Real Time Traffic Sign Detection Using Color and Shape-Based Features

Tam T. Le[1], Son T. Tran[2], Seichii Mita[2], and Thuc D. Nguyen[1]

[1] University of Science, VNU-HCMC, Vietnam
{lttam,ndthuc}@fit.hcmus.edu.vn
[2] Toyota Technological Institute of Nagoya, Japan
{ttson,smita}@toyota-ti.ac.jp

Abstract. This paper presents a new approach for color detection and segmentation based on Support Vector Machine (SVM) to retrieve candidate regions of traffic signs in real-time video processing. Instead of processing on each pixel, this approach utilizes a block of pixels as an input vector of SVM for color classification, where the dimension of each vector can be extended by a group of neighboring pixels. This helps to handle the diversification of data on both training and testing samples. After that, Hough transform and contour detection are applied to verify the candidate regions by detecting shapes of circle and triangle. The experimental results are highly accurate and robust for our testing database, where samples are recorded on various states of environment.

Keywords: traffic-sign detection, color detection and segmentation, SVM.

1 Introduction

Traffic signs on road tell drivers about traffic rules and states of road such as warning, prohibition, limitation of speed, and so on. With that useful information, our traffic is safer and more convenient. In vision-based methods, the traffic-sign detection and recognition have many difficulties due to changes of environment and speed of vehicles. For instance, traffic signs are usually faded when they are exposed a long time under the sunlight. Moreover, their colors are changeable for various lights influenced by weather conditions such as fog, cloud, or snow. It is not only affected by the illuminant color of the daylight but also by the weak nightlight. The different directions of view also make traffic signs difficult to detect and recognize from camera positioning inside of cars. In addition, traffic signs may be also occluded by trees, buildings, or pedestrians. At the present time, the evaluation of a new approach by comparing with the existing methods is not a simple task because there is not any standard traffic sign database.

Most reviewed literature has used color as one of the main features to detect traffic signs beside shape-based features. The first approach of color detection and segmentation has utilized thresholds in a suitable color space, where HSV color space is the first choice [1, 2] because it is more intuitive than the others and we can separate the color information from the brightness to generate a continuous space of color, where

N.T. Nguyen, M.T. Le, and J. Świątek (Eds.): ACIIDS 2010, Part II, LNAI 5991, pp. 268–278, 2010.

it is not supported in RGB color. This approach is simple, but it is difficult to define threshold of an interested color due to continuity of color space. A de la Escalcra [3], et. al., proposed two look up tables on hue and saturation channel in HSI color space to enhance the ability of interested-color separation. H. Fleyeh [4] presented algorithms of the dynamic threshold, fuzzy color segmentation, and shadow highlight invariance. These algorithms utilize one global value of all pixels in image as an additional constraint to detect and segment color more accurately. C. Fang [5], et. al., presented two-layer neural network on the hue channel in HSI color space to detect interested color. This method matched intensity values of testing pixels to those of interested color on hue channel by the neural network.

Support Vector Machine (SVM) is an efficient learning method, especially in traffic-sign detection problem. Satumino[6], et. al., and Kiran[7], et. al., applied this method for the shape classification phase as well as the recognition phase. And in this paper, we present a new approach for the color detection and segmentation based on SVM to retrieve candidate regions of traffic signs in real-time video processing. Instead of processing on each pixel, our approach utilizes a block of pixels as an input vector of SVM for color classification, where the dimension of each vector can be extended by a group of neighboring pixels. This helps to handle the diversification of data on both training and testing data. Our contribution is the first method of applying SVM in color classification for traffic sign detection.

2 Model of Traffic Sign Detection

We propose an algorithm of color detection and segmentation based on SVM to retrieve candidate regions. The algorithm assigns input of SVM as a block of pixels, a group of neighboring pixels located at each position in the image, to enhance the precision as well as the recall. Especially, the algorithm can reduce a part of noise when traffic signs are exposed in bad weather such as cloudy and snowy days. Figure 1 presents steps of our algorithm to detect traffic signs in real-time video processing. Our method is based on color detection and segmentation using SVM on blocks of pixels.

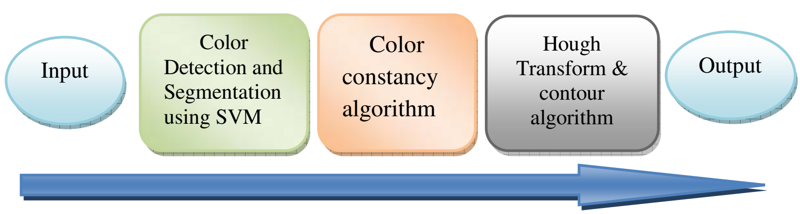

Fig. 1. The proposed model to detect traffic signs

To speed up the processing time, the size (640x480) of the input image is downscaled to (320 x 240). The color detection and segmentation using SVM is applied on the downscaled image. Then, the candidate regions are extracted from the original image by a projection from each position of the candidate region on downscaled image to that of the input image. The candidate regions are enhanced by color constancy algorithm [8]. Figure 2 presents one result of the color constancy algorithm. We can see in Fig. 2 that the edge and color features in the right image are enhanced from the left one.

Fig. 2. The result of using color constancy algorithm

Finally, Hough transform and contour algorithm are utilized to verify the candidate regions. The next sections present the color detection by using SVM and traffic-sign segmentation by using Hough transform and contour algorithm.

2.1 Color Detection and Segmentation by Using SVMs

Different from most of existing methods dealing with each pixel as input, our approach concerns blocks of pixels; therefore, the information of neighbor pixels can help to handle the diversification of both training and testing data. It means that SVMs can return a recall rate better than that of single-pixel-based algorithms. In this approach, instead of deciding whether a pixel has an interested color or not, our method chooses blocks of interested color through the results of SVMs. Using feature extraction of a pixel block presented in the section of 2.1.1 will help to reduce complexity of the calculation in SVMs because the dimensions of the input vectors, support vectors, and the hyper-plane are only equal to two times of the number of pixels in each block.

2.1.1 Feature Extraction and Data Set

There are many color spaces such as RGB (Red, Green, and Blue), HSV (Hue, Saturation, and Value), HSL (Hue, Saturation, and Lightness), CIELUV, and so on. Among those color spaces, the RGB space is used widely. The RGB channels have a high correlation and its chrominance and luminance data are mixed together. Although RGB color system is easy for transmission in communication channels, it is difficult to separate the interested color because it is not a continuous color space. Meanwhile, HSV color space has the intuitive color based on artist's ideas of tint, saturation, and tone. In the HSV system, the Hue channel contains dominant color and it is invariant with highlight in white light sources. The saturation channel represents the colorfulness of an area in proportion to its brightness and the value of the channel contains color luminance. CIELUV color space was proposed by G. Wyszecki and standardized by CIE (Commission International de L'Eclairage), it is a system of perceptually uniform color. Our approach chooses ratios of RGB channel in [9] because it can reduce the dimension and allow defining the specific features for the interested color.

Let I denote an image of $(n \ x \ m)$ and blk denote a block of $(2 \ x \ 2)$. Each block on the image is not overlapped with the other blocks.

$$I = \bigcup_{i=1}^{\frac{n}{2} x \frac{m}{2}} blk_i \tag{1}$$

Each pixel p of block blk has a value of (r, g, b) in the RGB color space. We define the value of α and β as follows;

$$\alpha = \frac{r}{g} \qquad \beta = \frac{r}{b} \tag{2}$$

Then, each pixel p can be expressed by $p = (\alpha, \beta)$, and block blk is defined as the follow;

$$blk = \begin{pmatrix} p_{ij} & p_{i(j+1)} \\ p_{(i+1)j} & p_{(i+1)(j+1)} \end{pmatrix} \tag{3}$$

where $i = 1,..,n-1$ and $j = 1,..,m-1$.

Hence, we can express the block feature denoted by FoB as an 8-dimension vector

$$FoB = \begin{pmatrix} \alpha_{x_{ij}} & \beta_{x_{ij}} & \alpha_{x_{i(j+1)}} & \beta_{x_{i(j+1)}} & \alpha_{x_{(i+1)j}} & \beta_{x_{(i+1)j}} & \alpha_{x_{(i+1)(j+1)}} & \beta_{x_{(i+1)(j+1)}} \end{pmatrix} \tag{4}$$

and it is utilized as an input vector of SVM.

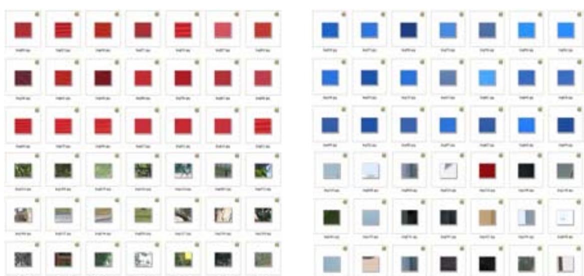

Fig. 3. Some examples of training data built by K-Means

In the training stage, the absence of standard traffic sign dataset is the main reason to build our dataset. The traffic-sign images are collected from the Internet with various states such as color distortion, blur, highlight as well as nightlight color, and so on. K-means method is applied to cluster block feature FoB in eq. (4) for building the database of interested color blocks. In this case, there are two groups (Red and Blue) of interested color blocks. In Fig. 3, the left group of images is used to learn the red color and the right one for the blue color. Each small image in Fig. 3 is a set of (2x2) blocks extracted from K-Mean algorithm. The three top lines of subgroup are sample sets of interested color and the other lines are set of non-interested color.

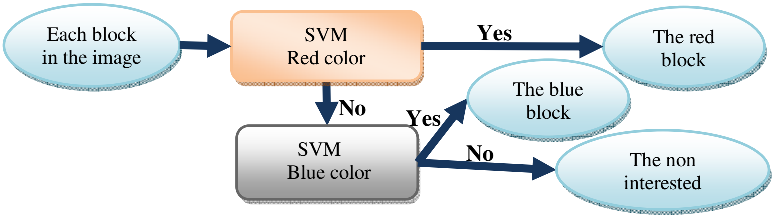

Fig. 4. The flow chart of using SVM to detect the interested blocks

2.1.2 SVMs for Color Classification

SVM is proposed by Vapnik and his group at AT&T Bell laboratory in 1992 [10]. From the database built in 2.1.1, let x denote a feature vector of a block - *FoB, y* denote a label of classification, $y \in \{1, -1\}$, b denote a bias value, and w denote the hyperplane of SVM. The primal formulation of SVM is to minimize $\|w\|_2$ such that the constraint of $y_k(w^T x_k + b) \geq 1$ is satisfied with all *k = 1, 2,.., N*. Applying Lagrange multipliers, we have the formula for the primal problem:

$$L(w,b,\alpha) = \frac{1}{2} w^T w - \sum_{k=1}^{N} \alpha_k (y_k (w^T x_k + b) - 1) \tag{5}$$

Therefore, the solution for this problem

$$\max_{\alpha} \min_{w,b} L(w,b,\alpha) \tag{6}$$

And the equivalent dual problem in the Lagrange multipliers α_k which can be solved in the quadratic programming is as follows:

$$\max_{\alpha} J_D(\alpha) = -\frac{1}{2} \sum_{k,l=1}^{N} y_k y_l x_k^T x_l \alpha_k \alpha_l + \sum_{k=1}^{N} \alpha_k \tag{7}$$

$$\text{s.t} \quad \sum_{k=1}^{N} \alpha_k y_k = 0 \tag{8}$$

The strong point of the dual problem is sparseness property. It means that most of $\alpha_k = 0$ except support vectors. Therefore, the complexity of the testing step is not large. Equation (10) presents the testing process of one block x.

$$y(x) = sign(\sum_{k=1}^{\#SV} \alpha_k y_k x_k^T x + b) = \begin{cases} 1, & \text{Red or Blue} \\ -1, & \text{Others,} \end{cases} \tag{9}$$

where the values of y_k, b, and α_k are the results of the learning step. The value of x_k is support vectors selected from the learning data. Note that we have two groups of (y_k, α_k, b, x_k), where one is correspondent to Red and the others for Blue. Figure 4 presents a process of retrieving Red and Blue blocks in an image by using eq. (9) of SVMs. Through experiments, we realize that the linear SVM returns a high accuracy with a low complexity to classify blocks of interested color - *FoB*. Although non-linear SVM can return a little better result, its calculation complexity is too large to apply on a real-time process. Therefore, we select the linear SVM instead of using non-linear model.

In practice, we propose a pre-processing step with the rough threshold γ on the HSL color space to enhance the performance of calculation. Its values are presented in eq. (10)-(11).

$$\gamma(red) = \begin{cases} \begin{cases} saturation \geq 51 \\ \left[\begin{array}{l} hue \leq 9 \\ hue \geq 170 \end{array} \right. \rightarrow red \end{cases} \\ others \rightarrow non-red \end{cases} \tag{10}$$

$$\gamma(blue) = \begin{cases} \begin{cases} saturation \geq 51 \\ 99 \leq hue \leq 112 \end{cases} \rightarrow blue \\ others \rightarrow non-blue \end{cases} \tag{11}$$

The rough threshold γ is applied to the value of Red and Blue color defined on the HSL space with each pixel of (2x2) block. As a result, the SVM method only classifies blocks which are positive error or correct answer of the pre-processing. This step helps to reduce the number of blocks which will be checked by SVM, and it makes the complexity of detection lower.

2.1.3 Grouping Region

After color-interesting blocks are selected by the classification on SVM, we make an extension of breadth-first-search (BFS) algorithm with pre-defined distance to group controlled block and retrieve candidate regions. We define the region $r = (x_{ij}, w, h)$, where x_{ij} denotes the top left pixel of region, w and h are the width and height of the region, respectively. So, the distance of two regions $r = (x_{ij}, w, h)$ and $r' = (x'_{ij}, w', h')$ is defined as the following;

$$DoR(r,r') = \begin{cases} \infty & T_1 = \emptyset \wedge T_2 = \emptyset \\ 0 & T_1 \neq \emptyset \wedge T_2 \neq \emptyset \\ \min(|i-i'-w'|,|i+w-i'|) & T_1 = \emptyset \wedge T_2 \neq \emptyset \\ \min(|j-j'-h'|,|j+h-j'|) & T_1 \neq \emptyset \wedge T_2 = \emptyset \end{cases} \tag{12}$$

With $T_1 = [i, i+w] \cap [i', i'+w']$ and $T_2 = [j, j+h] \cap [j', j'+h']$. Using eq. (12) to calculate distance of two regions, we have a value of *DoR*. If *DoR* is equal or smaller than the pre-defined distance *d*, the two regions will have a connection. Then, applying the extension of BFS algorithm can reduce the number of candidate regions to only a few color-interested regions from a set of color-interested blocks. As a result, we only need to recognize a few color-interested regions instead of a large number of all color-interested blocks in the classification phase. Figure 5(a)-(h) present an instance of the BFS result. Each red block in Fig. 5 is a block of interested color which is classified by the linear SVMs. The BFS algorithm will group color-interested blocks in Fig. 5(a) to retrieve a color-interested region bounded by a rectangle in Fig. 5(f). In this instance, the pre-defined distance *d* is set to *2*.

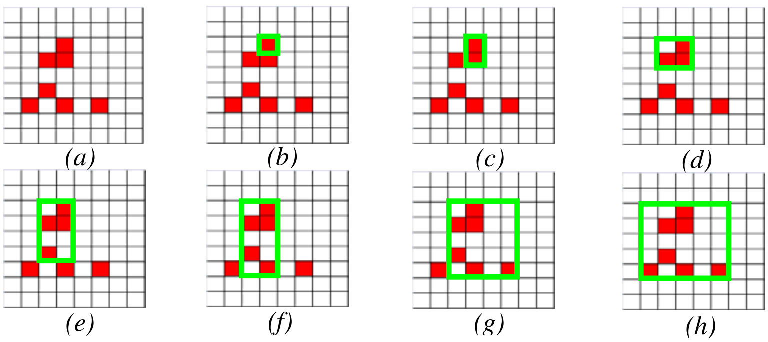

Fig. 5. Group region algorithm – extension of BFS with predefined distance (d=2)

Based on the position of car, we set a location of traffic signs positioning in the upper two-third part of images. Figure 6(a), (b), and (c) presents one result of our color detection by using SVM. This test is applied on database provided by the Research Center for Integration of Advanced Intelligent Systems and Devices, Toyota Technological Institute, Japan.

a) Original image b) Blue classification c) Red classification

Fig. 6. Example of color detection and segmentation using SVM on block of pixels

2.2 Using Hough Transform and Contour to Verify the Candidate Regions

After retrieving candidate regions, we apply Hough transform to detect circle traffic signs. In case of triangle traffic signs, we utilize Canny-edge detection with multi thresholds to get binary candidate regions and apply contour detection algorithm to extract geometry properties such as the number of edges and the size of angles between two lines for triangle verification. The complexity of classification phase depends on the number of candidate regions. Since the area of candidate regions is small, the complexity is small enough to process in real time. In our system, this classification phase averagely consumes less than 30 milliseconds per frame. Figure 7 presents the flow chart of traffic sign segmentation by using Hough transform and contour algorithm. It shows that one traffic sign is detected by both color and shape in our method.

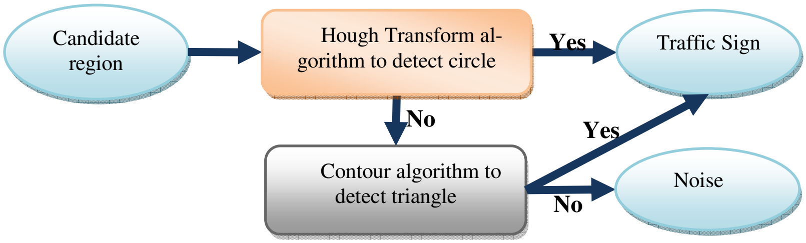

Fig. 7. Hough Transform and contour algorithm to verify candidate region

3 Experiment and Result

We utilize C language on a computer - 1.66 GHz core dual CPU and non-preprocessing images for our experiments. SVM-Light [11] is utilized to evaluate the proposed method.

Table 1. Results of linear-SVM training process

	Red color	**Blue color**
Positive samples	30000	30000
Negative samples	45000	60000
Recall	88,24%	98,03%
Precision	88,28%	98,03%

The learning data is defined in Table 1. After preprocessing with rough threshold γ in HSL, we set the value of γ such that there are only positive errors and color-interested blocks as outputs, so we have a bias in favor of negative samples towards positive ones. By using preprocessing with rough threshold γ in HSL, the algorithm is speeded up approximately 5 times. We can see the advantage of pre-processing step on the time consuming in Table 2. In addition, the average time consuming for all process in an image (640x480) is just about 50 milliseconds. It means that the proposed method can process about 20 fps.

Table 2. Time consuming of classifying color-interested blocks in an image (320x240)

	Linear SVM (Only)		**Linear SVM + pre-processing γ**	
	Red color	Blue color	Red color	Blue color
Worse case	0.14s	0.14s	0.032s	0.032s
Best case	0.09s	0.09s	0.015s	0.015s
Average case	0.11s	0.11s	0.023s	0.023s

To evaluate our method, we collect sampling images on the internet, some images in [4] as well as images of TTI lab to build a challenging database which contains the diversified images collected in many conditions of environment such as bad light, snow fall, or blurred, faded, occluded, and damaged traffic signs, and it is also published for free at http://www.fit.hcmus.edu.vn/~lttam/LTTam-TestingDatabase.rar

Table 3. The result of testing on the database

	# Images	# Traffic Signs	# False Negative	# False Positive	% False Negative	% False Positive
Bad light	30	45	3	1	6.67 %	2.22 %
Blurred	120	161	7	7	4.35 %	4.35 %
Faded	40	65	11	2	16.92 %	3.08 %
Noise	23	42	7	9	16.67 %	21.43 %
Occluded	36	66	8	6	12.12 %	9.09 %
Damaged	20	33	5	3	15.15 %	9.09 %
Snowfall	30	35	1	5	2.86 %	14.29 %
TTI campus	67	99	3	1	3.03%	1.01%
JP urban road	39	51	2	8	3.92%	15.69%
Total	**405**	**597**	**47**	**42**	**7.87%**	**7.04%**

Table 3 is the testing result of our proposed method on this database, the precision is 92.91% and the recall is 92.13%. Our result shows that the color classification using linear SVM gives a robust result in conditions of daylight, bad light, and snowfall.

We present some illustrating results of our test in Fig. 8 (a), (b), and (c) for the foggy, snowy weather, and Japanese urban road, respectively. Our algorithm utilized SVM for color detection combined with Hough transform for circle detection and contour algorithm for triangle detection. The result in Fig. 8 shows that our method can detect a far, small, and blurred traffic sign, where the camera is mounted inside of the car.

a) Foggy weather

b) Snowy weather

c) Japanese urban road

Fig. 8. Some examples from the testing database

Table 4 shows our comparison to previous works based on our database. The result shows that our method returns a high accuracy in a real time processing (average 20 fps) for autonomous-driving system.

Table 4. Results of our method and some related methods

	Detection Rate	False alarm Rate	Time consuming
Soetedjo [2]	85%	10%	2440 ms
Shneier [9]	88%	58%	50 ms
Our method	**92.91%**	**7.04%**	**50 ms**

4 Conclusions

In this paper, we presented a real-time processing method of traffic-sign detection to apply in autonomous driving system. Our proposed method utilized linear SVM to classify color by a low complexity (average 23 milliseconds per frame). After that shape matching has been applied to eliminate positive errors. We achieved 92.91 percent of detection accuracy and it has been applied on real-time autonomous driving system with the processing speed of 20fps, where the maximum speed of car is limited at 30 km per hour. In the near future, we will combine our processes of detection and recognition to generate a vision-based system of guidance and warning for the autonomous driving system.

References

1. Paclik, P., Novovicova, J., Pudil, P., Somol, P.: Road sign classification using Laplace kernel classifier. Pattern Recognition Letters 21, 1165–1173 (2000)
2. Soetedjo, A., Yamada, K.: An Efficient Algorithm for Traffic Sign Detection. Journal of Advanced Computational Intelligence and Intelligent Informatics 10(3), 409–418 (2006)
3. de la Escalcra, A., Moreno, L., Puente, E., Salichs, M.: Neural traffic sign recognition for autonomous vehicles. In: Proc. 20th Inter. Conf. on Industrial Electronics Control and Instrumentation, Bologna, Italy (1994)
4. Fleyeh, H.: Traffic and road sign recognition. PhD thesis at Dalarna University, Sweden (2008)
5. Fang, C., Fuh, C., Chen, S., Yen, P.: A road sign recognition system based on dynamic visual model. In: Proc. IEEE Computer Society Conf., Computer Vision and Pattern Recognition (CVPR), Madison, Wisconsin (2003)
6. Satumino, M.-B., Sergio, L.-A., Pedro, G.-J., Hilario, G.-M., Francisco, L.-F.: Road-sign detection and recognition based on support vector machines. IEEE Transactions on Intelligent Transportation Systems 8(2), 264–278 (2007)
7. Kiran, C.G., Lekhesh, V.: Prabhu, Abdu Rahiman V., Rajeev K.: Traffic Sign Detection and Pattern Recognition Using Support Vector Machine. In: Seventh International Conference on Advances in Pattern Recognition, pp. 87–90 (2009)

8. Ebner, M.: Color constancy using local color shifts. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3023, pp. 276–287. Springer, Heidelberg (2004)
9. Shneier, M.: Road Sign Detection and Recognition. In: Proceeding of SPIE, the International Society for Optical Engineering, Florida, USA (2006)
10. Boser, B.E., Guyon, I.M., Vapnik, V.N.: A training algorithm for optimal margin classifiers. In: Haussler, D. (ed.) 5th Annual ACM Workshop on COLT, Pittsburgh, PA, pp. 144–152 (1992)
11. Joachims, T.: Making large-Scale SVM Learning Practical. MIT Press, Cambridge (1999); Burges, C., Smola, A.: Advances in Kernel Methods – Support Vector Learning. MIT Press, Cambridge (1999)