Bulletin of Networking, Computing, Systems, and Software – www.bncss.org, ISSN 2186–5140
Volume 4, Number 1, pages 15–17, January 2015

Efficient Road-Sign Detection Based on Machine Learning

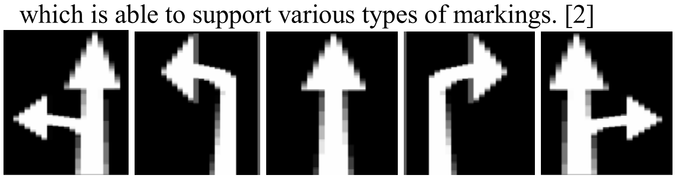
Dajun Ding, Jongsu Yoo, Jekyo Jung, Sungho Jin and Soon Kwon*
Daegu Gyeongbuk Institute of Science & Technology (DGIST), Daegu, Republic of Korea
{dajunding, soonyk}@dgist.ac.kr

***Abstract*— Road-signs which indicate the following traffic information can provide important assistance for driving safety in driving assistance system (DAS). In order to detect road-signs, we use a machine learning method in this paper. We first generate database using some image processing steps, then we use HOG for feature description. In the road marking identification step, we use SVM for this task. The designed method is applied using various video images from black box, and is verified to be robust and efficient.**

Keywords*— *Road-sign detection; HOG; SVM

I. INTRODUCTION

Road-sign refer to the signs drawn on the surface of the road. These differ from traffic signs erected by the side or on top of roads. Road-signs are as important to detect as traffic signs for navigation systems and driver assistive devices. [1] Road marking detection has been a popular research topic in the context of Autonomous Driver Assistance Systems (ADAS). Common road-signs include lane markings, arrows, zebra crossings, words, etc. Researchers aim to detect and locate these road-signs, and utilize the results to guide vehicle autonomous navigation. This paper proposes a general framework for road-sign detection and analysis using vision, which is able to support various types of markings. [2]



a. left & forward b. left c. forward d. right e. right & forward
Fig.1. Different road-signs type

In real road situation, arrow road-signs are among the most common markings. And since arrow markings are similar in different countries, the detection for such task is more important than other road-signs. Our aim is to develop a system that will reliably detect and classify the types of objects shown in Fig.1 with as little computational overhead as possible. [3] The rough shapes of different markings are shown in Fig.1.

Road-signs are painted in the middle of the road. [4] So the road-sign detection is close related with lane detection [5]. In a common road-sign detection technique, lane detection can provide important information for road region's setting and more precise result.

This structure of this paper is described as follows. In section II, we show our algorithm for road-signs' database extraction and description. We use some image processing steps such as Otsu threshold and contour filter for image normalization. The standard image is further described using HOG (Histogram of oriented gradient) algorithm [6]. The road-sign's detection and identification which based on lane detection result and SVM is shown in section III. At last, section IV gives concluding remarks. Our method was proved to be efficient and robust for different road conditions.

II. DATABASE GENERATION

Using a common machine learning strategy, our system consists of training and testing phases. According to our knowledge, there is no public database for road-signs in Korea. In that considering, we decided to make our own database for further processing. We normalized input images which have different orientation and perspective characteristics, the result image are standardized to get the same size and shape. We will call these training images as template images henceforth since these are used as templates for the road-sign detection.

The database generation part can be divided by two steps: normalization and description. In the first step, we use some image processing to get an image like Fig.2.b. In the description step, we use HOG [6] as well as SVM (support vector machine) [2] to make machine learning's database.

A. Database Normalization

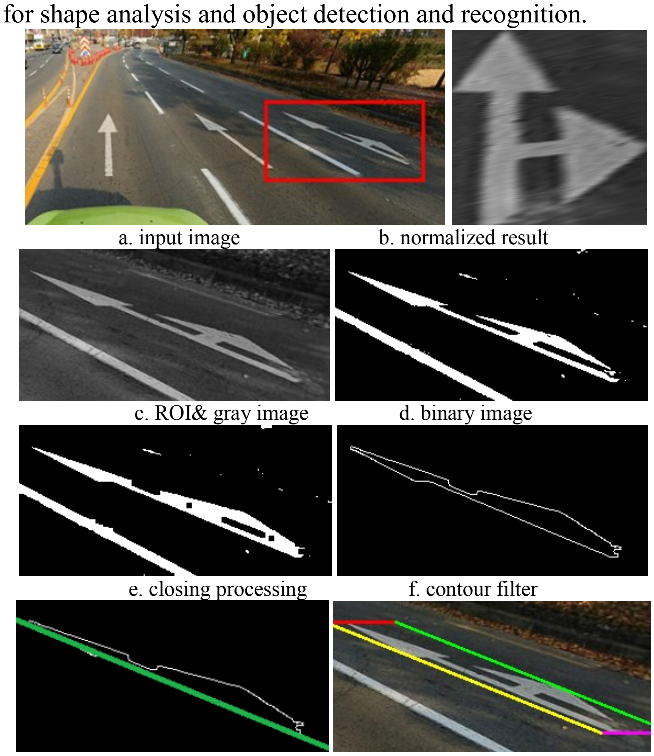
The input image is shown in Fig.2.a, and the road-sign we want to use is the right & forward one. Comparing with the other two forward markings, this one has a large slant direction and more noise as its special perspective view point.

At first, we cut the ROI and turn it into gray image. To make a clear database, other noise or markings shall be as few as possible. However, sometimes, it can't be avoided perfectly. As shown in Fig.2.c, a part of one lane marking is included in the ROI, we need to segment the road-sign and remove other objects.

In the second step, we applied the adaptive threshold on the image. We use the Otsu algorithm here. This method is used to automatically image threshold which is based on pixel clustering. It is assumed that the image only have two parts, background and foreground, it then calculates the optimum threshold separating the two classes so that their combined spread (intra-class variance) is minimal.

In real situation, road-signs might be broken and have lots crevices. In case of that, one road-sign can be consisted with several separated parts. We need to combine these parts and make the marking into one whole group. Then a morphology processing is utilized by using this binary broken image. We apply the dilation processing first and then erosion. Dilation can repair the broken crevices and erosion will help to keep the original image's size and shape. The output image in this step is shown in Fig.2.e.

As is shown in the binary image, there are lots noise and other objects. However, we only need the largest blob in this ROI. We use a blob extraction method for this step. A blob is a region of a digital image in which some properties are constant or vary within a prescribed range of values. The function retrieves the largest blob from the binary image using the contour filter algorithm. Contours are a useful tool for shape analysis and object detection and recognition.



a. input image b. normalized result

c. ROI& gray image d. binary image

e. closing processing f. contour filter

g. straight line detected by HT h. parallelogram ROI

Fig.2. Database normalization

According to the natural structure of the road-sign, we detect the largest straight line of the contour using Hough transform for straight line detection (shown in Fig.2. g). Then we utilize the straight line and slice it in the contour from left to right. The smallest parallelogram which can contain the whole region and with one side parallel to the bottom is what we need. It is shown in Fig. 2.h.

In the last step of the normalization, we use affine transformation to turn the image into a standard image. Affine transformation is a processing that preserves collinearity and ratios of distances. In this sense, affine indicates a special class of projective transformations that do not move any objects from the affine space to the plane at infinity or conversely. Formula (1) shows the affine transform. s is the changed scale, α is the rotation angle and x_0 y_0 are the offsets of x and y.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = s * \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} x - x_0 \\ y - y_0 \end{bmatrix} \tag{1}$$

After filling some black region using neighborhood's value, the final result is shown in Fig.2.b. It illustrates that the right & forward road-sign in this frame can be segmented can turned into a standard image. This image is used for further feature description and database generation.

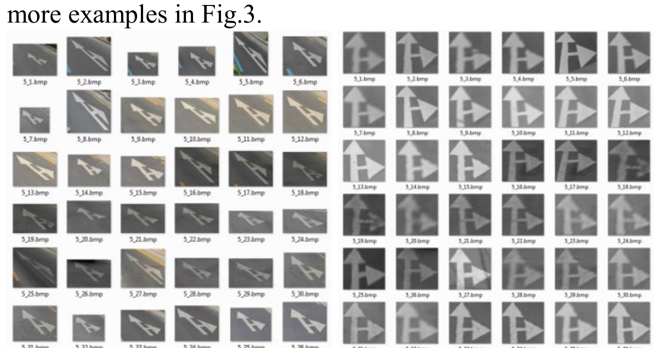
B. Database description

After database normalization, the input images for machine learning are set to be standard. And in the database description part we use HOG for image description and database generation. The HOG method is based on evaluating well-normalized local histograms of image gradient orientations in a dense grid. In practice this is implemented by dividing the image window into small spatial regions or cells, for each cell accumulating a local 1-D histogram of gradient directions or edge orientations over the pixels of the cell. The combined histogram entries form the representation.

TABLE1: DATABASE NUMBER

forward	left	right
1381	380	724
left & forward	**right & forward**	**total**
333	419	3237

After the database generation, the total number of each road-sign is shown in table1. In road condition, the forward is most common, so the number is the largest. And turning left and left & forward road-sign never exist in the highway road. So the number for this two markings is much smaller. The total number for road-signs detection is 3237. We can see more examples in Fig.3.



a. input image b. output database

Fig.3. Database example

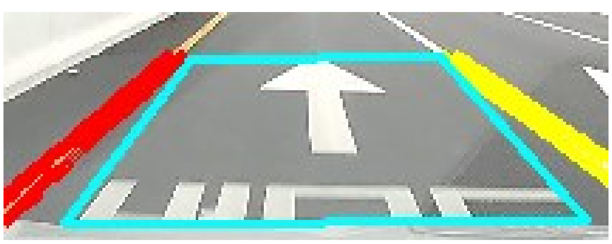
III. ROAD-SIGN IDENTIFICATION

Based on the database we extracted, in this part we focus on the road-sign identification. This step can be divided into two parts: detection and recognition.

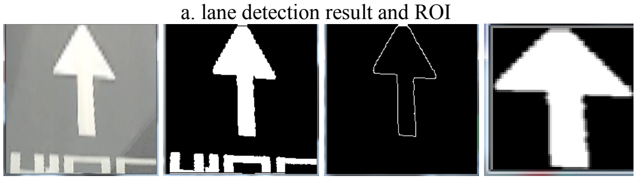
In the detection part, we use a similar strategy like database extraction. But there are also some differences. The ROI we set in this step is based on the lane detection result [5]. In our system, lane markings are detected using the IPM (inverse perspective mapping) as well as top-hat filter. The high response values are gathered using the points distribution histogram. And the lane detection result is shown in Fig.4.a. Red and yellow color indicate left and right lane. We set the ROI based on the lane boundary to make result more precise and efficient. The ROI which is drawn with blue line is illustrated in Fig.4.a. Then affine transformation is applied here to get Fig.4.b, Otsu threshold algorithm (Fig.4.c) and contour filter (Fig.4.d) are also processed. In the Hough straight line detection step, the slope angle of straight line we detected shall be more vertical than the value in section II, as the object we segmented has a parallel orientation with the lane markings. And the standard image is shown in Fig.4.e.

We use SVM for road-sign identification. In machine learning, support vector machines (SVM) is among the most widely used learning algorithms. SVM's idea is recognize patterns by analyzing data. Like other method, after lots training data is provided, every data with different labels will be separated into two groups. The training algorithm builds a model that assigns new examples into one category or the other, making it a non-probabilistic binary linear classifier.

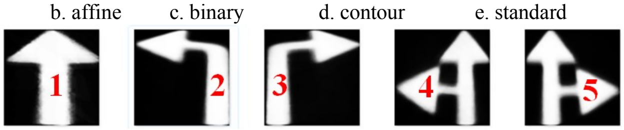
An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. We use 6 labels for road-sign's identification, and the model's labels are shown in Fig.4.f. The label 0 indicates the object is no-road-sign. Based on the SVM identification result, the detected road-signs are shown in the middle of the ROI. And the segmented object is shown in the up-right of the frame. The final result is displayed in Fig.4.g.



a. lane detection result and ROI



b. affine c. binary d. contour e. standard



f. database & label

g. final result

Fig.4. Road-sign identification

III. Conclusion

In this paper, we proposed a road-sign detection method based on HOG+SVM. We first generate road-signs database using image processing. The standard databases road-signs are further described using HOG and trained using SVM. Based on the lane marking detection result, we set the ROI and get the road-sign in the input image. As shown in our experiment results, the road-signs can be descripted correctly and efficiently.

Acknowledgment

This work was supported by the DGIST R&D Program of the Ministry of Science, ICT and Future Planning (14-IT-02, 14-HRLC-01).

References

[1] Wu, Tao, and Ananth Ranganathan. "A practical system for road marking detection and recognition." Intelligent Vehicles Symposium (IV), 2012 IEEE. IEEE, 2012.

[2] B. Qin et al., "A general framework for road marking detection and analysis," in Proc. 16th Int. IEEE ITSC, 2013

[3]Danescu, Radu, and Sergiu Nedevschi. "Detection and classification of painted road objects for intersection assistance applications." Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on. IEEE, 2010.

[4]Vacek, Stefan, Constantin Schimmel, and Rüdiger Dillmann. "Road-marking Analysis for Autonomous Vehicle Guidance." EMCR. 2007.

[5] Aly M. "Real time detection of lane markers in urban streets." Intelligent Vehicles Symposium(IV), IEEE, pp.7-12, 2008

[6] Dalal, Navneet, and Bill Triggs. "Histograms of oriented gradients for human detection." Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. Vol. 1. IEEE, 2005.