2018 IEEE Intelligent Vehicles Symposium (IV)
Changshu, Suzhou, China, June 26-30, 2018

Vision-based recognition of road regulation for Intelligent Vehicle

Kwangyong Lim, Yongwon Hong, Minsong Ki, Yeongwoo Choi, and Hyeran Byun

Abstract— **In this paper, we present a new framework to detect and recognize entire lanes and symbolic marks on high resolution road images. The first part of the framework utilizes local threshold to overcome the limitations of fixed threshold determination in road marking segmentation. The second part of the framework handles false detections caused by nearby objects on the roads such as vehicles and buildings by re-moving the areas that are not related to road surface using semantic segmentation. It also boosts recognition performance with a cascaded classifier structure that combines CNN for symbolic mark recognition and SVM for lane verification. The proposed lane detection achieves average F1-score of 0.96 and symbol recognition achieves average F1-score of 0.91. The proposed method is expected to advance the vehicle industry; with a GPU device, the proposed method can easily be embedded in smart vehicles.**

I. Introduction

Among the various components of ADAS technologies, one of the most fundamental parts is extracting information painted on the road surface. While the physical environments around a car can be captured by using radar and lidar sensors accurately, road surface is a flat plane with painted marks which cannot be captured by such sensors. Hence, vision-based approaches have been rigorously studied especially to detect lanes and symbolic marks. Lane detection technology already succeeds in ego-lane on some products, while symbolic mark recognition has not reached commercialization stage yet. Although symbolic marks are well regulated with simple line segments, they are not simple enough to be handled by simple methods due to variation in drawings, damages and illumination variations. Although recent success in deep learning offers powerful recognition performance, the limited processing capability of the Electronic Control Unit (ECU) inside the vehicle has been an obstacle. However, NVIDIA recently announced DRIVE PX [1], a GPU-optimized automotive environment that can process Full-HD images, to mount machine learning techniques on vehicles.

Traditional methods of lane and symbolic mark detection extract markings only on the ego-lane by applying a fixed threshold on brightness. Applying such a method on the entire image can lead to erroneous results due to occlusion by nearby vehicles and/or brightness variations. Recent studies on the lane and symbolic mark detection, diverse methods have been introduced. However, several problems still remain to be solved.

1) Targets are occluded by vehicles on the road.
2) Fixed thresholding method is not enough under inconsistent brightness.
3) Simultaneous detection of lanes and symbolic marks is not yet developed thoroughly.

In order to solve these problems, we detect and recognize the lanes and symbolic marks on the entire image by handling obstacles using a semantic segmentation method and applying a local thresholding for handling inconsistent brightness on the lanes and marks. The overview of the framework is illustrated in Figure 1 and described below.

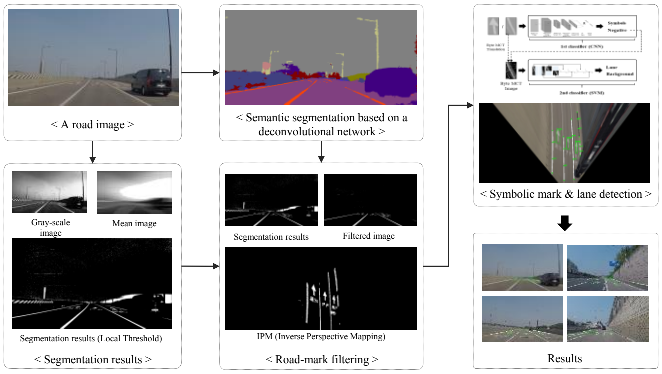


Figure 1. Block diagram of the proposed method

First, we generate a map of road-marks that will be used as candidates of lanes and symbolic marks by thresholding the image. The local threshold for each pixel is determined by a horizontal mean image (figure 1-c) and pixels brighter than neighboring region are marked. These procedures are well suited to be implemented with pixel-wise parallel processing on GPU.

Second, the map of the road-marks is filtered by pixel-wise labels from semantic segmentation of the original image to erase false positives due to non-road objects with similar appearances. A deconvolutional network is applied to label each pixel of the input image to one of the road, lane, vehicle, sky, pole, etc. and the marks are remained if the labels are road or lane.

Then, the filtered map of the road-marks and the original image are projected to top-down view to detect blobs without perspective distortion. The original image is cropped by detected blobs and recognized by a cascaded classifier of CNN and SVM.

Kwangyong Lim is with the department of computer science, Yonsei University, Seoul, Korea (e-mail: kylim@yonsei.ac.kr)
Yongwon Hong is with the department of computer science, Yonsei University, Seoul, Korea (e-mail: yhong@yonsei.ac.kr)
Minsong Ki is with the department of computer science, Yonsei University, Seoul, Korea (e-mail: kms2014@yonsei.ac.kr)
Yeongwoo Choi is with the department of computer science, Sookmyung Women's University, Seoul, Korea (e-mail: ywchoi@sookmyung.ac.kr)
Hyeran Byun is with the department of computer science, Yonsei University, Seoul, Korea (corresponding author; e-mail: hrbyun@yonsei.ac.kr)

978-1-5386-4451-5/18/$31.00 ©2018 IEEE

II. Related Works

Among wide range of ADAS technologies, this section reviews only the methods for lane detection and symbolic mark recognition.

A. Lane detection

Lane detection studies are divided into two areas: ego-lane detection and multi-lane detection. Ego-lane detection is mainly focused on Lane Departure Warning System (LDWS), and multi-lane detection is essential for autonomous vehicles and Mobile Mapping System (MMS).

Most of the multi-lane detection studies relied on the detection of ego-lanes, since it is difficult to detect a lane that is far from camera or a lane with low brightness. Thus, multi-lane detection methods are limited to ego-lane detection [2-7] or extend only to adjacent lanes [8,9]. But some of the methods find adjacent lanes based on the geometric position estimates to compensate for the limitations of the feature-based models. In order to reduce the errors of the ego-lane detection, Jiang et al.[10] utilize the location of the adjacent lanes using the width of the central lane. Most studies are based on lane fitting through RANSAC after edge detection using a fixed threshold. Thus the accuracies of the lane detection are getting worse when there is a vehicle running in the vicinity or is a lane with low brightness. To overcome heuristic ways of selecting the brightness thresholds, Foucher et al.[11] proposed a method to extract the road marking based on the 43rd Percentile Local Threshold (PLT) of the MinRGB [11] input image. The 43rd PLT is a median filtering method for selecting a value of the 43th pixel out of 100 pixels in the horizontal direction at an arbitrary position. This method uses a median value of the pixels in the horizontal direction at a certain pixel position as a local threshold to find the region of marks on the road surface. The marks have upper 10~30% brightness values, and that median value can be an effective local threshold for extracting marks. However, this method cannot be processed in a real-time due to many operations involved in selecting the median value.

Aly [12] introduced a method to detect multi-lanes using a selective oriented Gaussian filter and a random sample consensus (RANSAC) in top-down view converted an input image for all the lane detection on the road. This method eliminates the perspective effect by using Inverse Perspective Mapping (IPM) to obtain clear view of the lanes. The limitation of this method is that the computation time is longer since the vanishing point must be detected in every frames for accurate conversion in the top-down view generation where the perspective is removed. Recent studies have been introduced to detect lanes using vehicle detection based on Histograms of Oriented Gradients (HOG) [13].

B. Symbolic mark recognition

For symbolic mark detection, detection of markings inside the ego-lane has been proposed in [14,15]. In contrast, there are methods of detecting the symbolic mark first and detecting the surrounding lanes [16-20]. Yamamoto et al.[14] introduced a method of symbolic mark detection using boundary-tracking in the ego-lane based on the edges. The method extracts the horizontal and vertical projection features of the detected marking candidate region and recognizing those features with a multi-layer neural network is introduced. This method has difficulties in recognizing symbols with distortions and has limitations of only detecting inside the ego-lane.

Suhr et al.[15] introduced a HOG-based symbol recognition method. This method first detects the ego-lane using top-hat filtering and then detects positions of the symbolic marks through the horizontal axis projection in the ego-lane area. Kheyrollahi et al.[16] introduced a method of detecting a symbol region through an adaptive thresholding in a top-down view image and extracting a fuzzy zoning feature to feed into a neural network. Wu et al.[17] and Greenhjalgh et al.[18] introduced a method for detecting symbol regions through Maximally Stable External Regions (MSER), and these methods are basically same as the MSER-based region detection in the top-down view image. Greenhjalgh et al.[18] recognized symbols using HOG and Support Vector Machine (SVM), and Wu et al.[17] used a template matching for symbol recognition. However, these methods are difficult to recognize symbolic marks with distortions. W. Liu et al.[19] introduced the symbol recognition method using the HOG feature and the extreme learning machine (ELM) by detecting the symbol regions through an AdaBoost-based sliding-window search in the top-down view image. Unlike other methods, Chen et al.[20] introduced a method for detecting candidate regions using object proposals, which are mainly used for the object detection researches. This method detects the candidate region using an binarized normed gradient (BING) proposed by Cheng et al.[21] and recognizes the symbolic marks using PCANet[21] and SVM. However, this method has the limitation in elaborated localization, so that the position of the inaccurate symbol is extracted, which may lead to misclassification.

C. Semantic Segmentation

On a road scene, to extract lanes and symbolic marks, the only notable features are colors and brightness. For this reason, the conventional researches are mostly based on segmentation for extracting the lanes and symbolic marks. Since the pixel-wise segmentation method is focusing on each pixel's color and/or brightness without understanding an object, it is easily confused by an object that has similar colors or brightness such as white color vehicle, guardrail etc. On the other hand, the recent development using a deep learning algorithm allows the semantic segmentation of the road scenes as pixel-level classification. Kendal et al.[23] introduced an encoder-decoder architecture with a CNN based model, called SegNet, by adding a deconvolution after the convolution layers, which act as a decoder and an encoder respectively. The SegNet encodes an input image as features and decodes the features as pixel-level class. The SegNet also removes the fully-connected layer from the last layer of the conventional encoder, and decodes the features with unpooling and pooling indices from the encoder to restore resolution of the input image. Noh et al.[24] also suggests a similar encoder-decoder CNN based model called DeconvNet. The DeconvNet shares the basic architecture network with the SegNet: convolution for encoding and deconvolution for decoding, but the existence of a fully-connected layer at the last stage makes them different.

III. Road mark extraction

Road mark extraction is challenging due to the following reasons: 1) inconsistent light intensity, 2) interference with other objects such as nearby vehicles on the road. We introduce a binary segmentation with a local thresholding for extracting candidates of the road marks and also a method to effectively remove false positives on the roads.

A. Road-mark segmentation

Marks on the road provide important information for driving. They include symbolic marks and letters related to driving direction, speed limits and lanes. All the marks on the road surface are drawn only with three colors: white, blue, and yellow. But the road surface itself is mostly black and that is an asphalt color. The difference in brightness between marks and asphalt is designed to be large enough to be easily distinguished. There are many studies using such characteristics and the simplest method is to apply a binarization with a global threshold value. However, this simple thresholding cannot guarantee the reliable segmentation results with various illuminated road images. The inaccurate segmentation results also come from nearby vehicles with similar colors to the lanes. Figure 2 shows the segmentation results using a global threshold value. Scenes on the first and second rows are captured under the same lighting condition that results in correct binarization regardless of small changes in thresholding (100~130). But, in the third row, the brightness of the lane is too low that a threshold value of 120 or higher is not adequate. In the fourth row, the overall intensity of the scene is high that a threshold values with less than 120 does not reveal any marks.

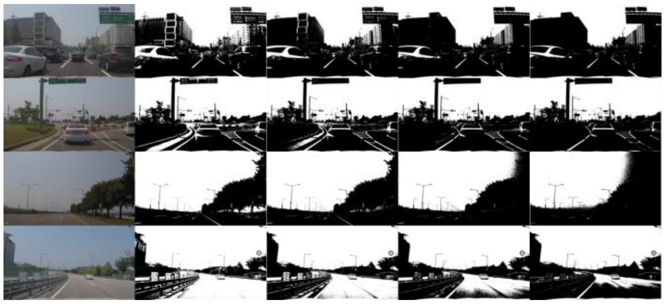


Figure 2. Binary segmentation results with various threshold values. (a) shows road images, and (b)~(e) show the results of binary segmentation with threshold values of 100, 110, 120 and 130.

Therefore, we suggest a mark segmentation method using local threshold values for each pixel considering brightness of the surrounding regions. Our method computes a mean brightness of the horizontal neighboring pixels for each pixel in the image and uses it as a threshold for each pixel. Equation 1 shows how to compute the mean image.

$$I_{mean} = \sum_{(u,v)\in N(x,y)} \frac{I(u,v)}{|N|} \tag{1}$$

where $I(\mathrm{u,v})$ denotes a pixel intensity at pixel (u,v), $N(x,y)$ is a set of horizontal neighbor pixels of (x, y). In our approach, we set the size N as 401 pixels so that it covers 200 pixels in each left and right directions. To get the mean image fast, it is computed in parallel on Graphics Processing Unit (GPU) so that it can be obtained in 3ms on the average for a Full-HD resolution image. The binarization with the mean image is performed using equation 2.

$$I_{bin} = C_1(x,y) \times C_2(x,y) \tag{2}$$

$$C_1(x,y) = \begin{cases} 1, & if\ I_{mean}(x,y) < I(x,y) \\ 0, & else \end{cases} \tag{3}$$

$$C_2(x,y) = \begin{cases} 1, & if\ \theta < I(x,y) \\ 0, & else \end{cases} \tag{4}$$

In equation 2, I_{bin} indicates the binarization results, and C_1, C_2 are indicator function for the mean brightness and minimum brightness threshold value respectively. C_2 suppresses responses from a relatively bright but still dark regions such as cracked asphalts. Figure 3 shows the mean images and their binarization results. The proposed method can effectively segment the road mark areas than the existing methods with a fixed threshold value, since the local thresholding method adaptively finds regions brighter than the horizontal neighbors. But, the objects with bright colors are still incorrectly segmented as road marks. To handle this problem, we propose a filtering method based on deep learning.

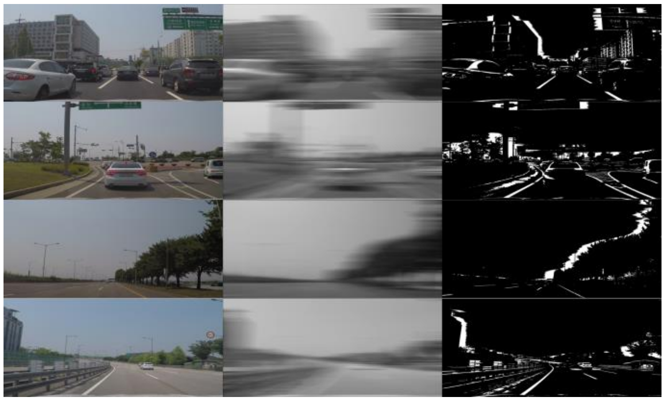


Figure 3. Examples of binary segmentation results based on local thresholding: (a) Road images, (b) Their mean images, and (c) Segmentation results

B. Road-mark map filtering based on deconvolutional network

While the proposed local-thresholding method reliably extract regions that are brighter than their neighbors, such regions usually include unrelated white objects such as cars, curbs and guard-rails. Naive thresholding depends on brightness which is not distinctive enough to distinguish lanes and symbolic marks from irrelevant white objects. Semantic segmentation methods are recently being applied to road environments to assign each image pixel to one of the labels such as cars, trees, road, etc. [25,26] Although they are not able to segment small objects like lanes, they can reasonably segment cars, trees and buildings. Thus, we apply a semantic segmentation method to filtering unrelated objects. The incorrect candidates can be removed if they are not labeled as roads by SegNet [23] trained on CamVid [25] and Cityspaces [26] datasets. Figure 4 shows the effectiveness of the proposed method. Figure 4(b) shows SegNet results and the results also include many incorrect segmentation labels such as guardrails and pavements that are very similar visual features of the lane on the road images. This is also partly due to the nature of the convolutional networks which lack invariance to affine transformation. Figure 4(d) shows the final results that do not include marks of buildings or cars.

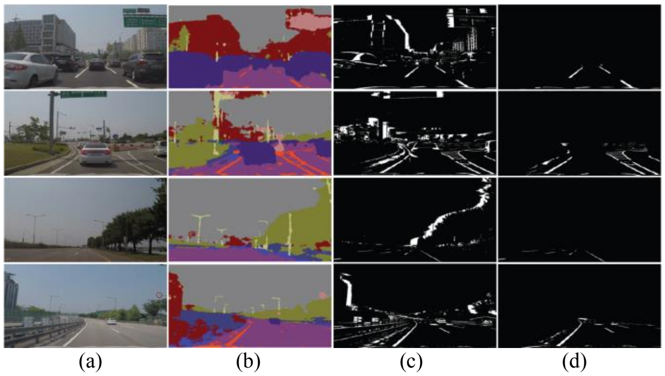


Figure 4. Examples of the filtering results with a deconvolutional network: (a) Road images, (b) SegNet results, (c) Candidate masks, and (d) Filtering results

IV. Lane and Symbolic mark recognition

Blobs – sets of 8-neighboring adjacent pixels – are extracted from the final filtered mask and the corresponding input images are fed into CNN and SVM to be classified into lanes, symbolic marks, and backgrounds. The structure of the proposed classifier is depicted in Figure 5.

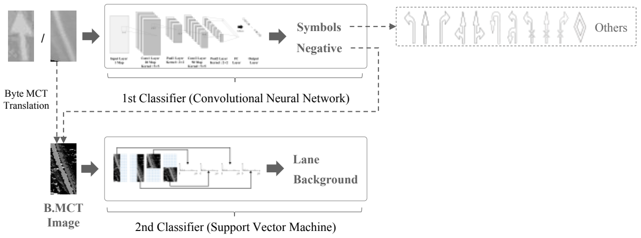

/
Symbols
Negative
Others
Byte MCT Translation
1st Classifier (Convolutional Neural Network)
B.MCT Image
Lane
Background
2nd Classifier (Support Vector Machine)

Figure 5. Structure of the proposed classifier for symbolic marks and lane recognition

The first classifier recognizes 14 symbols through two sets of convolution layers and a fully-connected layer. The second classifier based on SVM classifies Byte-MCT feature of the blobs that are labeled negatively by the first classifier. The details are explained in the later sections.

The images captured on the road inevitably contain perspective distortion of the road. The further an object is, the smaller it looks. Thus the perspective distortion can lower recognition accuracies. Hence, we adopt Inverse Perspective Mapping (IPM) to remove such distortion in the input image and to generate top-down view. We follow Bertozz et al.[27] to back-project a distorted image to a rotated view regarding the vanishing point. Due to an intensive computational amount of the back-projecting a high resolution image, we implemented IPM on GPU by computing matrices in parallel.

A. *Symbolic mark recognition*

This section explains a procedure for extracting blobs from a road mark map for generating candidate region of interests (RoIs) and recognizing the symbolic marks among the candidates. Blob labeling is the most common technique for extracting blobs from a binary image. It travels 8-neighbors of each white pixel and indexes pixels as a group.

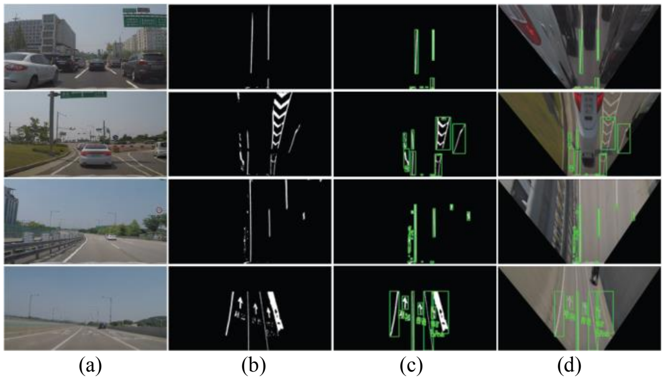


Figure 6. Results of blob detection: (a) Road images, (b) IPM results, (c) Detected blobs, and (d) Final results

Figure 6 shows the results of intermediate steps of the candidate detection. IPM is applied to the map of the road mark to remove distortions as in figure 6(b) and blobs are shown as green rectangles as in figure 6(c) and 6(d). The extracted blobs go through the classification and are labeled as one of the symbolic marks, lane, or negative class. While contemporary deep CNNs are prevalent in visual recognition of objects, the symbolic marks painted on the road are simple enough to be recognized by an overall shape rather than by feature extraction and recognition. Hence we first filter the blobs by aspect ratio and proportion of white pixels of the bounding boxes and recognize the symbolic marks by CNN.

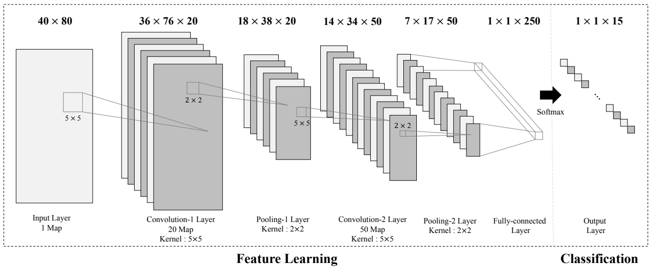


Figure 7. The proposed model architecture for symbolic mark recognition

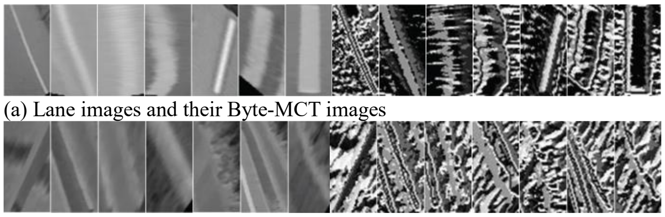
Figure 7 shows the proposed cascaded CNN architecture for the symbolic mark recognition. It consists of two pairs of convolution-pooling layers and the layers are sufficient to recognize small and simply shaped objects. The size of the input layer is chosen with respect to the aspect ratio of symbolic marks, instead of using a fixed size of the aspect ratio of square that is commonly used for object recognition. The last layer is feed into 15-way softmax classifier for 14 symbol classes and one negative class. The negative class covers the lanes and the backgrounds. It provides to suppress false-alarms of the symbolic marks that are similar in shape such as straight arrow.

B. Lane detection

The negative class is further classified into the lanes and the backgrounds. The blob images are blurry due to interpolation in IPM and they are easy to be confused with non-lane objects especially nearby borders such as curb, wall, etc. The first seven images of Figure 8 show candidate ROIs of the lanes and the backgrounds nearby border which are distinguishable. Such a severe blurring causes an edge-based method to fail and results the visual features unreliable. We solve the problem by using Byte-MCT [28] as feature extraction. Byte-MCT features have the following advantages:

1) Local edge structure can be extracted even on blurry images.
2) Features are robust to illumination changes.

The images in the second row of Figure 8 shows Byte-MCT results of the background. Note that the ROIs containing lanes have clear outlines, but the background ROIs is not that clear.



(a) Lane images and their Byte-MCT images

(b) Background images and their Byte-MCT images

Figure 8. Examples of the lane and the background images with Byte-MCT images

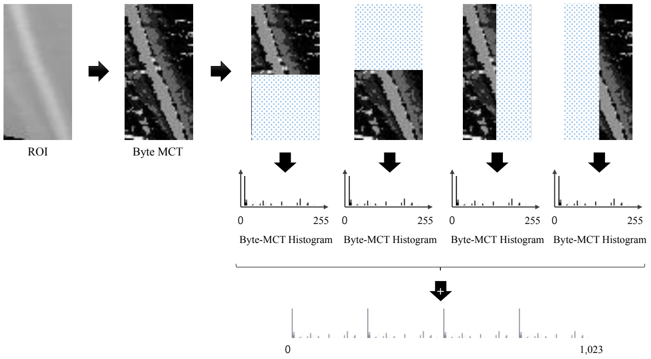


Figure 9. Example of feature extraction for lane verification

Figure 9 shows an example of the feature extraction for lane verification. For each candidate region, each pixel value represents a local edge structure using a 3-by-3 Byte-MCT. First, the ROI is divided into four overlapping regions as shown in Figure 9. Byte-MCT histogram of each region is a 1,024-dimensional feature vector. Equation 5 shows how to extract features;

$$H = \sum_{r=1}^{4} \frac{H_r}{\max(H_r)} \qquad (5)$$

where H is the final feature descriptor and r is the number of candidate regions. H_r is the histogram representation of the divided region r and $\max(H_r)$ is the maximum histogram value from H_r. This equation normalizes each divided region using its maximum value to get the 256-dimensional histograms. By concatenating four 256-dimensional histograms, a 1024-dimensional feature vector is generated. Since the features are generated by the Byte-MCT, they are also robust to the illumination changes. Finally, the SVM calculates the distance for each feature vector using the Radial Basis Function (RBF) kernel to decide the input classes: lane or background. In order to finalize the verification of a candidate region, two filtering steps are performed. The first filtering step removes zig-zag lanes based on the angles between separate line pieces. For the verification of the zig-zag lanes, we generate five line pieces by choosing six vertically uniform-distributed marking points between start and end points. The marking points are average x coordinates of horizontally searched white points on lines of *y=0.05, 0.2, 0.4, 0.6, 0.8, 0.95*. Steps of the selection of the marking points are shown below:

```
Input:
  $I$ - lane candidate image
  $Y$ - a set of vertical reference point
Output:
  $set_{point}$ - a set of marking label point
for every reference point $y$ in $Y$
  while $avg_x > 0$ and $y <$ height of image $I$
    $avg_x$ := calculate an average $x$-axis point of marking label at reference point $y$
    $set_{point}$ := add $point(avg_x, y)$
return $set_{point}$
```

If the standard deviation of angles from four pairs of marking points is greater than 2 radians, we remove the lines. It is also applicable to the curved lines since the curved lines are nearly straight on top-down view image.

The second filtering step removes short lanes. For the verification of the short lanes, we build histogram of x coordinates for white pixels and remove blobs with the histogram value less than a certain threshold.

V. EXPERIMENTS

A. Datasets

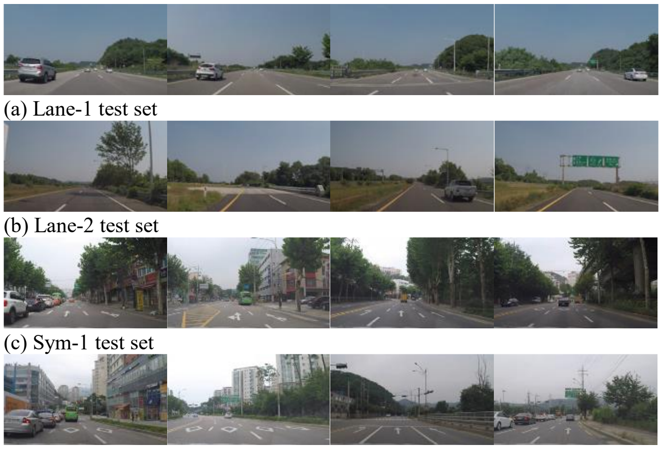
To evaluate the proposed method, we use a real-world driving data set that records driving scenes using a GoPro action camera mounted on the windshield of a car in the forward direction. The GoPro supports a wide angle view like a black box camera, while it has a less distortion than the black box camera. The test set consists of four different sets as shown in Table I. Lane-1 and Lane-2 test sets are captured on the highway to focus on the lane detection while Sym-1 and Sym-2 sets are captured in urban and suburban areas to focus on symbolic marks. Lane-1 test set mostly contains four-lane roads and lane-2 mostly contains two-lane roads. While most of the lane detection researches model the lanes into the lines, we individually detect lane segments since it helps a mobile mapping system. Each lane segments are manually labeled with top and bottom coordinates differently from the public data. Symbolic marks are also manually labeled with four corner coordinates and their types. All the images in the data set have Full-HD (1920×1080) resolution, and contain lane or symbolic marks. Figure 10 shows examples of the four test sets. Figure 10(a) shows the highway driving scenes in the morning, and 10(b) shows the highway driving scene in the afternoon. Figure 10(c) and (d) show the driving scenes in city area (Seoul) at daytime. The training sets consist of ROI images of lands and symbolic marks. They are captured with windshield mounted GoPro camera. The data sets include various images with different illumination conditions and the cropped symbolic marks and lanes are warped to be top-down view. Table II shows the specifications of the two training sets used in the experiments.

Table I.
Specifications of the four test sets

No	Name	Type	Frames	ROIs
1	Lane-1	Lane	2,288	13,161
2	Lane-2	Lane	2,056	7,634
3	Sym-1	Symbol	3,700	4,740
4	Sym-2	Symbol	3,700	4,723

Table II.
Specification of training sets

No	Name	# of classes	Images	Resolution
1	T-Sym	15	28,800	40×80
2	T-Lane	2	8,969	40×80


(a) Lane-1 test set
(b) Lane-2 test set
(c) Sym-1 test set
(d) Sym-2 test set
Figure 10. Examples of our real-world driving dataset

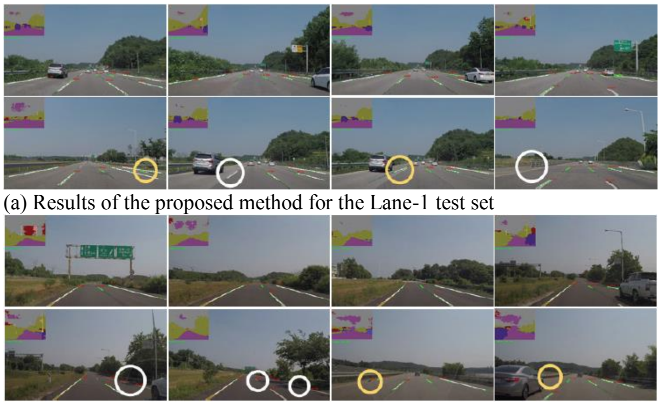
B. Lane detection performance

We evaluate the proposed lane detection method with Lane-1 and Lane-2 test sets and compare the results with the state-of-the-art method [12]. The proposed method achieves the highest accuracy with average F1-score of 0.96 in two test sets. Effectiveness of each proposed component is measured by ablation study as shown in Table III. Figure 11 shows the results of the proposed method on Lane-1 and Lane-2 test sets. Note that the proposed lane detection method achieves a stable performance even in the wide 4-lane roads and even with the nearby vehicles. Although some harsh environments with damaged paints in the lanes may defect our method, they mostly do not affect consequent frames and can be handled by averaging previous and following frames.

Results of the proposed method without SegNet-based filtering achieve an average F1-score of 0.87. This is due to the influence of nearby cars and structures and it explains the effectiveness of SegNet-based filtering. For the SegNet-only method, many lanes are not properly detected since the method has limitations on handling scale-variance. Finally, we compare our method with the state-of-the-art lane detection method [12], and the method achieves an average F1-score of 0.53 that is much lower than our proposed method. Since [12] relies on the edge detection, the images taken from the lower angles highly suffers from perspective distortions.

Table III.
Results of Lane detection performance

Method	Dataset	TP	FP	FN	Precision	Recall	F1
Proposed	Lane-1	12,581	328	478	0.97459	0.96339	0.96896
	Lane-2	7,351	543	241	0.93133	0.96825	0.94943
Proposed without filtering	Lane-1	11,793	1,629	1,353	0.87863	0.89797	0.88775
	Lane-2	6,319	929	1,272	0.87182	0.83243	0.85167
SegNet Only	Lane-1	6,783	2,026	6,357	0.77000	0.51621	0.61806
	Lane-2	4,943	1,724	2,682	0.74141	0.64826	0.69171
Caltech [12]	Lane-1	5,961	3,819	7,187	0.60950	0.45337	0.51997
	Lane-2	4,052	6,801	10,762	0.57605	0.53127	0.55275


(a) Results of the proposed method for the Lane-1 test set
(b) Results of the proposed method for the Lane-2 test set
Figure 11. Result for Lane-1 and Lane-2 test set. First row shows correct detection results and second row shows incorrect detection results. White circle indicate false negatives and yellow circles indicate false positives.

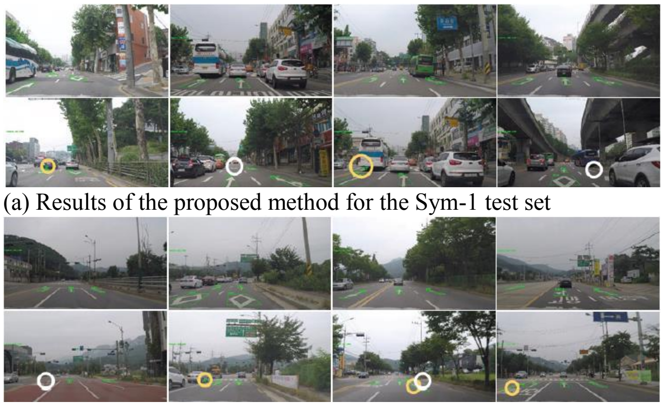
C. Symbolic mark recognition performance

We test our symbolic mark recognition with Sym-1 and Sym-2 test sets. Table IV shows the results comparing the proposed method and R-CNN[29]. Ablation study is given including (1) without SegNet-based filtering, (2) SegNet-based end-to-end mark segmentatation, and (3) naive R-CNN. The proposed method achieves the highest symbolic mark recognition accuracy with average F1-score of 0.92. Figure 12 shows the results of proposed method on Sym-1 and Sym-2 test sets suggesting that our method is robust to the nearby vehicles on wide 4-lane roads. While some of symbolic marks nearby borders and lanes are misclassified as symbolic marks, they mostly do not affect consequent frames and can be classified by averaging previous and following frames.

SegNet-based filtering procedure is not as critical as lane detection resulting in average F1-Score of 0.91. Symbolic marks suffer less from nearby vehicles because they are not entirely occluded at all unlike lanes. For the SegNet-only method, many symbols are not properly detected by SegNet since it lacks robustness against scale-variance. The state-of-the-art object detection method R-CNN[29] achieves the lowest average F1-score of 0.48. R-CNN starts from EdgeBox[30], an object proposal, and it highly depends on the edge detection results. Since it suffers from severe perspective distortion captured from windshield-mounted camera, the symbolic marks on the ego-lane are only detected. The EdgeBox detect objects candidate regions through the measuring an objectness. However, the distorted symbolic mark has difficulty in detecting because it has low-objectness compared to other objects such as vehicles. Although it shows moderate performance when used on top-down view images, it still fails on far away symbolic marks since it still have unclear edge responses. Also, it tends to mislocalized on cluttered area for classifying the symbolic marks.

Table IV.
Result of Symbolic mark recognition performance

Method	Dataset	TP	FP	FN	Precision	Recall	F1
Proposed	Sym-1	4,483	699	209	0.86511	0.95546	0.90804
	Sym-2	4,615	543	83	0.89473	0.98233	0.93649
Proposed without filtering	Sym-1	4,473	758	217	0.85509	0.95373	0.90172
	Sym-2	4,612	600	82	0.88488	0.98253	0.93115
SegNet Only	Sym-1	2,635	1,362	1,875	0.65924	0.58426	0.61949
	Sym-2	2,661	1,159	1,930	0.69660	0.57961	0.63274
R-CNN [29]	Sym-1	1,253	625	2,997	0.66720	0.29482	0.40894
	Sym-2	1,909	636	2,459	0.75010	0.43704	0.55229



(a) Results of the proposed method for the Sym-1 test set

(b) Results of the proposed method for the Sym-2 test set

Figure 12. Result of proposed method on Sym-1 and Sym-2 test set. First row shows correct detection results and second row shows incorrect detection results. White circle indicate false negatives and yellow circles indicate false positives.

VI. Conclusion

In this paper, we present a carefully designed framework of methods to detect and recognize entire lanes and symbols on high resolution road images. The first part of the framework overcomes the problem of fixed threshold determination in road marking segmentation using local threshold. The second part of the framework handles false detections caused by nearby objects e.g., vehicles and buildings by removing the areas that are not related to road surface using semantic segmentation. It also boosts recognition performance with a cascaded classifier structure that combines CNN for symbolic mark recognition and SVM for lane verification. Four primary contributions can be listed as follows:

1) Constructing a dataset for recognition of entire lanes and symbols on wide angle view

2) Removing false detections caused by vehicles driving around lanes and symbols using semantic segmentation

3) Simultaneous recognition of all lanes and symbols

4) Cascaded classifier using CNN and SVM

The proposed lane detection achieves average F1-score of 0.96 and symbol recognition achieve average F1-score of 0.91. With a GPU device, DRIVE PX [1], for smart vehicles, the proposed method can be implemented in the vehicle industry in the near future.

Acknowledgment

This work was partly supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIP) (No. 2016-0-00152, Development of Smart Car Vision Techniques based on Deep Learning for Pedestrian Safety) and MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2018-2016-0-00464) supervised by the IITP(Institute for Information & communications Technology Promotion).

References

[1] NVIDIA DRIVE PX, http://www.nvidia.com/object/drive-px.html, (Accessed December 19, 2017)

[2] A. Lopez, and J. Saludes, "Detection of Lane Markings based on Ridgeness and RANSAC," IEEE Conference on Intelligent Transportation Systems, pp. 254-259, 2005.

[3] A. Criminisi, "A plane measuring device," Image and Vision Computing, Vol. 17, No. 8, pp. 625-634, 1999.

[4] S. S. Ieng, J. P. Tarel, and R. Labarade, "On the design of a single lane-markings detectors regardless the on-board camera's position," IEEE International Symposium on Intelligent Vehicles, pp. 564-569, 2003.

[5] W. Een, and N. Suvanvorn, "A Study of the Edge Detection for Road Lane," IEEE International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, pp. 995-998, 2003.

[6] S. Zhou, and H. Chen, "A Novel Lane Detection based on Geometrical Model and Gabor," IEEE International Symposium on Intelligent Vehicles, pp. 59-64, 2010.

[7] G. Kucukyildiz, and H. Ocak, "Development and optimization of dsp based real time lane detection algorithm on a mobile robot platform," IEEE International Conference on Signal Processing and Communications Applications Conference, pp. 1-4, 2012.

[8] H. Deusch, and J. Wiest, "A random finite set approach to multiple lane detection," IEEE International Conference on Intelligent Transportation Systems, pp. 270-275, 2012.

[9] J. Hur, S. Kang, and S. Seo, "Multi-lane detection in urban driving environments using conditional random fields," IEEE International Symposium on Intelligent Vehicles, pp. 1297-1302, 2013.

[10] Y. Jiang, and F. Gao, "Computer vision-based multiple-lane detection on straight road and in a curve," IEEE International Conference on Image Analysis and Signal Processing, pp. 114-117, 2010.

[11] P. Foucher, Y. Sebsadji, J. P. Tarel, P. Charbonnier, and P. Nicolle, "Detection and Recognition of Urban Road Markings Using Images," IEEE International Conference on Intelligent Transportation Systems, pp. 1747-1752, 2011.

[12] M. Aly, “Real time Detection of Lane Markers in Urban Streets,” IEEE International Symposium on Intelligent Vehicle, pp. 7-12, 2008.

[13] R. K. Satzoda, and M. M. Trivedi, “Efficient lane and vehicle detection with integrated synergies (ELVIS),” IEEE International Conference on Computer Vision and Pattern Recognition Workshops, pp. 708-713, 2014.

[14] J. Yamamoto, S. Karungaru, and K. Terada, “Road Surface Marking Recognition using Neural Network,” IEEE International Symposium on System Integration, pp. 484–489, 2014.

[15] J. K. Suhr, and H. G. Jung, “Fast Symbolic Road Marking and Stop-line Detection for Vehicle Localization,” IEEE International Symposium on Intelligent Vehicles, pp. 186-191, 2015.

[16] A. Kheyrollahi, and T. P. Breckon, “Automatic real-time road marking recognition using a feature driven approach,” Machine Vision and Applications, pp. 1-11, 2010.

[17] T. Wu, and A. Ranganathan, “A Practical System for Road Marking Detection and Recognition,” IEEE International Symposium on Intelligent Vehicles, pp. 25-30, 2012.

[18] J. Greenhalgh, and M. Mirmehdi, “Detection and Recognition of Painted Road Surface Markings,” International Conference on Pattern Recognition Applications and Methods, pp. 130-138, 2015.

[19] W. Liu, J. Lv, B. Yu, W. Shang, and H. Yuan, “Multi-type road marking recognition using AdaBoost detection and extreme learning machine classification,” IEEE International Symposium on Intelligent Vehicles, pp. 41-46, 2015.

[20] T. Chen, Z. Chen, Q. Shi, and X. Huang, “Road Marking Detection and Classification Using Machine Learning Algorithms,” IEEE International Symposium on Intelligent Vehicles, pp. 617-621, 2015.

[21] M. M. Cheng, Z. Zhang, W. Y. Lin, and P. Torr, “BING: Binarized normed gradients for objectness estimation at 300fps,” IEEE International Conference on Computer Vision and Pattern Recognition, pp. 3286-3293, 2014.

[22] T. Chen, K. Jia, S. Gao, J. Lu, Z. Zeng and Y. Ma, “PCANet: A Simple Deep Learning Baseline for Image Classification?,” IEEE Transactions on Image Processing, Vol. 23, No. 12, pp. 5017-5032, 2015.

[23] V. Badrinarayanan, A. Kendall, and R. Cipolla, “SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation,” IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 39, No. 12, pp.2481-2495, 2017.

[24] H. Noh, S. Hong, and B. Han, “Learning Deconvolution Network for Semantic Segmentation,” IEEE International Conference on Computer Vision, pp. 1520-1528, 2015.

[25] Cambridge Video Dataset, http://mi.eng.cam.ac.uk/research/projects/VideoRec/CamVid,(Accessed December 19, 2017)

[26] Cityscapes Dataset, https://www.cityscapes-dataset.com, (Accessed December 19, 2017)

[27] M. Bertozz, A. Broggi, and A. Fascioli, “Stereo inverse perspective mapping: theory and applications,” Image and Vision Computing, Vol. 16, No. 8, pp. 585-590, 1998.

[28] K. Lim, Y. Hong, Y. Choi, and H. Byun, "Real-time traffic sign recognition based on a general purpose GPU and deep-learning," PLoS one, 12(3), e0173317, 2017.

[29] R. Girshick, J. Donahue, T. Darrell and J. Malik, “Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation,” IEEE International Conference on Computer Vision and Pattern Recognition, pp. 580-587, 2014.

[30] C. Zitnick, and P. Dollár, “Edge Boxes: Locating Object Proposals from Edges,” European Conference on Computer Vision, pp. 391-405, 2014.