsing, J. Salmen, C. Igel, „The German Traffic Sign Recognition Benchmark: A multi-class classification competition, *Int. Joint Conf, on, Neural Networks*, 2011, pp.1453-1460
[19] M. Revilloud, D. Gruyer, and E. Pollard, „An improved approach for robust road marking detection and tracking applied to multi-lane estimation”, *IEEE Intelligent Vehicles Symposium*, 2013, pp.783-790
[20] O. O. Khalifa, I. M. Khan, and A.M. Abdulhakam, „A Hyperbola-Pair Based Lane Detection System for Vehicle Guidance, Proceedings of the World Congress on Engineering and Computer Science”, 2010,
[21] A. Mogelmose, M. M. Trivedi, and T. B. Moeslund, „Vision-Based Traffic Sign Detection and Analysis for Intelligent Driver Assistance Systems: Perspectives and Survey”, *IEEE Transactions on Intelligent Transportation Systems*, Vol. 13, No. 4, 2012, pp.1484-1497