2010 International Conference on Pattern Recognition

REAL-TIME TRAFFIC SIGN DETECTION: AN EVALUATION STUDY

Ying Li, Sharath Pankanti

IBM T.J. Watson Research Center, NY USA

Weiguang Guan

RHPCS, McMaster University, Canada

ABSTRACT

This paper presents an experimental evaluation of three different traffic sign detection approaches, which detect or localize various types of traffic signs from real-time videos. Specifically, the first approach exploits geometric features to identify traffic signs, while the other two are developed based on SVM (Support Vector Machine) and AdaBoost learning mechanisms. We describe each of the three approaches, conduct a detailed comparison among them, and examine their pros and cons. Our conclusions should lead to useful guidelines for developing a real-time traffic sign detector.

1. INTRODUCTION

Automatic traffic sign recognition (TSR) has been attracting many researchers' attentions in recent years. In fact, it makes one important application for advanced driver assistance systems (ADAS). Two stages, namely, detection and recognition, are usually employed by existing TSR systems. For this work, we focus on the detection part.

Many techniques have been proposed to detect traffic signs, which can be grouped into the following three categories: 1) applying both color and edge information; 2) applying shape information; and 3) applying machine learning mechanism. A common approach among the work in the first category ([4], [5]) usually applies a color segmentation first, followed by some edge detection. Then, specific shapes are identified from the edge map using various techniques such as RANSAC and Hough transform. Comparatively, work in the second category relies on pure edge information. For instance, [8] proposes a fast circle detector that applies rigid pairing rules on gradient vectors. In [2] and [6], a fast algorithm based on radial symmetry is implemented, which could be adapted to various regular shapes including triangle, square, diamond, octagon, and circle. It operates on the gradient of a gray scale image and exploits the geometric nature of shapes.

Finally, work in the last category applies machine learning techniques. Neural Networks, Support Vector Machine (SVM) and AdaBoost are among the most commonly used ones. For instance, [7] applies linear SVM to classify blobs into different shapes, once they are color-segmented from the input image. In contrast, Bahlmann et al. applies AdaBoost to detect signs using a set of Haar wavelet features that take both color and position into account [1].

For this work, we aim to investigate a few different traffic sign detection (TSD) approaches and evaluate their effectiveness and efficiency. In particular, the first approach exploits geometric features to identify traffic signs as such approach is more robust to varying lighting conditions than those based on color and edge information. We chose the other two to be learning-based, namely, SVM- and AdaBoost-based, as such approach tends to be more flexible or tolerable in terms of the object's shape and appearance than the first approach. Moreover, it would be interesting to compare the performance of the two popular learning approaches. In the end, we hope that our conclusion would lead to useful guidelines for developing a real-time traffic sign detector, which is critical for a successful TSR system.

Figure 1 shows some examples of traffic signs that we explored in this work. Specifically, we aim to detect the following three categories of German traffic signs: 1) red-bordered circular signs; 2) red-bordered triangular signs; and 3) inverted red-bordered triangular signs. Note that all signs are required to be detected at varying sizes from varying distances.

Fig. 1. Some examples of traffic signs in problem.

2. FEATURE-BASED TRAFFIC SIGN DETECTION

The basic idea of this feature-based TSD approach comes from the radial symmetry theory described in [2]. Specially, for circles that are radially symmetric, it is observed that, for a given pixel p, and its gradient $\vec{g}$ calculated using some edge operator that yields orientation, if p lays on the arc of a circle, then the center of the circle would be in the direction of gradient $\vec{g}$ and at the distance of its radius (denoted by r).

Based on this theory, for any edge pixel p in a Sobel edge map, we can estimate the center of the circle that p possibly

1051-4651/10 $26.00 © 2010 IEEE
DOI 10.1109/ICPR.2010.743

belongs to, by casting a vote to the pixel at a radius r along or against the direction of its gradient. Intuitively, the more votes that a pixel gets from other pixels, the more likely it is a true circle center. Figure 2 (a) shows an image with a circular sign, and (b) shows the voting image when the searching radius is set to 11. As we can see, pixels that surrounding the center of the circular sign are much brighter than the others.

(a) (b) (c) (d)

Fig. 2. (a) An input image, (b) its voting image when $r = 11$ for circular sign detection; (c) an input image, and (d) its shape response image when $r = 15$ for triangular sign detection.

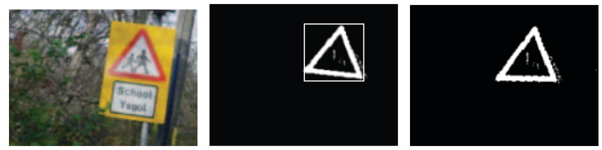
We apply similar ideas to detect triangular signs with additional knowledge about the pattern of the edge orientations. Specifically, assuming that a triangle inscribes upon a circle with r denoting its radius, we detect the sign based on the following two cues: 1) for each edge pixel p that lies on a triangle, its gradient points to a straight line that goes through the center and is at the distance of r. In other words, a line of votes is cast describing possible shape centroid positions that would account for the observed gradient element; and 2) triangular sign is equi-angular. Consequently, a rotationally invariant measure is employed to check how well a set of edges fit a particular angular spacing. Due to the space limitation, readers are referred to [6] for more details.

Figure 2 (c) shows an input image, and (d) shows its shape response image obtained after exploiting the aforementioned cues. As we can see, pixels in the neighborhood of the triangle centroid are standing out from the background.

3. SVM-BASED TRAFFIC SIGN DETECTION

The idea of using SVM to detect traffic signs comes from the observation that all target signs have enclosed red borders. Consequently, we can treat them as blobs in the red channel of the image, and classify them accordingly.

The SVM learning process starts with extracting the binarized red channel I^R from an input frame I. This is performed in HSI color space. We then perform a connected component analysis on I^R to obtain a list of blobs. Figure 3 (a) shows an input frame. Its binarized red channel is shown in (b), where the detected blob is bounded by a white rectangle.



(a) (b) (c)

Fig. 3. (a) An input frame I, (b) detected blob in I^R, and (c) orientation-corrected triangular sign.

The detected blobs are then matched with annotated ones in the ground truth. Moreover, if it is a triangular sign, then the relative positioning of its left and right vertices is examined to verify that it has the right orientation. Otherwise, necessary operations will be performed to rotate it back to its right position. One such example is shown in Figure 3 (c). Next, forty features are extracted from every matched blob to form a training sample. Specifically, the first twenty features represent the equally sampled distances from the left edge of the blob to that of its bounding box (termed as *left distances*). Referring the set of distances calculated in the same fashion but from the right as the *right distances*, the second twenty features capture the differential distances between the corresponding left and right distances. Finally, when all training samples are obtained, three binary SVM models are trained for classifying circles, triangles and inverted triangles, respectively. The LibSVM tool [3] is used in this case with the kernel chosen to be RBF (Radial Basi Function).

During the traffic sign detection, we first follow the same steps to extract red channel I^R. Then blobs are extracted from both I^R and its morphologically dilated version so as to pick up all possible red blobs. Next, we filter out blobs that are unlikely of circular or triangular shape, by examining if certain small corner areas within the blob's bounding box only contain background. This is a critical step as it distinguishes triangular objects from others, speeds up the process and helps reduce the false alarm.

Potential triangular blobs then undergo possible orientation correctness, and features are extracted from each blob to form the test sample. Finally, the three pre-trained SVM classifiers are applied to each test sample, and the class that has the the highest probability is returned.

4. ADABOOST-BASED TRAFFIC SIGN DETECTION

The features we used for AdaBoost learning contain five classical Haar wavelets and four HoG (Histogram of Oriented Gradient) features. Each HoG feature captures the gradient strength along one specific direction such as the horizontal or vertical direction.

During the training process, a cascade classifier is trained which contains a cascade of rejectors. At each layer of this classifier, we use AdaBoost to train an ensemble classifier based on a set of weak classifiers. In the end, as similar to the SVM-based approach, three cascade classifiers will be trained.

During the classification process, given a frame, it is first gray-scaled and subsequently scanned with a sliding window of different sizes at different locations. The image patch within each window is then extracted and classified.

5. PERFORMANCE EVALUATION

To evaluate the performance of the above three approaches, we collected 24 videos using a camera mounted on the windshield of a vehicle. In total, there are 83482 frames, amounting to approximately 1 hour. The frame resolution is 640×480. The scenarios in the videos include urban, rural, highway and freeway traffic, taken during daytime. Weather conditions are cloudy and sunny.

Within this test set, there are in total 113 red circular signs, 53 red triangular and 14 inverted red triangular signs. Note that while a sign almost always appears in a sequence of consecutive frames (we term it as *sign sequence*), we call such case as one sign with multiple occurring instances.

For each test video, we manually annotated all traffic signs by labelling their bounding boxes. Signs that are smaller than 20×20, occluded, or not clearly visible are ignored. On the other hand, signs that completely reside in shadow are also annotated and required to be detected.

5.1. Performance Metrics

We use the following two criteria to validate a *match* between an annotated sign and a detected sign: a) the intersection between them should be greater than 70% of the annotated bounding box; and b) the non-intersection part of the detected bounding box should be less than 30% of the annotation. In other words, we require that the detected bounding box be as close and as tight to the true object as possible.

Moreover, considering that a sign does not need to be matched (or detected) in every frame of its sign sequence in order to be recognized, we declare that a sign is correctly *detected* if it has been matched for at least 50% frames of its sign sequence. Otherwise, we declare that the sign is missed, even if it indeed has been matched for some number of frames. In this context, we report the *detection rate*, that is, the number of detected signs over the total number of signs, as one of the performance measurements. Another metric is the *false positive rate*, defined as the number of falsely detected signs over the total number of frames. While it is not as important as the detection rate, it should be kept as low as possible.

Finally, to make our work comparable to others, we have also measured the performance in terms of *frame detection rate*, which is defined as the number of frames where a traffic sign is detected and matched with the ground truth, over the total number of frames where a traffic sign exists. This is the common metric employed by most existing work.

5.2. Performance Report

The training data set for SVM and AdaBoost learning contains 871, 702 and 470 positive samples for circular, triangular and inverted triangular signs, respectively. As for the negative samples, we use non-annotated blobs for the SVM approach, while in the case of AdaBoost, certain number of unrelated image patches are randomly selected at each training layer.

Table 1 tabulates the evaluation results for the three approaches, where we have reported individual detection results as well as the overall detection rate (*i.e.* the combined column in the table). We also report the processing speed in terms of frames per second, which is achieved on a 2.67 GHz Intel Xeon processor with 1G RAM. Finally, the false positive rate is reported in the last column.

From the table we see that the AdaBoost-based approach has outperformed the other two for all three sign categories. It has the highest detection rates and the lowest false positive rate. On the other hand, the SVM-based approach has the fastest processing speed (around 5 frames per second), due to its single scan at the blob level. Moreover, while its current performance is already quite acceptable, it can be further improved with a more sophisticated feature set.

As for the feature-based approach, the detection rate is fairly good for circular signs, yet it drops significantly with triangular signs. It also has the highest false positive rate. One major reason for that is, while its underlying principle of detecting triangular signs is very neat and theoretically sound, it fails to reject samples that have a randomly formed triangular shape (*e.g.* the shape formed by tree branches), which happens more often than the circular shape. On the other hand, as this approach is built upon edge information, a traffic sign that presents weak edges might be missed.

It is also observed that all approaches tend to achieve a higher detection rate on the inverted triangular signs than on the triangular ones. This is because that there is only one type of inverted triangular sign, as compared to over 30 variations on triangular ones. On the other hand, the circular signs, when under the same driving and lighting conditions, tend to be easier detected than the triangular ones.

As a summary, compared to the machine-learning based approaches, the feature-based approach runs at the lowest speed, due to its diligent search of signs within a large range of pre-defined radius sizes. It also has the lowest detection rate. Nevertheless, this approach does not need any time-

Table 1. Performance of the three traffic sign detection approaches, where FPS stands for frames per second.

Approach	Total Number of Detected Signs			Sign Detection Rate				Processing Speed (FPS)	False Positive Rate
	Circle	Triangle	Inverted Triangle	Circle	Triangle	Inverted Triangle	Combined		
Feature-based	105	32	12	93%	60%	86%	83%	0.53	1.5
SVM-based	102	41	12	88%	80%	86%	86%	5.6	0.74
AdaBoost-based	108	44	14	96%	83%	100%	92%	0.69	0.25

consuming and labor-intensive human annotations to start off, which is very attractive when designing TSR systems with a light-weight detection component.

Table 2 tabulates the frame detection rates for the three approaches. Again, it confirms with our earlier observation that the AdaBoost-based approach performs the best, while the feature-based approach lags behind the other two. Nevertheless, the differences between them are not major.

Table 2. The frame detection rate of the three traffic sign detection approaches.

Approach	Frame Detection Rate			
	Circle	Triangle	Inverted Triangle	Comb.
Feature-based	81%	56%	78%	72%
SVM-based	79%	72%	83%	77%
AdaBoost-based	88%	74%	97%	85%

Finally, we would like to comment on the generalizability of the three proposed approaches, specifically, how easily they can be extended to detect other types of traffic signs. Considering that the AdaBoost-based approach works on gray-scale image (although it can be easily extended to include color information as necessary), and does not have any prerequisite for feature extraction, it can be readily applied to any other categories of traffic signs. On the other hand, while the feature-based approach does not exploit color information either, it does require a traffic sign to be radial symmetric and equi-lateral. This would limit the set of traffic signs that it can be applied to. As for the SVM-based approach, while it has no preference on the shapes to be detected, the current implementation is founded on blobs in red channel. Consequently, it will not work for non-red traffic signs. Nevertheless, with some additional work on extracting the corresponding color channels (*e.g.* blue, yellow, or gray), and a careful selection of appropriate blob features, the same SVM training and testing mechanism can be readily reused.

6. CONCLUSION

An experimental evaluation of three different traffic sign detection approaches, namely, feature-based, SVM-based and AdaBoost-based, is presented in this paper. To our knowledge, this is the most comprehensive performance evaluation over existing work on traffic sign detection, in terms of either the variety of targeted traffic signs or the amount of test data. It is our hope that the conclusion of this work would lead to useful guidelines for developing a real-time traffic sign detector.

7. REFERENCES

[1] C. Bahlmann and et al. A system for traffic sign detection, tracking, and recognition using color, shape and motion information. *IEEE Intelligent Vehicles Symposium*, 2005.

[2] N. Barnes and A. Zelinsky. Real-time radial symmetry for speed sign detection. *IEEE Intelligent Vehicles Symposium*, 2004.

[3] C. C. Chang and C. J. Lin. LIBSVM–A library for support vector machines. *http://www.csie.ntu.edu.tw/ cjlin/libsvm*.

[4] M. Eichner and T. Breckon. Integrated speed limit detection and recognition from real-time video. *IEEE Intelligent Vehicles Symposium*, 2008.

[5] A. Lorsakul and J. Suthakorn. Traffic sign recognition using ueural network on OpenCV: Toward intelligent vehicle/driver assistance system. *Journal of Intelligent Service Robotics*, page 335, 2007.

[6] G. Loy and N. Barnes. Fast shape-based road sign detection for a driver assistance system. *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 70–75, 2004.

[7] S. Maldonado-Bascon and et al. Road-sign detection and recognition based on Support Vector Machine. *IEEE Transactions on Intelligent Transportation Systems*, 8(2), 2007.

[8] A. Rad, K. Faez, and N. Qaragozlou. Fast circle detection using gradient pair vectors. *7th Digital Image Computing: Techniques and Applications*, 2003.