Detection of traffic signs based on Support Vector Machine classification using HOG features

David Coțovanu
Gheorghe Asachi Technical University of Iasi
Faculty of Electrical Engineering
Iași, Romania
cotovanudavid@gmail.com

Cristian Zet
Gheorghe Asachi Technical University of Iasi
Faculty of Electrical Engineering
Iași, Romania
czet@tuiasi.ro

Cristian Foşalău
Gheorghe Asachi Technical University of Iasi
Faculty of Electrical Engineering
Iași, Romania
cfosalau@tuiasi.ro

Marcin Skoczylas
Faculty of Computer Science
Bialystok University of Technology
Bialystok, Poland
m.skoczylas@pb.edu.pl

***Abstract*—Real time traffic sign recognition algorithms are in high demand as technology pushes for autonomous vehicles. Identifying traffic signs that regulate the flow of traffic provides another way to increase the safety of the driver and other road participants. In the present paper we propose an algorithm that introduces a filtering step which significantly reduces the processing times of an SVM based classification algorithm. Our approach uses the image processing techniques to identify regions of interest (ROIs) in an image, based on color information and on certain object properties. In the experiments, we evaluated our TSDR system on recordings of Romanian roads with traffic signs in various lighting conditions.**

Keywords—traffic sign recognition; image processing; SVM; color segementation; real-time; object detection;

I. Introduction

The safety of drivers and other traffic participants is of great importance and has been researched extensively since the modern days of motorized transportation. Besides the development of physical systems to improve the driver's safety, different algorithms are being developed, algorithms that use inputs like videos from cameras or environment mapping from a laser sensor like LIDAR, in order to understand what is around the vehicle and to provide feedback or even vehicle control to avoid the worst-case scenario or notify (prevent) the driver from making a dangerous or illegal maneuver. Thus, in order to increase the road safety, systems that perform the automatic detection of people, cars, traffic signs, obstacles or any other object of interest are being developed continuously.

The computer vision concept deals with processing the frames coming from cameras, analyzing each one for extracting the useful information, which is later used in an algorithm. In the automotive industry, computer vision algorithms are used to implement traffic sign detection and recognition systems (TSDR). Depending on the road conditions, weather, or even the driver's condition, a traffic sign may not be noticed by the driver, which can increase the likelihood of an accident.

Different situations, such as changes in the environment or variations of lighting conditions, make the algorithm's job of identifying a traffic sign a difficult task. There are also problems with blurred images, partial occlusion of a traffic sign, multiple instances of the same traffic sign appearing in a single frame.

The purpose of this paper is to develop a TSDR system which uses image processing algorithms to increase the speed and accuracy of the SVM classification method for the detection and recognition of multiple and different types of traffic signs that vary in shape and color. The detection part relies on the color information to identify candidates that will later be classified using an SVM classifier trained using Histogram of Oriented Gradients (HOG) features.

II. Related work

Authors in previous works [1] presented robust algorithm based on keypoints feature detectors. The ROIs were detected based on geometric shape of signs, and then the keypoints within ROIs were iteratively compared with a training database. This paper continues the research; however, approach is changed and instead of keypoints comparison the recognition is performed using SVM classifier.

In [2] the authors used color-based method to identify possible locations for a traffic sign of a specific color (red, blue or yellow) in an image. They also employed the use of shape approximation using the Ramer-Douglas-Peucker method to identify the shape of the traffic signs. Computers use the RGB (Red, Green, Blue) color space to display images, but this color space links the base colors, making the color processing tedious and susceptible to light and tone variations.

978-1-5386-5062-2/18/$31.00 ©2018 European Union

A color space closer to human perception is HSV (Hue, Saturation, Value) and its advantage is that it separates the chromatic and light information, thus providing a low sensitivity to brightness variation. Because of this, HSV color segmentation has been used in [3], [4] with great success by selecting the appropriate thresholding parameters for their specific application.

Color segmentation is one of the most used methods to identify the traffic sign candidates. By utilizing the HSV color space, we eliminate the high sensitivity to light intensity of RGB color space. There are authors that used other color spaces like YIQ [5], L*a*b, YUV or CIE for the detection part. Shape based methods that use the Hough transform [6] or neighborhood characteristics and symmetric features [7] to identify ellipses, rectangular or triangular objects provide additional information used to identify the objects present in an image.

Machine learning algorithms are used extensively to solve a wide variety of problems. There are several tasks a machine-learned system can do depending on the desired output. One of them is classification, which consists in dividing the inputs into two or more classes. HOG based SVM have been used with success in different applications over the years for detecting different objects, such as humans [8], [9] and cars [10].

HOG features are obtained by evaluating normalized local histograms of image gradient orientations in a grid. Characterization of local object shape and appearance can be done without knowing the corresponding gradient or edge position; the distribution of local intensity gradients or edge directions is enough to provide good local feature descriptions which can be used to train an SVM classifier.

III. System overview

The proposed system contains the following two stages: detection of traffic sign candidates using color-based segmentation and recognition of traffic signs through classification of the candidates. In the first part the ROIs are extracted from the image. They will be used in the second part, where the traffic signs are recognized. The figure bellow illustrates the algorithm outline of our traffic sign recognition method.

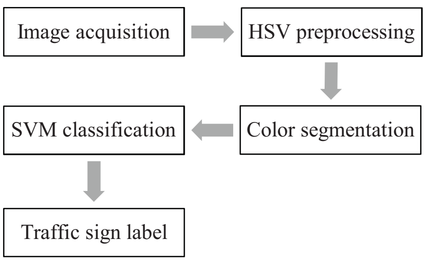

Fig. 1. Proposed TSDR system algorithm outline.

A. *Image acquisition and preprocessing*

Image acquisition was done using a personal dashcam (Viofo A119) that recorded daily, from which I selected several parts that reflect different times of the day in different conditions. Recordings were done using a frame rate of 30 frames/second at a resolution of 1920 by 1080 pixels.

An essential step after obtaining an image and applying any image processing algorithms is the preprocessing step. This preliminary stage is used to deal with background noise and prepare the image for further processing. Thus, a conversion from the RGB to the HSV color space is applied, followed by the use of a Gaussian filter on each channel in order to smooth the image and to eliminate the noise that could introduce false detections later on.

In order to further improve the speed of the algorithm, we assumed that most traffic signs present on roads are located close to the horizon line or above it. Thus, we applied a simple rectangular mask to ignore 40% of the lower section of the image.

B. *Detection of traffic sign candidates*

The purpose of the detection step is to extract a smaller number of candidate bounding boxes compared to simply using a sliding window approach over the image. This ensures a faster object detection through preliminary selection of regions of interest that may or may not contain a traffic sign.

For identifying ROIs in our images, we utilized color segmentation. Variables like time of the day, weather conditions, shadows or reflections affect the performance of color segmentation algorithms. As a result, we can use color information only to identify objects that later may be classified as a traffic sign.

We used the HSV color space to lower the sensitivity to light intensity variations and other unknown defects that may be present on the signs. Fixed thresholds are used to extract portions of the image that contain red and blue colors. The values for the thresholds were deduced empirically using images with traffic signs and can be seen in Table I. The values for hue H are within the interval [0, 180], while for the saturation S and value V the interval is [0, 255].

TABLE I. THRESHOLD VALUES FOR TRAFFIC SIGN DETECTION

Color	HSV Channels		
	Hue	***Saturation***	***Value***
Red	1 < H< 20 Or 160 < H< 180	80 < S< 240	50 < V< 220
Blue	95 < H< 155	80 < S< 240	50 < V< 220

The result of color segmentation is a binary image containing groups of the pixels that passed the above-mentioned threshold limits. Because of the variety in road conditions and other uncontrollable variables, the groups may contain imperfections like open ends or unwanted noise which will provide an inaccurate detection stage. To solve this, we used morphology operations that remove extremely small groups and fill the gaps in groups that contain holes. On the red color binary image, as seen in Fig. 2, closing operation was applied to fill any gaps followed by an opening operation which removes small groups of pixels. For the blue detections, only an opening operation was necessary to remove small noise detections. The red and blue detections are merged using the bitwise OR function, the result being intersected with the saturation and value threshold results using the AND operator. Finally, a closing and an opening operation provides the result seen in Fig. 3.

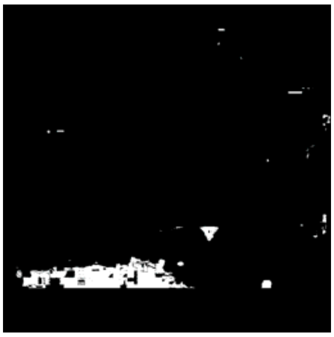

Fig. 2. Red hue color segmentation.

Fig.3. Results of morphology operations

Fig. 4. Detection step results. Candidate is contained by the yellow box

To reduce the number of candidates even more without losing actual traffic signs, we used an additional filtering criterion, namely aspect ratio. Most of the traffic signs have width to height aspect ratio around 1, but due to their placement on the road or unwanted damages, this ratio can vary. The value for the *aspect ratio* of detected objects is calculated using the equation (1) and all candidates with this value lower than 0.7 will be rejected.

$$aspect\ ratio = \frac{\min(width,height)}{\max(width,height)} \tag{1}$$

$$scale\ factor = \frac{image\ size}{\max(width,height)} \tag{2}$$

After we have selected the most probable objects for our traffic sign candidates, we added a 20% padding to the detected bounding boxes to make sure the entire traffic sign is inside its respective bounding box. An example of the final bounding box obtained during the detection phase can be seen in Fig. 4.

All detected regions are then extracted from the original image and resized to normalize the dimensions of small detections or large detections by using a *scale factor* obtained using equation (2) where *image size* is the arbitrary value for the desired resized image dimensions. This eliminates the need to use the pyramid approach that SVM detectors traditionally use, method that increases computation time.

C. *Recognition of traffic sign candidates*

Identifying the traffic sign is the final step and it consists in classifying all the candidates obtained in the previous step. We used a linear SVM binary classifier trained with HOG features to check if a sign candidate is an actual traffic sign and identify it. The necessary tools to create our SVM based classifier were provided by the Dlib library [11]. The machine learning section of the library created by Davis King contains two versions of SVM solvers, one of which implemented LIBSVM [12] and a kernelized version of the Pegasos algorithm introduced by Shalev-Shwartz [13]. The max-margin object detection algorithm [14] describes in detail the SVM method.

TABLE II. SVM CLASSIFIER TRAINING DATASET

Traffic sign	**Training**		**Testing**	
	Images	***Labels***	***Images***	***Labels***
Yield	64	69	10	10
Pedestrian crossing	42	44	10	11

Fig. 5. Yield traffic sign examples used for training our SVM classifier

Fig. 6. Pedestrian crossing traffic sign examples used for training our SVM classifier

For the training part of the SVM classifier we used a set of images of Romanian roads obtained from different public sources. These images contain the traffic signs we need to recognize. A supervised training method was used, meaning that we labeled all signs necessary to be recognized, known as positive samples and everything else present in the image is considered negative sample.

A total of 64 images containing 69 instances of the yield traffic signs were used in the training process. For the pedestrian crossing, we used 42 images containing 44 labeled pedestrian crossing traffic sign. Examples of images with different lighting conditions and various orientations for the traffic signs can be seen in Fig. 5 and Fig. 6, respectively. Compared to other machine learning algorithms used for classification task, SVMs do not require many samples to create a good classifier.

The result of the training process is an SVM model that contains vectors arranged such that points belonging to different classes are separated by max-margin hyperplane. In our application we used HOG features to train our SVM classifier. A visualization of the trained models for the two traffic signs, yield and pedestrian crossing, can be seen in Fig. 7.

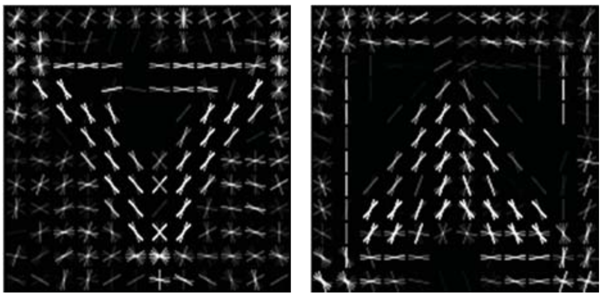

Fig. 7. Visualization of HOG features for yield (left) and pedestrian crossing (right) traffic signs

Because the SVM classification algorithm utilizes the sliding window approach to scan an image, the fixed arbitrary size for the sliding window determines a size and aspect ratio for the object to be classified. Thus, to recognize one traffic sign in different orientations or perspectives, it is necessary to train different classes with different aspect ratios.

IV. Experimental results

TABLE III. Test dataset and results

Traffic sign	***Signs***	***Detections***	***Recognitions TSDR/SVM***	***Average FPS TSDR/SVM***
Yield	645	545	371/422	10.2/0.31
Pedestrian crossing	2254	1883	918/1061	8.3/0.31

TABLE IV. Experimental results

Traffic sign	***Detection rate***	***Classification rate***	***Recognition rate TSDR/SVM***
Yield	84.49%	68.07%	57.51%/65.42%
Pedestrian crossing	83.54%	48.75%	40.72%/47.07%

Fig. 8. Detection and recognition results. Detected candidates and recognized traffic signs are marked with yellow and red bounding boxes respectively.

To test our algorithm proposal, we used two sets of recordings, different from the training dataset. First set of recordings contains predominantly yield traffic signs present and the second one contains mostly pedestrian crossing traffic signs. The video recordings were saved as image sequences and loaded as individual images in a for-loop to have control over the processing time and other extracted statistics.

The exact results can be seen in Table III and Table IV, where TSDR is the algorithm implemented by us which includes the detection step and SVM is just the classifier without the detection step.

Out of 645 yield traffic signs, 545 were detected as candidates in the first step of the algorithm and 371 were recognized with a frame rate of 10.2. Without the detection step, 422 yield signs were recognized with a low frame rate of 0.31. For the pedestrian crossing signs, 1883 out of 2254 were detected. Using only the classification algorithm, 1061 pedestrian crossing signs were recognized and with our system 918 signs were recognized at a frame rate of 8.3.

The detection algorithm we implemented prior to the classification step increased the processing speed 30 times with a 12.77% loss in traffic signs recognized. We obtained an 84% detection rate, 58.41% of the detected traffic signs being classified. Out of all traffic sign present in the test images, 49.11% of them have been recognized as either a yield sign or a pedestrian crossing sign.

The low recognition rate can be attributed to a non-ideal model used for our classifier and loss of candidates in the detection step. By using a better classifier, the recognition rate can be increased up to the detection rate. After that, a better implementation of the detection step should provide an increased rate of detection and recognition of the traffic signs.

With our TSDR system we achieved our goal of implementing a method that allows for real time use of SVM classification to detect and recognize traffic signs. Utilizing the only the SVM classification algorithm, the average processing time for a frame was 3 seconds, making it unusable for real-time applications. After applying our candidate detection algorithm, we obtained an average processing time of 0.1 seconds, enabling real-time usage.

V. Conclusion

We created an algorithm which improves the processing times for the SVM based classification method, enabling real-time capabilities necessary for today's autonomous vehicles. The use of a candidate selection algorithm provided an increase in processing rate, up to 10 frames/second, while losing only 12.77% recognition rate compared to the same SVM classification algorithm without the detection step.

The low recognition rate can be increased by using a better dataset for training the SVM model of each traffic sign class. Additional models for the same class can be trained to improve the recognition of cases when the signs are distorted by perspective or incorrect positioning due to external causes. Increasing the number of classes will lower the processing rate by a small amount.

There is room for improvement for both the detection step and the recognition step in our algorithm. Implementing methods to ignore sporadic or inconsistent detections can further lower the processing time. Another implementation that can provide increased recognition rates would be a tracking algorithm that can remove the need of successive classification steps for an already identified traffic sign.

Acknowledgement

This work has been supported from the Bridge Grant PN-III-P2-2.1-BG-2016-0236 financed by the Romanian Ministry of Research and Innovation.

REFERENCES

[1] M. Lasota, M. Skoczylas, "Recognition of multiple traffic signs using keypoints feature detectors," Procedings of the 2016 International Conference and Expositon on Electrical and Power Engineering, IEEE, 2016, pp. 2016

[2] I. S. David Soendoro, „Traffic sign recognition with Color-based Method, shape-arc estimation and SVM," în Proceedings of the 2011 International Conference on Electrical Engineering and Informatics, 2011.

[3] J. P. a. P. P.Paclik, „Roadm sign classification using Laplace kernel," Pattern Recognition Letters, 2000.

[4] K. Y. Aryuanto Soetedjo, „An Efficient Algorithm for Traffic Sign Detection," Journal of Advanced Computational Intelligence and Intelligent Informatics, vol. 10, nr. 3, pp. 409-418, 2006.

[5] N. Dalal şi B. Triggs, „Histograms of Oriented Gradients for Human Detectio," INRIA Rhone-Alps, 2005.

[6] X. Cao, C. Wu, P. Yan şi X. Li, „Linear SVM classification using boosting HOG features for vehicle detection," In Proceedings of the 2011 18th IEEE International Conference on Image, 2011.

[7] Z. Changjun şi C. Yuzong, „The Research of Vehicle Classification Using SVM and KNN in a Ramp," 2009.

[8] N. Dalal şi B. Triggs, „Histograms of Oriented Gradients for Human Detectio," INRIA Rhone-Alps, 2005.

[9] X. Cao, C. Wu, P. Yan and X. Li, „Linear SVM classification using boosting HOG features for vehicle detection," In Proceedings of the 2011 18th IEEE International Conference on Image, 2011.

[10] Z. Changjun and C. Yuzong, „The Research of Vehicle Classification Using SVM and KNN in a Ramp," 2009.

[11] Davis E. King, "Dlib-ml: A Machine Learning Toolkit", Journal of Machine Learning Research 10, pp. 1755-1758, 2009

[12] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A Library for Support Vector Machines, 2001

[13] Shai Shalev-Shwartz, Yoram Singer, and Nathan Srebro. Pegasos: Primal estimated sub-gradient solver for svm. In ICML '07, pages 807–814, New York, NY, USA, 2007. ACM

[14] Davis E. King "Max-Margin Object Detection", https://arxiv.org/abs/1502.00046, 2015