2014 IEEE International Conference on
Autonomous Robot Systems and Competitions (ICARSC)
May 14-15, Espinho, Portugal

Traffic Sign Recognition for Autonomous Driving Robot

Tiago Moura[(1)], António Valente[(1,2)], António Sousa[(1,2)], Vítor Filipe[(1,2)]

[(1)]Department of Engineering, School of Science and Technology, Universidade de Trás-os-Montes e Alto Douro, Quinta de Prados, 5000 - 801 Vila Real, Portugal.

[(2)]INESC TEC, Campus da FEUP, Rua Dr. Roberto Frias, 378, 4200 - 465 Porto, Portugal.

mav@null.net, {avalente, amrs, vfilipe}@utad.pt

***Abstract* — This paper introduces a fast Traffic Sign Recognition system developed for a robot, participant in the Autonomous Driving Competition in the Portuguese Festival of Robotics. The Autonomous Driving Robot performs detection and classification of traffic signs and traffic lights based on the analysis of images acquired by a camera mounted on its chassis.**

The proposed algorithm is composed of three processing stages: detection, pictogram extraction and classification. After the two firsts processing stages, a binary pattern matrix is obtained by color segmentation. In the classification stage two different neural networks were trained to recognize the traffic signs or the traffic light sign. Experimental results show that the system *precision* is very close to 100% whereas *recall* presents values above 90% in most of the signs. The proposed system also proves to be reliable and suitable for real-time processing.

Keywords* — *traffic sign recognition, neural network, autonomous driving robot.

I. INTRODUCTION

Robotic contests are one of the most promising ways to attract students to the field of robotics, since winning an award at a competition not only gives students a sense of accomplishment but also gives pride and visibility to schools. Autonomous Driving is one of the senior (manly for university students) contests in the National Robotic Festival. The Autonomous Driving Competition is a contest for fully autonomous robots that takes place in a track with the shape of a traffic road, surrounded by two parallel side lines and including two lanes separated by a dashed mid line, depicted on Fig. 1a. In the case of advanced robots (Challenge Class) there are different traffic signs (warning signs, mandatory signs, and information or services signs) that must be detected and recognized (Fig. 1b). As shown in Fig. 2, there are also traffic lights mounted right above the zebra crossing (inverted 17' TFT monitors) in order to conduct the robot trials and giving them orders to STOP, follow straight ahead, turn left, parking, and end of trial. Since 2011 University of Trás-os-Montes and Alto Douro participates on this competition with a four-wheel Ackermann steering geometry robot (UTAD McQueen) [1]. The robot chassis is from a 1/5 scale RC car racing adapted with a Maxon EC brushless DC motor droved by a Maxon EPOS 70/10 4-Q servocontroller and the steering is controlled by a servo motor. All the low level control (steering, Maxon EPOS, light, and RGB LED to tell the jury the traffic sign detected) is performed by an Arduino Mega 2560 board. The main controller is a laptop (Samsung RC530, Core i7, with 6 GB of RAM), which communicates with the Arduino board through a USB serial port. Attached to this main computer are two USB u-Eye cameras (UI-1225-LE-C)from IDS, one camera for line following and obstacle detection and the other for traffic signs and traffic lights recognition.

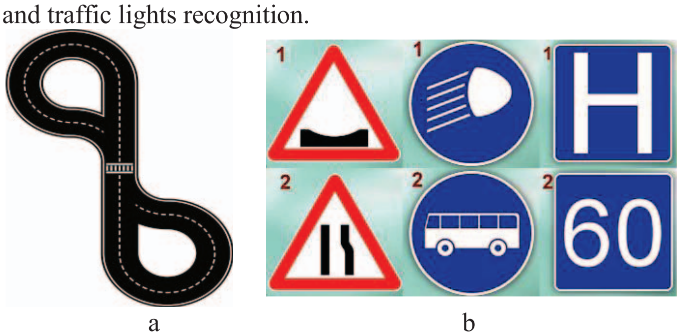


Fig. 1. a) Autonomous driving competition track. b) Some of the traffic signs to be detected and recognized in the competition.

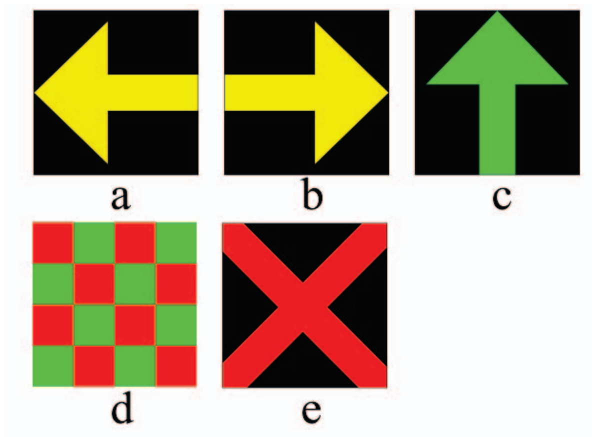


Fig. 2. Traffic lights. a) turn left, b) parking, c) follow straight ahead, d) end of trial, and e) STOP.

The aim of this work is to design and implement a real-time Traffic Sign Recognition (TSR) system for a robot participant in the Autonomous Driving Competition in the National Robotic Festival. The system should be able to recognize nine different traffic signs that appear on the side of

2014 IEEE International Conference on
Autonomous Robot Systems and Competitions (ICARSC)
May 14-15, Espinho, Portugal

the track and five traffic lights displayed by a LED panel above the departure area.

Traffic signs are designed with distinctive features like color, shape and pictogram. These features are the basic elements used for detection and recognition stages in TSR systems. Usually the detection stage is based on the color and shape information of the sign after an initial filtering of the image. Frequently, color-based segmentation is performed thru color thresholding as part of this stage. Several methods have been proposed based on a wide range of color spaces: RGB [2], HSI [3], HSV [4], CIELAB [5] or combinations like Hue thresholding with chromatic colors [6]. The sign shape can be extracted from edge information using Hough shape detectors [7], contour signatures [8] or from Fourier descriptors using correlation based matching [9].

The task of the recognition stage is to decide whether a candidate is in fact a traffic sign and identify it. This classification can be performed using template-matching techniques like correlation [6], or machine learning algorithms [10] like artificial neural networks (ANNs) [11] or Support Vector Machines (SVMs) [12] robust to various illumination conditions, which can perform with an average speed up to 20 frames per second [13]. Authors in [14] propose the use of 2D Independent Analysis (2DICA) and the nearest neighbor classifier for 50 categories traffic sign with 90% of accuracy.

This paper is focused on the challenge of recognize traffic signs and traffic lights, for autonomous robots in the Autonomous Driving Competition. The traffic sign and traffic light problem for this particular challenge has already been addressed in several ways throughout the years. *Zavadil et al.*, proposed a solution, which combines two techniques, namely blob analysis and pattern recognition [15].

The solution proposed in this paper follows a different approach, which combines sign detection via thresholding in HSV space, shape analysis and a trained neural network for classification. The rest of this article is organized as follows. The next section presents the proposed algorithms for detection and recognition of traffic signs and traffic lights. In section III we present and discuss the results obtained from system testing. Final conclusions are presented in Section IV.

II. TRAFFIC SIGN RECOGNITION SYSTEM

The proposed TSR system has three stages of processing: detection, pictogram extraction and classification. The algorithm is applied twice, with subtle modifications, to identify the traffic signs and the traffic lights displayed by a LED panel. In the detection stage, the image regions with signs are selected by color segmentation. In the second stage the interior pictogram (symbol) of the sign is extracted. Finally, in the classification stage, a binary pattern is presented to a feed-forward neural network to recognize the sign (Fig. 3).

A. Traffic Signs Algorithm

Detection:

The system uses a digital camera mounted on the car chassis to acquire frames of the environment surrounding the track where the traffic signs and traffic lights are present.

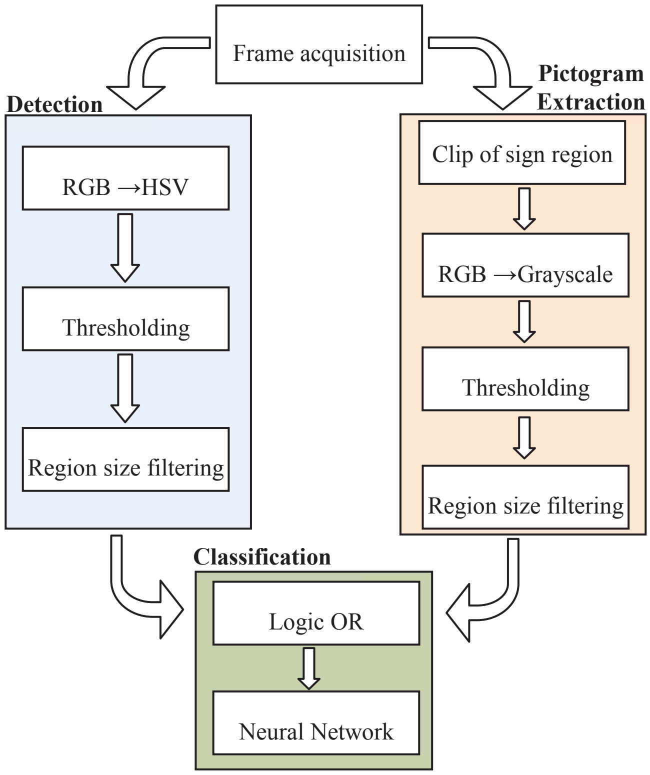


Fig. 3. Overview of the proposed TSR system.

The frames acquired in the RGB color space are converted to HSV color space. HSV color space separates chromatic information from the brightness information making the detection of a particular color simplified and more robust in different illumination conditions, which frequently occur in the competition. Fig. 4a presents a RGB frame captured by the robot camera where traffic sign is visible.

Traffic signs detection is accomplished by thresholding the image represented in the HSV color space. Table I shows the used ranges for thresholding each color component (H, S and V) to detect red and blue regions in the signs. These values were set after testing approximately about 1000 images that have between themselves slight variations in scene illumination.

TABLE I. COLOR RANGES USED FOR THRESHOLDING IN HSV COLOR SPACE.

	H [0 - 180]	S [0 – 255]	V [0 – 255]
Red	[0 - 10] || [160 – 180]	[80 – 150]	[80 – 150]
Blue	[100 - 120]	> 80	> 80

By applying thresholding, a binary image is obtained with several candidate regions for sign location (Fig. 4b), being one of them the external contour, which has blue or red as dominant color. The area (in pixels) of the bounding box involving each candidate region is checked for a minimum and maximum value (region size filtering). The chosen values are one of the distance validations used to discard both too far away or too close traffic signs and other non-sign regions.

Taking into consideration that the bounding box of every sign region (regardless of its shape: circular, triangular, squared or octagon) fits a squared shape, only approximately square regions are further evaluated as signs. After this stage, the external contour of the sign is detected, but the sign pictogram could still be missing (Fig. 4c).

a b c

Fig. 4. a) Frame acquired with a visible traffic sign. b) Regions detected after thresholding. c) Detected external contour.

Pictogram Extraction:

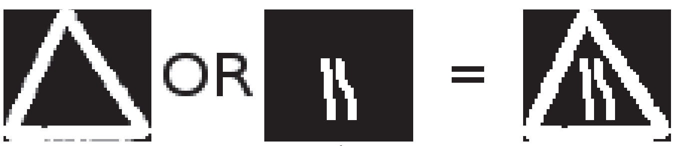
In this stage, the pictogram inside the traffic sign is segmented, extracted and combined with the external contour to create the full binary pattern for classification.

Once the traffic sign position in the image is obtained, a sub-image (clip) of the sign is extracted from the original RGB frame. This sub-image is scaled/normalized to 50x50 pixels and converted to grayscale (Fig. 5a). Two thresholdings are applied to the sub-image in order to segment the interior pictogram: one for dark pictograms and another for white pictograms. For a specific traffic sign, both thresholdings are always applied, and while one extracts the pictogram correctly (Fig. 5b), the other yields nothing, depending on the interior pictogram color to be either white or black. To remove the small noise regions of the resultant binary image, a region size filter is applied in order to preserve only larger regions inside the sign limits. Regions not totally contained in the 4/5 of the image size, are eliminated.

a b c

Fig. 5. a) Sub-image (grayscale) with the sign extrated from the RGB frame. b) Binary image after threshold operation. c) Result of pictogram extraction.

These operations extract the pictogram of the sign (Fig. 5c), which is finally combined with the external contour (from detection stage) through a logical OR operation, to generate the 50x50 pixels binary pattern matrix of the traffic sign (Fig. 6).



a b c

Fig. 6. a) External contour obtained in the detection stage. b) Internal symbol obtained in pictogram extraction stage. c) Binary pattern obtained after logical OR operation.

Classification:

The recognition stage uses a feed-forward Neural Network (NN) as a classifier to recognize the sign. The traffic sign NN has 3 layers: an input-layer with 2500 neurons, each one receiving an element of the pattern binary matrix; a hidden-layer with 20 neurons; and the output-layer with a number of neurons equal to the number of signs to be trained (9 in this work). The neurons from the hidden and output layers use the logistic regression (sigmoid) activation function. Logistic regression outputs a value between 0 and 1, indicating a likeliness of the input pattern belong to one of the class signs. The NN assigns each binary input pattern to the class associated with the output neuron that has the highest probability.

The NN was trained in supervised mode by the backpropagation learning algorithm in a set of 3150 real images consisting in signs from the different classes. Each sample of the training set was previously validated and classified by a human operator. The training set includes signs with different levels of perspective distortion and inaccuracies in the pictogram in order to make the NN more robust to these cases.

B. Traffic Lights Algorithm

The proposed traffic lights recognition algorithm is similar to the traffic sign algorithm described in section III A, with slight modifications in the parameterizations and steps removed or simplified as they are not needed or just do not require as much complexity to correctly detect and recognize the traffic lights. The algorithm explanation will only highlight the differences, without mentioning the steps that are identical to the recognition algorithm used for traffic signs.

In the detection stage, color segmentation is accomplished by thresholding, using another set of ranges for each color component (H, S and V), to detect the red, yellow and green regions that may appear in the traffic lights.

In the traffic light algorithm the pictogram extraction stage is suppressed because the detection stage is enough to locate and obtain the binary pattern matrix. However the chess semaphore is a special case that requires the use of additional processing after threshold to ensure that all the white squares of the binary image will be interconnected, resulting in a single region in the image. To close the gaps that may exist between the white squares on the binary image the morphological operator dilation is applied.

The NN for semaphores classification is slightly different from the one used for traffic signs. It has as many inputs in the input-layer as the traffic signs algorithm, since the detection step also generates a 50x50 pixels binary pattern matrix in the same way. Differences arise at the hidden-layer, which is composed of 15 neurons with logistic regression activation function. The output-layer is composed of 5 outputs, corresponding to the 5 different traffic lights used in the Autonomous Driving Competition. The neurons of the output layer use linear regression activation function. The number of classes for traffic lights classification are fewer and the patterns are simpler than the ones originated from traffic signs, requiring less complexity in the NN to correctly classify them,

2014 IEEE International Conference on
Autonomous Robot Systems and Competitions (ICARSC)
May 14-15, Espinho, Portugal

hence the reduced number of neurons in the hidden-layer and the use of linear regression as the output-layer's activation function.

Finally, in order to increase the TSR response confidence, the system only attests the presence of the traffic sign when it is recognized in several frames in a short time period.

III. Experimental Results and Discussion

The TSR system was implemented in C++ language, using the *Opencv* [16] image processing library and the *Fast Artificial Neural Network* (FANN) [17] open source library. The application uses the IDS software suite [17] for control and management of the *u-Eye* camera.

The camera field of view and the algorithms parameterizations were adjusted so that the TSR system is able to recognize the signs when they are relatively close to the vehicle (between 1m to 2,5m) and appear relatively aligned with the camera.

Experimental tests were performed on a track mounted on the robotic laboratory at UTAD with similar conditions to those of the Autonomous Driving Competition. A total of fourteen video sequences were acquired to evaluate the system performance. In each video a different traffic sign or traffic light is present in a large number of frames. The TSR system is prepared to recognize 9 different traffic signs (3 of them are warning signs, 3 are information signs and another 3 are mandatory signs) and 5 different traffic lights.

Table II presents the results obtained on the traffic signs recognition. Each line of the table has the number of frames where the correspondent sign appears (# samples), the number of true positives, false negatives and false positives, after stage 2 (detection + pictogram extraction) and after the classification stage.

The results after stage 2 show a low rate of false positives, but a relatively high rate of false negatives, caused by the incorrect sign region detection or the incorrect pictogram extraction. The classification stage shows a null or very low rate of false positives, but a relatively high rate of false negatives, mainly because in the input of the neural network, the patterns already present large inaccuracies resulting from the previous stage. Furthermore the output layer of the NN was configured to assign the binary input pattern to one of the classes, only if the highest response is above a threshold value of 0.9. If the value is below 0.9, the sign is classified as unknown, which on one hand increases the rate of false negatives but on the other hand prevents the increase of false positives.

The two last columns of Table II present the metrics *precision* and *recall* used to measure the overall system performance. *Precision* (ratio between *true positives* and the sum of *true positives* and *false positives*) can be interpreted as the probability of a sign assigned to a class being correctly classified, while *recall* (ratio between *true positives* and the sum of *true positives* and *false negatives*) can be interpreted as the system probability to correctly recognize a sign.

Depending on the sign, the traffic sign algorithm presents different *recall* values, between 52.2% for the narrow passage sign and 88.2% for the hospital sign. The high *recall* values observed in the majority of signs, suggests that in most cases the system detects and classifies them correctly. It is observed that *precision* is 100% or a value close to it in all the classes, suggesting that when the system assigns a class to a sign hardly misses.

The system was configured to report the presence of a sign if at least it is detected in 10 frames in a time interval of three seconds. Using this multi-frame validation, even for the worst case which occurs with the narrow passage sign (*recall* equal to 52.2%), the system was able to recognize nearly all signs that appear around the track. Also considering the low rate of false positives, we can affirm that the system has a high probability to recognize any sign correctly.

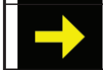
TABLE II. Results of traffic signs recognition

		Detection + Pictogram extraction			Classification				
	# Samples	True positives	False negatives	False positives	True positives	False negatives	False positives	Recall	Precision
	2592	2516	76	4	2004	516	4	77.3%	99.8%
	3201	2702	499	0	2477	225	0	77.4%	100%
	3084	2379	705	12	1611	780	0	52.2%	100%
	3202	2962	240	0	2720	242	0	84.9%	100%
	3289	2984	305	10	2835	159	0	86.2%	100%
	2718	2474	244	0	2236	238	0	82.3%	100%
	3071	2713	358	149	1896	966	0	61.7%	100%
	3044	2988	56	50	2685	353	0	88.2%	100%
	3109	2828	281	0	2558	270	0	82.3%	100%

2014 IEEE International Conference on
Autonomous Robot Systems and Competitions (ICARSC)
May 14-15, Espinho, Portugal

Table III shows the results in traffic lights recognition with the same information as in Table II.

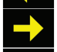
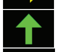
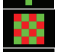
TABLE III. RESULTS OF TRAFFIC LIGHTS RECOGNITION

		Detection + Pictogram extraction			Classification				
	# Samples	*True positives*	*False negatives*	*False positives*	*True positives*	*False negatives*	*False positives*	*Recall*	*Precision*
	1750	1747	3	0	1723	24	0	98.5%	100%
	1401	1384	17	0	1384	0	0	98.8%	100%
	1496	1496	0	0	1492	4	0	99.7%	100%
	2001	1968	25	8	1931	45	0	96.5%	100%
	2001	2001	0	0	1986	15	0	99.3%	100%

In general, the results obtained in traffic lights recognition are better than the ones obtained in traffic signs mainly because the pictogram is simpler to extract, obtained immediately after the detection stage. The results after stage 2 show that detection and pictogram extraction work very well in all the classes, with exception to the red and green checkers sign which registers 8 false positives and 25 false negatives instances. The classification stage has no false positives and few false negatives. False negatives arise because the inputs patterns that feed the neural network have inaccuracies that impair its performance. *Precision* is 100% and *recall* also presents values close to 100%, which demonstrates the good performance of traffic light algorithm. As in the traffic sign algorithm, the neural network uses a threshold value to guarantee a high precision in its response, even if it means an increase in the number of false negatives.

The traffic lights algorithm was also tested with images taken from an online dataset [19]. The results in Table V show a lower performance of the algorithm, namely in the *recall* metric in some of the signs. The images included in the dataset have a lower resolution (320x240) than the images acquired with our camera *u-Eye* (752x480). Also, many images have strong blurriness levels which difficult the classification and allows for a higher detection of false positives. There are also big variations in illumination between some images of the dataset, which render the algorithm incapable of detecting some traffic lights, increasing the number of false negatives.

TABLE IV. RESULTS OBTAINED WITH IMAGES TAKEN FROM AN ONLINE DATASET.

	# Samples	***True positives***	***False negatives***	***False positives***	***Recall***	***Precision***
	72	47	21	4	65,3%	92,2%
	64	41	20	3	64,1%	93,2%
	79	67	10	2	84,8%	97,1%
	113	100	12	1	88,5%	99,0%
	172	164	1	7	95,3%	95,9%

Low computational time has been one of the main concerns in the algorithm implementation. Table V shows average time in processing one frame captured by the camera, when running the TSR application on a laptop with a Core i7 2.4 GHz processor.

The traffic sign algorithm takes approximately about 3.5 milliseconds to complete the recognition task, while the traffic light algorithm takes approximately 4 milliseconds. The slower detection in traffic lights is due to the need of looking for three colors unlike the traffic signs algorithm that looks only for two colors. It is also observed that the first two stages, sign detection and pictogram extraction are the most time consuming in any of the algorithms

TABLE V. PROCESSING TIME OF THE ALGORITHM

	Traffic Signs (ms)	**Traffic Lights (ms)**
Detection + Pictogram	3.487	4.004
Classification	0.057	0.059
Total	3.544	4.063

The execution time of both algorithms is a very important factor in the robot performance, since it needs computational power for other processes related with the autonomous driving. After some empirical tests the camera was configured to acquire 30 frames per second which is more than enough to have excellent results with the TSR system, releasing this way computational power for other tasks.

IV. CONCLUSION

In this paper a real time system for traffic sign recognition by a robot participant in the Autonomous Driving Competition in the National Robotic Festival is proposed. The TSR was implemented on a laptop that manages and controls all the robot action in the competition. The laptop processes the frames acquired by a *u-Eye* camera mounted on the robot chassis.

Two similar algorithms were implemented, one for traffic sign recognition and the other for traffic lights recognition. Both algorithms are composed by three stages of processing: detection, pictogram extraction and classification. To evaluate the developed algorithms, tests were performed under the same conditions of the competition. The results obtained in both algorithms achieved *precision* of 100% in most of the signs. *Recall* is the other metric used for algorithms evaluation which presents rates above 96% in the case of traffic lights and slightly lower (between 52% and 88.2%), in the case of traffic signs.

Detailed analysis of the results of each of the processing stages reveals that recognition errors are due mostly to failures in detection and pictogram extraction. Naturally, the neural network when receiving inaccurate inputs fails the classification of these samples. By setting a high threshold value to accept the NN response, the system presents low rates of false positives. This prevents non-sign regions resulting from stage 2 (detection + pictogram extraction) of being classified by mistake as signs. It can be concluded that when the system assigns a class to a sign the degree of confidence in the response is very high (confirmed by high values of *precision*).

The system has been prepared to simultaneously recognize more than one sign if they are visible in the image. When multiple traffic signs are detected, multiple regions of interest (ROI) are detected in the image. In these cases, the ROI are prepared as usual, as if only one sign had been detected. Each of the detected regions is classified by the NN. The algorithm has been limited to accept only the first three detected ROI per frame. Also, the system has been trained for rotated traffic signs, allowing it to successfully recognize signs with slightly distortion perspective.

The results obtained assume that the algorithms were previously parameterized with fixed values in order to fit the environment illumination conditions. In the future this task can be totally automated. Once the algorithm is prepared for the environment, it demonstrates to have a very high *precision* rate for most traffic signs. Traffic signs that were not classified correctly were mostly patterns of the sign, which were highly distorted or were considerably fragmented from the detection stage.

Since the system uses fixed values for thresholding, abrupt lighting changes in the environment make it vulnerable to errors. Also, occlusion and motion blur affect the quality of the results. Even when the classifier gives a high likeliness answer for a given sign, it might not be enough to overcome the defined threshold to actually account it as a valid traffic sign, yielding a false negative.

REFERENCES

[1] T. Moura, J. Teixeira, F. Tuna, F. Moreira, A. Valente, V. Filipe, S. Soares. "Reconhecimento de Sinais de Trânsito para Prova de Robótica de Condução Autónoma". Proceedings of 19th Annual Seminar on Automation, Industrial Electronics and Instrumentation (SAAEI'12), pp. 645-649, 2012.

[2] V. Prisacariu, R. Timofte, K. Zimmermann, I. Reid, and L. Van Gool, "Integrating object detection with 3D tracking towards a better driver assistance system", in Proc. 20th ICPR, pp. 3344–3347, August 2010.

[3] V. Prisacariu, R. Timofte, K. Zimmermann, I. Reid, and L. Van Gool, "Integrating object detection with 3D tracking towards a better driver assistance system", in Proc. 20th ICPR, pp. 3344–3347, August 2010.

[4] X. Qingsong, S. Juan, and L. Tiantian, "A detection and recognition method for prohibition traffic signs", in Proc. Int. Conf. IASP, pp. 583–586, April 2010.

[5] F. Ren, J. Huang, R. Jiang, and R. Klette, "General traffic sign recognition by feature matching", in Proc. 24th Int. Conf. IVCNZ, pp. 409–414, November 2009.

[6] E. Krsak, S. Toth, "Traffic sign recognition and localization for databases of traffic signs", in Acta Electrotechnica et Informatica, vol. 11, no. 4, pp. 31-35, 2011.

[7] S. Houben, "A single target voting scheme for traffic sign detection", in Proc. IEEE IV Symp., pp. 124–129, June 2011.

[8] S. Xu, "Robust traffic sign shape recognition using geometric matching", IET Intell. Transp. Syst., vol. 3, no. 1, pp. 10–18, March 2009.

[9] F. Larsson and M. Felsberg, "Using Fourier descriptors and spatial models for traffic sign recognition", in Proc. Image Anal., pp. 238–249, 2011.

[10] D. Pei, F. Sun and H. Liu, "Supervised Low-Rank Matrix recovery for Traffic Sign Recognition in Image Sequences", IEEE Signal Processing Letters, vol. 20, no. 3, March 2013.

[11] J. Stallkamp, M. Schlipsing, J. Salmen and C. Igel, "Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition", Neural Networks, vol. 32, pp. 323-332, 2012.

[12] J. Park, J. Kwon, J. Oh, S. Lee, J-Y. Kim and H-j. Yoo, "A 92-mW Real-Time Traffic Sign Recognition System With Robust Illumination Adaptation and Support Vector Machine", IEEE Journal of Solid-State Circuits, vol. 47, no. 11, November 2012.

[13] J. Greenhalgh and M. Mirmehdi, "Real-Time Detection and Recognition of Road Traffic Signs", IEEE Transactions on Intelligent Transportation Systems, vol. 13, no. 4, December 2012.

[14] C. Zi-xing and G. Ming-qin, "Traffic sign recognition algorithm based on shape signature and dual-tree complex wavelet transform", J. Cent. South Univ., vol. 20, pp. 433-439, 2013.

[15] Zavadil, J.; Tuma, J.; Santos, V.M.F., "Traffic signs detection using blob analysis and pattern recognition" Carpathian Control Conference (ICCC – 2012), pp. 776 – 779, 2012.

[16] Web pages of the OpenCV library, [online], 2014, Accessible on www: http://opencv.org

[17] Web pages of the FANN library, [online], 2014, Accessible on www: http://leenissen.dk/fann/wp/

[18] Web pages of IDS company, [online], 2014, Accessible on www: http://en.ids-imaging.com/

[19] Online image dataset of traffic lights, [online], 2014, Accessible on: http://lars.mec.ua.pt/public/LAR%20Projects/Vision/2011_JaromirZavadil/Images/.