

THE SPEED-LIMIT SIGN DETECTION AND RECOGNITION SYSTEM

Kuo-Hsin Tu (塗國星), Chiou-Shann Fuh (傅楸善)

Dept. of Computer Science and Information Engineering,
National Taiwan University, Taiwan

E-mail: p04922004@csie.ntu.edu.tw, fuh@csie.ntu.edu.tw

ABSTRACT

This paper proposes a system that detects and recognizes speed-limit signs automatically. The system detects potential speed-limit sign regions by detecting maximally stable extremal regions (MSERs). The recognition is achieved by using a Support Vector Machine (SVM) classifier. The proposed system provides robustness and high accuracy in detecting and recognizing speed-limit signs when input images have different lighting conditions and blurriness.

Keywords: *Advanced Driver Assistance Systems (ADAS), Traffic Sign Recognition (TSR), object recognition, maximally stable extremal regions (MSERs), support vector machines (SVMs), histogram of oriented gradient (HOG)*

1. INTRODUCTION

The Advanced Driver Assistance Systems (ADAS) is considered as one of the fastest growing areas in automotive industries recently. Traffic Sign Recognition (TSR) system, especially the vision-based speed-limit sign recognition system is an essential part to an ADAS. A fast and efficient automatic speed-limit sign recognition system could warn the driver in case of unawareness about the present speed limit and thus provide significant contribution for improving safety. Moreover, a reliable speed-limit sign recognition system could also bring important benefits to improve other conventional ADAS features such as Adaptive Cruise Control (ACC) system.

Generally, the speed-limit sign recognition system can be divided into two separate tasks: detection and recognition. In the detection stage, signs are detected from the image, and the values specified in the signs will be identified during the recognition stage.

In the detection stage, color and shape information are commonly used. As distinct colors (red) are used in speed-limit signs (Fig. 1), segmentation using color information is a common approach. Color models such as Hue-Saturation-Value (HSV) and YUV [1, 2], have

been used. These models can handle scenes with strong illumination and poor lighting.

On the other hand, shape based detection methods take advantages of the special appearance of speed-limit signs (circle shape) to detect them. For circular signs, the circular Hough Transform (HT) is the preferred algorithm, but the algorithm requires more computation time and memory. Loy and Zelinsky introduced the Radial Symmetry Detector (RSD) [3]. More recently, Barnes and Zelinsky [4] show that the algorithm of RSD can handle noises while maintaining high detection rates and having real-time performance. In addition to color and shape based detection methods, some researches have used MSERs to detect traffic signs [5, 6].

In the recognition stage, template matching or learning-based classification techniques are usually used. Malik et al. [1] presented a template matching based recognition module to recognize traffic signs. For learning-based classification, Artificial Neural Networks and Support Vector Machines (SVMs) have been widely investigated [7, 8, 9]. Recently, Convolutional Neural Network (CNN) has been used to recognize traffic signs and has high accuracy [10, 11], however, huge computation resource is usually required either in training or test phases.

In this paper, we introduce an automatic detection and recognition system of speed-limit signs that can handle different lighting conditions and blurriness in images. The system uses a detection method based on MSER detection and recognizes speed-limit signs with Histogram of Oriented Gradient (HOG) features by a SVM classifier.



Fig. 1: A typical speed-limit sign

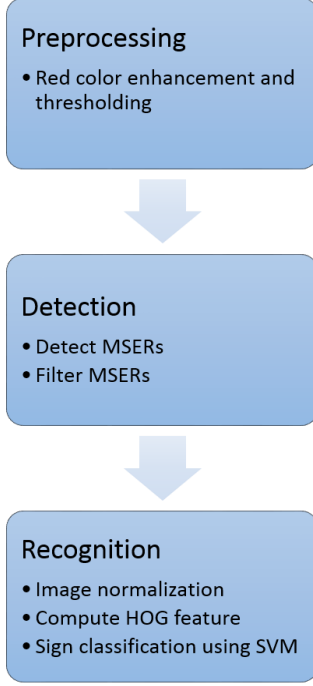


Fig. 2: Pipeline of the proposed system.

2. METHODS

2.1. Overview

The proposed speed-limit sign recognition system has three stages: preprocessing, detection, and recognition. The system pipeline is shown in Fig. 2. The first step, the preprocessing stage, enhances the input image so that the speed-limit sign can be more obvious. The second step, the detection stage, uses MSER detection as described by Matas *et al.* [12] to find speed-limit sign candidate regions. Finally, the last step, the recognition stage, extracts HOG features from the candidate regions and feeds it to a SVM and the class of a speed-limit sign will be classified.

2.2. Preprocessing

Before detection, the input image needs to be processed for the detection stage to work properly and efficiently. Since the input to the MSER computation requires grayscale image, we need to convert the input RGB image to a grayscale one. Also, the speed-limit signs are designed to have bright red colors, so we can take advantage of this feature and emphasize the pixels that have dominant red channel values. Therefore, we adapt the red color enhancement approach proposed by Ruta *et al.* [13] and combine the red thresholding method [14]. While applying red color enhancement, we also threshold the input image using a value determined empirically. This will reduce the search space that potentially contains speed-limit signs and the number of

false alarms of the detector will also decrease. For each pixel $\mathbf{x} = [x_R, x_G, x_B]$ in the image, where x_R , x_G , and x_B are red, green, and blue channels of the image, the red color enhancement is provided by

$$f_R(\mathbf{x}) = \max\left(0, \frac{\min(x_R - x_G, x_R - x_B)}{x_R + x_G + x_B}\right)$$

And the final grayscale image can be obtained by

$$Red(\mathbf{x}) = \begin{cases} f_R(\mathbf{x}), & x_R \geq x_G \text{ and } x_R \geq x_B \\ & \text{and } x_G/(x_R - x_G) \leq T_R \\ 0, & \text{otherwise} \end{cases}$$

Where T_R is a parameter that obtained from experiment results. The grayscale image $Red(\mathbf{x})$ obtained from this way has the advantage that the red colors of speed-limit signs become more apparent. Fig. 3 shows input images and their corresponding results after the preprocessing steps (Fig. 4). We can see that the red channel is enhanced and the most part of the image are ignored.



Fig. 3: Input images.



Fig. 4: Preprocessed input images.

2.3. Detection

In the detection stage, the system will detect MSERs and find potential speed-limit sign regions among them. The main advantage of this approach is that the detection of MSERs is robust under different environmental conditions, such as lighting and weather of the scene or the complexity of the background. MSERs are regions in an image that maintain their shapes when the image is thresholded at several levels. The detection of MSERs is performed on a grayscale image that comes from the result of the preprocessing stage. The grayscale image is binarized by a number of thresholds, and connected components are found in each binarized image. These connected components are recorded and analyzed. Finally, MSERs are detected as connected components that maintain their shapes through different threshold values (Fig. 5). Fig. 6 shows the results of MSER detection on the preprocessed images in Fig. 4, and the detected MSERs are colored with different colors. As it is shown, the circle shapes of speed-limit signs can be detected as MSERs and they are also candidates for recognition. After MSERs are found the connected components that represent these MSER are also found. Since there may be some false candidates in the resulting MSERs, we will filter the result and reduce the number of candidate regions. To filter the result, several metrics of the candidate regions are used, e.g., aspect ratios and areas of the bounding boxes enclose these regions, which were determined empirically. After removing false candidates, the speed and accuracy of the recognition stage can be improved. Generally,

computation of MSERs can be expensive. In order to increase the processing speed, we will threshold at an appropriate step rather than thresholding at every possible value.

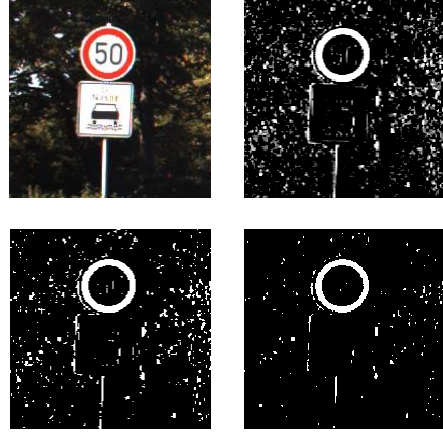


Fig. 5: The speed-limit sign maintains its shape under different thresholds. (Top-left) Original image. (Top-right) The preprocessed image. (Bottom-left) threshold = 150. (Bottom-right) threshold = 250.

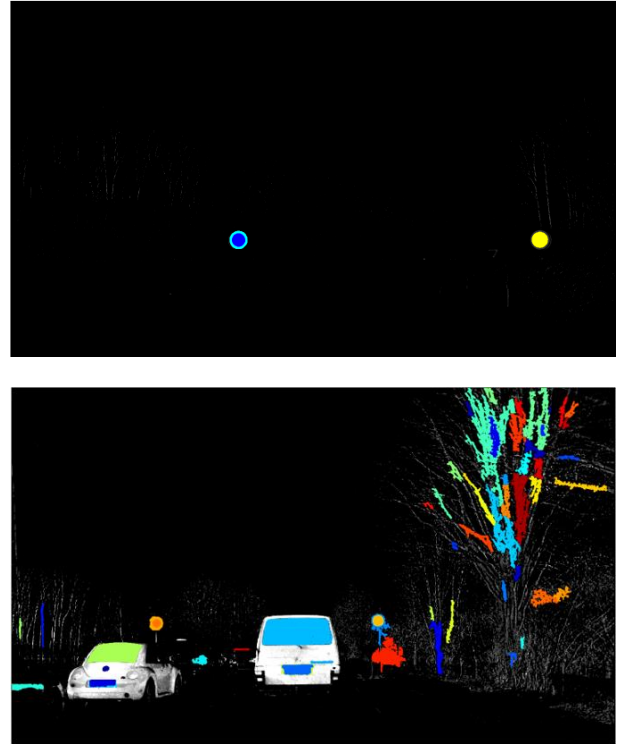


Fig. 6: Detected MSER regions from the preprocessed images.

2.4. Recognition

The recognition stage verifies a candidate region as a speed-limit sign. If the candidate region is actually a speed-limit sign, the speed-limit values on the sign will be determined. Otherwise, the candidate region will be

discarded. The input to the recognition stage is candidate regions that are cropped from the original image and resized to 64×64 pixel (Fig. 7). In the recognition stage, HOG features will be extracted from the candidate region. Since speed-limit signs have strong appearance—circle shapes and high-contrast edges, HOG features are very suitable to capture these characteristics. HOG feature is computed for each candidate region and a vector will be generated. To compute HOG feature, a candidate image is divided into cells with size 4×4 pixels and blocks with size 2×2 . A histogram of gradient orientation with nine bins will be generated for each cell. In the end, a vector of size 8100 will represent the HOG feature of the candidate region.

For speed and simplicity, SVM is used as a tool for the recognition stage. After HOG features are computed, the generated feature vectors will be the input to SVM for classification of speed-limit signs. Candidate regions are classified using a multiclass SVM. Generally, SVM is used as a binary classifier. However, a multiclass classification SVM can be realized by combining many one-against-one binary SVMs. We define speed-limit signs have certain speed-limit values as one class, for example, values such as 30, 40, and etc. In addition, we define a class for background, objects that are not speed-limit signs belong to this class. There are eight classes of speed-limit signs and one class of background (Fig. 9). Finally, a SVM will be trained for each class, and SVMs of all classes are combined to form a multiclass SVM.



Fig. 7: Input to the recognition stage.

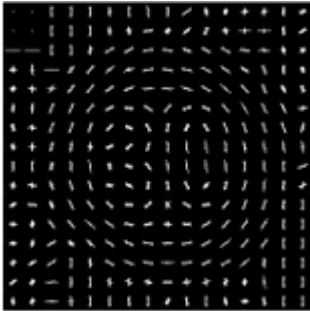


Fig. 8: HOG feature for Fig. 7.

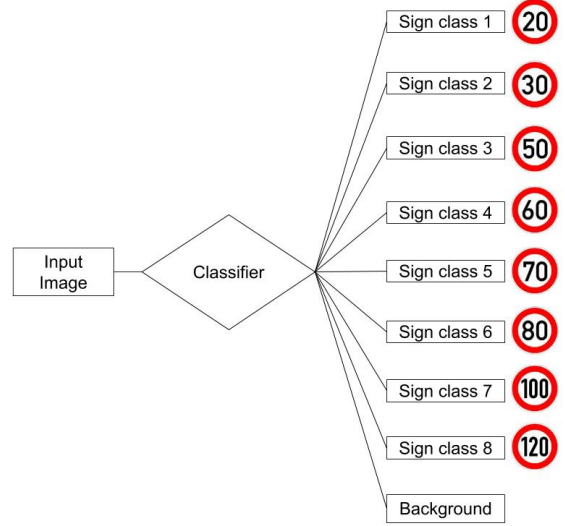


Fig. 9: Multiclass SVM for speed-limit sign recognition.

3. EXPERIMENTS AND RESULTS

3.1. Overview

The proposed system is implemented with Matlab 2015b and runs on a PC with Intel i7 3.4GHz CPU and 32GB of memory.

3.2. Datasets

In this paper, two public traffic sign datasets: the German Traffic Sign Recognition Benchmark (GTSRB) [15] and the German Traffic Sign Detection Benchmark (GTSDB) [16] are used. The GTSRB dataset is used to train our multiclass SVM and the GTSDB dataset is also used to generate negative (background) samples. In addition, the GTSDB dataset will be used to test the overall performance of our proposed system. The images from the GTSDB test set are challenging and the test set includes many images that are under poor lighting conditions or blurry (Fig. 10). These two datasets are chosen because they were subject to competitions. The datasets are reliable and they have diverse content and large amount of annotations. Although the GTSRB dataset has all kinds of traffic sign, it separates different classes of image data, so we can just take the speed-limit sign images and other similar sign images for our training purpose.



Fig. 10: Test image under poor lighting condition.



Fig. 11: Speed-limit sign images in the GTSRB dataset.



Fig. 12: Other traffic sign images with similar colors or shapes to the speed-limit sign in the GTSRB dataset.



Fig. 13: A sample test image in the GTSDDB dataset.

3.2. Training Setup

For training, we will use images from the GTSRB training set. First of all, we collect eight classes of

speed-limit sign images (Fig. 11) and label them as speed-limit signs during training, while other classes of traffic sign images might be labeled as backgrounds. In addition, we collect 4656 background image patches from the GTSDDB training set and label them as backgrounds. These negative samples are generated by randomly sampling in images from the GTSDDB training set and we remove speed-limit signs from the samples manually. Finally, these training samples are normalized to 64×64 pixels before computing their HOG features.

3.3. Testing Setup

For testing, we will use 300 images from the GTSDDB test set, each of which has the size of 1360×800 pixels (Fig. 13). During testing, we will run speed-limit sign detection on each test image and obtain candidate bounding boxes for further processing. We only consider a bounding box as strong candidate for further processing if the corresponding bounding box overlaps with at least 40% of the area covered by the ground truth. After running the speed-limit sign detector, we will run speed-limit sign recognition on these candidates. The location of each candidate bounding box will be used to extract image patch from the input image and it will be normalized to 64×64 pixels before recognition. Finally, the trained multiclass SVM classifier will be applied to these image patches, and we can determine which classes they belong to.

3.4. Experiment on different values of T_R

In this experiment, we try to find the optimal threshold value T_R for red thresholding in order to get the best performance for the detection stage. The evaluation of the detection stage is based on the recall and precision values that are computed as follows:

$$\text{Recall} = \frac{\text{Number of correctly detected signs}}{\text{Number of signs in the ground truth}} \times 100$$

$$\text{Precision} = \frac{\text{Number of correctly detected signs}}{\text{Number of detected signs}} \times 100$$

We have test different T_R values: 4.0, 5.0, 6.0, and 7.0, and the value 6.0 produces the best result. From Table 1 we can see that when T_R is 6.0 the detection stage detected the most signs. In addition, the recall value of T_R when it is 6.0 is the highest among all four values, while the precision value is slightly lower than other values of T_R .

Table 1: Results for different values of T_R .

T_R	4.0	5.0	6.0	7.0
Number of correctly detected signs	108	112	118	116
Number of detected signs	111	115	122	119
Number of signs in the ground truth	128	128	128	128
Recall (%)	84.37	87.50	92.19	90.62
Precision (%)	97.30	97.39	96.72	97.48

3.5. Experiment on different models for SVM

In the recognition stage, the trained model for the multiclass SVM can affect the result. Therefore, we experimented on different models and wished to find the model that brings the best performance. The evaluation of the classification stage is based on the precision values that are computed as follow:

$$Precision = \frac{\text{Number of correctly detected and classified signs}}{\text{Number of detected signs}} \times 100$$

In the experiment, we trained three different models: M_1 , M_2 , and M_3 . M_1 is a model that trained with eight classes of speed-limit signs images and one class of background images. The background class only contains negative samples that collected previously. M_1 , M_2 , and M_3 only differ in samples of background class. In M_2 , the background class includes both negative samples and images of six other classes of traffic sign that have similar color and shapes, e.g., red colors with circle or octagonal shapes (Fig. 12). M_3 is a model that trained with all classes of traffic sign images. The background class for M_3 contains all negative samples and images of all other classes of traffic sign except the speed-limit sign classes. From Table 2 we can conclude that M_2 has the best performance among all three models. The result of M_1 reflects that it has too few information to perform well in recognition of speed-limit signs, while the result of M_3 shows that it is overfitting, so that it does not has better performance than M_1 and M_2 .

Table 2: Results for different models.

Model	M_1	M_2	M_3
Number of correctly detected and classified signs	117	118	116
Number of detected signs	122	122	122
Precision (%)	95.90%	96.72%	95.08%

3.6. Overall Performance

The result of the proposed system is shown in Table 1. There are total 128 speed-limit signs in the ground truth among the 300 test images. Besides, we categorize the results to calculate the desired performance metrics. Signs detected and classified are considered true positives (TPs); signs detected but falsely classified and undetected signs together are considered false negatives (FNs); background objects classified as signs are considered false positives (FPs).

From the result we have 91.47% precision and 92.19% recall for the system. Among the speed-limit signs that are detected and classified correctly, many of them are from images with low light conditions or blurry images (Fig. 15), but the system can handle them and reach high accuracy. The execution time of the system is 0.26s per input image.

Table 3: Result for the proposed system.

Number of signs detected and classified	118
Number of signs detected but falsely classified	4
Number of undetected Signs	6
Number of background classified as signs	11
Recall (%)	92.19
Precision (%)	91.47

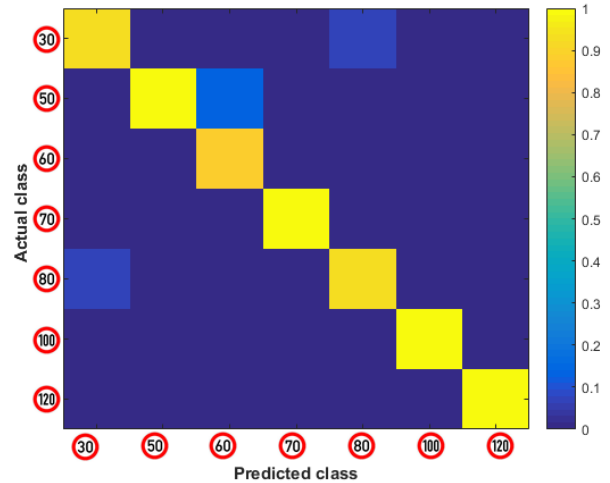


Fig. 14: Confusion matrix for the proposed system. (The “20” valued speed-limit sign class is ignored because it is not appear in the test set images.)

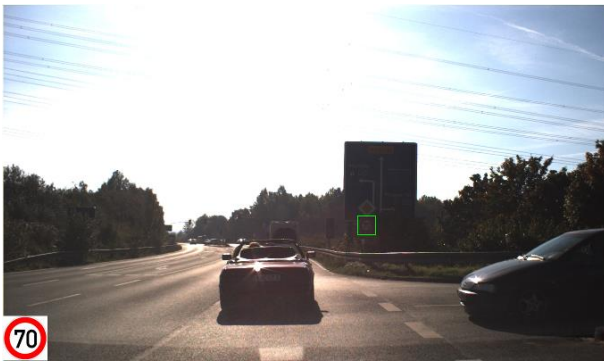


Fig. 15: The results of the proposed system executes on some hard cases, e.g., images that are blurry, in shadow, and under poor lighting conditions.

4. CONCLUSION

This paper proposes a system for the automatic detection and recognition of speed-limit signs. After preprocessing is applied to the input image, MSER detection is used to find potential speed-limit signs. Different values of speed-limit signs can be identified by using a multiclass SVM classifier. The proposed system can handle different lighting condition whether poor lighting scenes or blurry images in the GTSDB dataset. The system has the performance of 91.47% precision and 92.19% recall. Finally, the system runs in 0.26s per image. Future works include implementing the system using C++ and exploit parallelism during preprocessing to reduce execution time and improving the effectiveness of the detection by adapting better red color enhancement.

REFERENCES

- [1] R. Malik, J. Khurshid, and S. N. Ahmad, "Road sign detection and recognition using colour segmentation, shape analysis and template matching," *IEEE International Conference on Machine Learning and Cybernetics*, pp. 3556-3560, 2007.
- [2] W. Shadeed, D. Abu-Al-Nadi, and M. Mismar, "Road traffic sign detection in color images," *Proceedings of the IEEE International Conference on Electronics, Circuits and Systems*, Vol. 2, pp. 890-893, 2003.

- [3] G. Loy and N. Barnes, "Fast shape-based road sign detection for a driver assistance system," *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, Vol. 1, pp. 70-75, 2004.
- [4] N. Barnes and A. Zelinsky, "Real-time radial symmetry for speed sign detection," *IEEE Intelligent Vehicles Symposium*, pp. 566-571, 2004.
- [5] J. Greenhalgh and M. Mirmehdi, "Real-time detection and recognition of road traffic signs," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 13, 1498-1506, 2012.
- [6] S. Salti, A. Petrelli, F. Tombari, N. Fioraio, and L. Stefano, "Traffic sign detection via interest region extraction," *Pattern Recognition*, Vol. 48, pp. 1039-1049, 2015.
- [7] J. Torresen, J. W. Bakke, and L. Sekanina, "Efficient recognition of speed limit signs," *Proceedings of the IEEE International Conference on Intelligent Transportation Systems*, pp. 652-656, 2004.
- [8] Y. Y. Nguwi and A. Kouzani, "Detection and classification of road signs in natural environments," *Neural Computing and Applications*, Vol. 17, pp. 265-289, 2008.
- [9] S. Maldonado-Bascon, S. Lafuente-Arroyo, P. Gil-Jimenez, H. Gomez-Moreno, and F. Lopez-Ferreras, "Road-Sign Detection and Recognition Based on Support Vector Machines," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 8, pp. 264-278, 2007.
- [10] P. Sermanet and Y. LeCun, "Traffic sign recognition with multiscale convolutional networks," *The International Joint Conference on Neural Networks (IJCNN)*, pp. 2809-2813, 2011.
- [11] Y. Wu, Y. Liu, J. Li, H. Liu, and X. Hu, "Traffic sign detection based on convolutional neural networks," *The International Joint Conference on Neural Networks (IJCNN)*, pp. 1-7, 2013.
- [12] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image and Vision Computing*, Vol. 22, pp. 761-767, 2004.
- [13] A. Ruta, Y. Li, and X. Liu, "Real-time traffic sign recognition from video by class-specific discriminative features," *Pattern Recognition*, Vol. 43, pp. 416-430, 2010.
- [14] S. K. Berkaya, H. Gunduz, O. Ozsen, C. Akinlar, and S. Gunal, "On circular traffic sign detection and recognition," *Expert Systems with Applications*, Vol. 48, pp. 67-75, 2016.
- [15] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "The German Traffic Sign Recognition Benchmark: A multi-class classification competition," *The International Joint Conference on Neural Networks (IJCNN)*, pp. 1453-1460, 2011.
- [16] 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.