Traffic Lane Detection using Fully Convolutional Neural Network

Jinju Zang, Wei Zhou✉,Guanwen Zhang, Zhemin Duan
Northwestern Polytechnical University, Xi'an, China
E-mail: zhouwei@nwpu.edu.cn

***Abstract*—Numerous groups have conducted many studies on traffic lane detection. However, most methods detect lane regions by color feature or shape models designed by human. In this paper, a traffic lane detection method using fully convolutional neural network is proposed. To extract the suitable lane feature, a small neural network is built to implement feature extraction from large amount of images. The parameters of lane classification network model are utilized to initialize layers' parameters in lane detection network. In particular, a detection loss function is proposed to train the fully convolutional lane detection network whose output is pixel-wise detection of lane categories and location. The designed detection loss function consists of lane classification loss and regression loss. With detected lane pixels, lane marking can be easily realized by random sample consensus rather than complex post-processing. Experimental results show that the classification accuracy of the classification network model for each category is larger than 97.5%. And detection accuracy of the model trained by proposed detection loss function can reach 82.24% in 29 different road scenes.**

I. Introduction

Advanced Driver Assistant System (ADAS) is based on the comprehension and perception of traffic environment surrounding the car. The traffic environment of cars includes other cars, lane line, pedestrian and so on. Traffic lane detection plays an important role in ADAS. Lane detection can be deployed to build a lane-departure warning system and control the direction of self-driving cars. Traditional lane detection methods normally performed complex image processing to obtain the lane feature, which are previously designed by human. Recent years, deep learning, or neural networks, has shown strong power in image processing. Unlike traditional image processing methods, neural networks are trained by a large number of images to extract proper feature. Applying the deep learning technology in lane detection is the focus of this paper.

Previous lane detection and tracking works are mostly based on designed hypothetical lane shape models or lane color features. For the hypothetical shape model, the parameters of designed model have strong influence on the detection performance. In Hyperbola pair [1], calculation of lane curvature determines whether the lane pixels can be searched. With B-snake model [2], the extracted control pixels are deployed to match parametric model. Lane shape models also include piecewise line segments [3] and adoptive weak lane model [4]. As well in the bird's-eye view, the lane shape is straight lines. To obtain the bird's-eye views [5][6][7][8], camera's parameters are necessary for establishing an inverse perspective transformation matrix. In [9], the lane is modeled as double lines and it is supposed that the camera be in the center of cars. Unfortunately, the shape model based lane detection normally has low flexibility. Most lane shape models normally have good performance on lanes with the specific shape. In color feature based lane detection, the most obvious feature at lane boundaries is local gradient change [10], which is enhanced in gray images obtained by linear discrimination [11]. In [12], illumination invariant lane marker candidate detection by the distinct lane color feature is proposed. Some researchers also exploit the spatial information with mask [13][14] to realize lane detection. However, feature based lane detection method always needs complicated image preprocessing which increases the complexity of lane detection. Furthermore, specific feature based lane detection usually has good performance in particular scene [15]. Typically random sample consensus (RANSAC) [16] is coupled with different filters or detectors, like Kalman filters [5], Gaussian sum particle filter(GSPF) [17] and Geometric overture for lane detection by intersections entirety(GOLDIE) [18], to mark the lane.

Some excellent detection algorithms have been proposed in the field of deep learning. R-CNN [19] adopts Selective Search (SS) [20] to search proposal regions on original pictures with large computing. Faster R-CNN [21] improves R-CNN through region proposal networks. Furthermore, SSD [22] and YOLO [23] directly divide feature maps of each convolution layer, eliminating the proposal region extraction. In [24], lane regions are divided into many small frames. And detection outputs are coordinates and depth of each small frame of lane regions. Reference [24] only detects location of lanes without outputting lane classification. Furthermore, Region FCN [25] constructs a fully convolutional network without fully connected layers and evaluates candidate regions with score maps to detect objects. Although the performance of above deep learning detection methods is outstanding, the scale of networks used in them are very big and deep. Unfortunately, the large network always requires more storage and computation devices.

In this paper, we propose a traffic lane detection method using fully convolutional neural network. The proposed approach mainly consists of two parts, lane classification and lane detection. In lane classification, lane feature is extracted by a small neural network instead of the feature designed by human. For lane detection, a detection loss function is designed to calculate detection loss on the whole output feature maps in training detection network. The detection loss consists of lane classification loss and lane regression loss. Specially, there is no region proposal and feature maps division in detection network. In addition, detection network is built by converting from lane classification network into a fully convolutional network. Therefore, detection network is small and occupies less computing source. Unlike previous deep learning detection methods, the output of proposed detection network is pixel-wise detection instead of rectangular frames covering segments of lanes. The detection network is trained end-to-end: from raw input images to detection maps which contain the locations and categories of lane region pixels. The lane marking with RANSAC based on detected lane pixels by detection network model is also discussed in this paper.

The main contribution of our work is twofold. First, a small neural network is constructed to extract suitable lane feature. Second, we propose a detection loss function which is made up of classification loss and regression loss. The obtained detection network model can provide pixel-wise detection of lanes categories and location. The advantage of pixel-wise detection is avoiding the shape limitation of lanes. The remained structure of this paper is arranged as follows. Section II shows the details of lane classification and lane detection of proposed approach. The algorithm for lane marking is also displayed in Section II. Section III introduces experiments results. This paper concludes in Section IV.

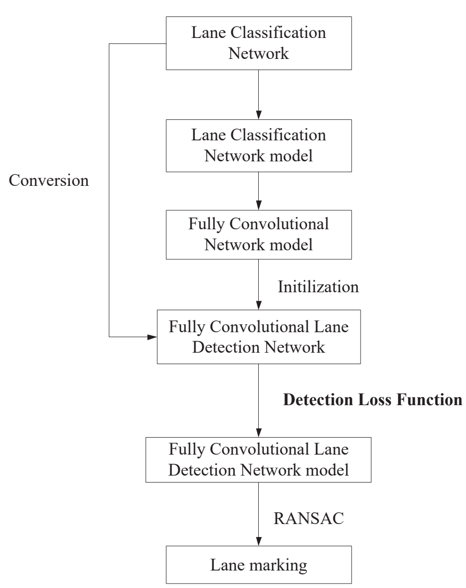


Fig. 1. The overview of the proposed approach.

II. Proposed Approach

In lane classification, we build a small neural network for feature extraction from lane classification dataset. In lane detection, the detection network is constructed through substituting the fully connected layers in the classification network with convolution layers. With initializing detection network layers by parameters of classification network model, a detection network model is obtained by training detection network with proposed detection loss function. With detected lane pixels, the lane marking is executed by RANSAC. The overview of proposed approach is shown in Fig. 1.

A. Lane Classification

Lane classification dataset is obtained by cropping lane regions of images in video sequences. The total lane images are divided into three categories: the white lanes, the yellow lanes and the background. The cropped lane images are resized to 32×32. The proposed classification network includes three convolutional structures and two fully connected layers. For each convolutional structure, the feature maps' size is kept in convolution layers and reduced in Max Pooling layers. SoftMax layer is at the terminal of classification network to calculate the probability of each category. The architecture of whole lane classification network is shown in Table I.

After obtaining the model with good performance on lane classification, the classification network is transformed into a fully convolutional detection network. Furthermore, the classification network model is converted into a fully convolutional network model to initialize the detection network.

TABLE I
The Architecture of Lane Classification Network.

Type	Structure
Input	3× 32× 32
Conv	kernel:32× 5× 5, stride: 1, pad: 2, ReLU
Pool	Max,kernel: 2× 2, stride: 2
Conv	kernel:32× 5× 5, stride: 1, pad: 2, ReLU
Pool	Max,kernel: 2× 2, stride: 2
Conv	kernel:32× 3× 3, stride: 1, pad: 1, ReLU
Pool	Max,kernel: 2× 2, stride: 2
FC	64,ReLU
SoftMax	3

B. Lane Detection

Since the classification network model is applied to initializing the detection network, two networks should have the same number of parameters. As the input size of the classification network is 32×32, the size of the input feature maps for the first fully connected layer is 4×4. Therefore, the kernel size of convolution layer converted from the first connected layer should be 4×4. For the rest of fully connected layers, the kernel size of converted convolution layers should be 1×1. Besides, the kernel numbers of converted convolution layers are consistent with the output number of previous fully connected layers.

When raw images in video sequences are fed to the detection network, the output of the last convolution layer before detection loss layer is feature maps (also called final feature maps in this paper), not a specific predicted label. As video sequences provide the lanes locations in images, we build a three-channel array with the same size as input images to store the information of lanes location. And each element in each channel of this array is 0 or 1. In the first channel, '1' presents the current pixel is a background pixel and '0' for a lane pixel. In the second channel, '1' is for the yellow lane pixel and '0' for background or white lane pixel. For the last channel, the value for a white lane pixel is '1' and other pixels are '0'. Due to the translation invariance of convolution layers, label maps can be obtained by resizing above array with the size of final feature maps. Fig. 2 displays the whole structure of detection network in training process.

The detection loss function is designed to calculate the classification loss and regression loss between label maps and output maps. Classification loss is the loss between predicted labels of lane line and true labels of the lane line. Regression loss, namely location loss, represents the loss between predicted coordinates and true coordinates of lane pixels.

1) Classification Loss: As the element of label maps is 0 or 1, a sigmoid function is applied to compressing values of final feature maps into values in interval $[0, 1]$. The detection dataset labels have the same dimension with final feature maps. Classification loss is a sum of L2 loss between pixels on final feature maps and label maps. Equation (1) and (2) show the method of calculating classification loss.

$$L_C = \frac{1}{M \times K \times H \times W} \sum_{i=1}^{N} \sum_{m=1}^{M} \sum_{k=1}^{K} l_{i,m,k} \tag{1}$$

$$l_{i,m,k} = \left\| p\left(m, k, h_i, w_i\right) - g\left(m, k, h_i, w_i\right) \right\|^2 \tag{2}$$

Where, M represents the number of input images of detection network. K represents the number of channels of label maps, which is the same to the number of categories of lanes. N is the quantity of pixels exploited in different channels of final feature maps. (h_i, w_i) presents the coordinate of $i-th$ selected pixel. H represents the height of sub-map in each channel. W represents the width of sub-map in each channel. $l_{i,m,k}$ is the classification loss of $i-th$ lane pixel in $k-th$ channel of $m-th$ input image. $g(m, k, h_i, w_i)$ is the true value of label k at coordinate (h_i, w_i) in $k-th$ feature map of $m-th$ input image. $p(m, k, h_i, w_i)$ is the predicted compressed value at coordinate (h_i, w_i) in $k-th$ final feature map of $m-th$ input image. Equation (3) shows how to calculate the predicted compressed value of each pixel in final feature maps.

$$p = (m, k, h_i, w_i) = \frac{1}{1 + \exp\{-y(m, k, h_i, w_i)\}} \tag{3}$$

Where, $y(m, k, h_i, w_i)$ represents the value of pixels in final feature maps.

2) Regression Loss: The location information of each lane can be obtained from the detection dataset label maps. For each lane, there is a rectangle (called label rectangle) which can cover all pixels in it. Calculation of the regression loss is only executed for pixels which are detected as lane pixels in each label rectangle. As for pixels which are outside all label rectangles or detected as background in label rectangles, we only calculate the classification loss defined in (1) and (2).

The coordinates of lane pixels can be obtained by searching '1' in detection dataset label maps. Then quadratic curve fitting is performed for each lane. The fitting results are quadratic curve coefficients. In quadratic curve fitting, the height coordinate of a pixel is set as dependent variable and width coordinate as independent variable. The curve equation is shown in (4).

$$h = a_0 + a_1 w + a_2 w^2 \tag{4}$$

After getting curve coefficients, w in (4) is substituted with width coordinates of detected pixels in label frames to obtain true height coordinates. For detected lane pixels in label rectangles, we calculate L2 loss between the detected height coordinates and true height coordinates to get regression loss. The quadratic curve coefficients are obtained through least square method displayed in (5) and (6).

$$M_a = \left({M_w}^T M_w\right)^{-1} {M_w}^T M_h \tag{5}$$

$$M_w = \begin{pmatrix} 1 & w_1 & w_1^2 \\ 1 & w_2 & w_2^2 \\ \vdots & \vdots & \vdots \\ 1 & w_m & w_m^2 \end{pmatrix} M_h = \begin{pmatrix} h_1 \\ h_2 \\ \vdots \\ h_m \end{pmatrix} M_a = \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} \tag{6}$$

Where, w_i is the width coordinate of lane pixels in detection dataset label maps. h_i is the height coordinate of lane pixels in detection dataset label maps. And a_i is label quadratic curve coefficient. Equation (7) shows how to calculate the regression loss of detected lane pixels in each label rectangle of lanes.

$$l_{j,k} = \sum_{i}^{n} \left(\frac{h_{j,i,k} - h'_{j,i,k}}{H}\right)^2 \tag{7}$$

Where, $l_{j,k}$ represents the regression loss of all lane pixels in $j-th$ label rectangle of $k-th$ channel. Specially, the regression loss is only executed on the yellow lane channel and white lane channel of feature maps. H is the height of feature maps. $h_{j,i,k}$ is the height coordinate of $i-th$ detected lane pixel in $j-th$ label rectangle of $k-th$ channel. n represents the number of total detected lane pixels in $j-th$ label rectangle of $k-th$ channel. $h'_{j,i,k}$ is the true height coordinate of $i-th$ detected lane pixel in $j-th$ label rectangle of $k-th$ channel and obtained by $h'_{j,i,k} = a_{1,j,k} + a_{2,j,k} w_{j,i,k} + a_{3,j,k} {w_{j,i,k}}^2$. And $a_{1,j,k}$, $a_{2,j,k}$ and $a_{3,j,k}$ are label quadratic curve coefficients of $j-th$ label rectangle in $k-th$ channel. The total regression loss is displayed in (8).

$$L_R = \frac{1}{M \times K \times H \times W} \sum_{m=1}^{M} \sum_{k=2}^{K} \sum_{j=1}^{J} l_{m,j,k} \tag{8}$$

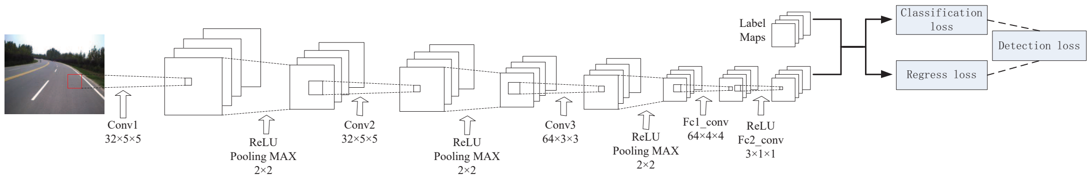


Fig. 2. The overview of detection network.

Where, M, K, H and W are constants in (1). $l_{m,j,k}$ is the regression loss of the total lane pixels in $j-th$ label rectangle of $k-th$ channel of $m-th$ input image. For each channel, the total number of label frames is different. J represents the number of label rectangles in current channel. When the detection of yellow lanes and white lanes is right, detection results are also highly confident on background channel. So calculation of regression loss for the background channel is not necessary in the optimization.

The total detection loss is shown in (9). α represents the proportion of classification loss. β represents the proportion of regression loss in detection loss.

$$L = \alpha L_C + \beta L_R \tag{9}$$

C. Lane Marking

The lane marking is executed on detected lane pixels by detection network model on resized test images, which have the same size to final feature maps. The information of surrounding pixels can be used to remove noise pixels for modifying original detection results. In clustering process, lane pixels of different lanes are put into different sets. Then we apply RANSAC to merge the clustered pixels to mark lanes. At last, we resize the fitted lanes to mark lane regions on original input images.

The whole lane marking process is shown in Algorithm 1. P is a one-channel map whose size is the same as resized test images. The value of P 's element represents lane category of pixels in resized test images. D is an improved results map of P modified through the information of surrounding pixels. And the absolute difference between width (or height) coordinate of surrounding pixels and p is one. The indexes of elements in list l represent different lane categories. The index '0' is corresponding to background, '1' is yellow lane and '2' for white lane. $maxn$ is utilized to check the voting tie. g in G is the set of pixels in the same lane. Through applying RANSAC to g, the coefficients of the fitting line function and coordinates of pixels used to get the coefficients can be obtained. However, the pixels in same dash lines are divided into different sets. Then the slopes of fitted lines are transformed into angles so that the sets which have close angles and intercepts can be merged by setting proper thresholds.

Algorithm 1 Lane marking algorithm

original detection results map P, improved detection results map D.
for each $p \in P$ **do**
    count the numbers of pixels for each category around p and store them in list l;
    $maxn =$ the number of max elements in l;
    if $maxn == 1$ **then**
        change the prediction of p to the index of max element in l;
    end if
    push the modified result of p into D;
end for
for each lane category **do**
    clustered detection results G.
    for each $d \in D$ **do**
        $flag = 0$;
        get surrounding pixels sp of d;
        for each $g \in G$ **do**
            if $sp \in g$ **then**
                add d to g;
                $flag = 1$;
                break;
            end if
        end for
        if $flag = 0$ **then**
            create a new group in G with d;
        end if
    end for
    Empty RANSAC results set R.
    for each $g \in G$ **do**
        $r =$ RANSAC result of g;
        push r to R;
    end for
    merge the close r in R and mark the lane.
end for

III. Experiment Results

A. Lane Classification

Since classification dataset is obtained by cropping video frames, the number of background images is much larger than the number of lane images in classification dataset. To balance the image numbers of each category in classification dataset, data expansion is applied to lane images by adding salt and pepper noise on obtained lane images. Stochastic gradient descent is deployed to optimize SoftMax loss. Learning rate is kept on 0.001 during the whole training. The classification model is obtained by training classification network with 200 epochs and evaluated by accuracy confusion matrix displayed in Table II.

In Table II, the highest accuracy, which is 98.37%, occurs at white lane. Even the lowest accuracy is 97.53% on background. The mistaken proportion is very small to guarantee trained model has extracted the suitable feature for lane detection.

TABLE II
ACCURACY CONFUSION MATRIX OF LANE CLASSIFICATION NETWORK (%).

	Background	Yellow lane	White lane
Background	97.53	0.43	2.04
Yellow lane	0.11	97.92	1.97
White lane	0.38	1.25	98.37

B. Lane Detection

As mentioned in last subsection, background regions (negative samples) have more pixels than lane regions. When we directly optimized detection loss on whole feature maps, the full image is detected as the background region. Although the detection loss is decreased to the minimum, lane pixels cannot be identified. Hence, it's necessary to tune the proportion between background pixels and lane pixels. To control the number of background pixels in the optimization, we multiply the number of total lane pixels with a factor, called sample ratio. With randomly selecting pixels in background region, each negative sample can be made use of in optimization.

Stochastic gradient descent is also applied in the optimization of detection loss. As the training of detection network is fine-tuning on the model converted from classification network model, learning rate is fixed to 0.00001. Moreover, it is enough to obtain a detection network model through training detection network with 40 epochs. At first, the sample ratio is set to 2 and the proportion between the classification loss and regression loss is kept on $1 : 1$. The detection results in some cases are shown in Fig. 3. All detection results are displayed on original images instead of resized images. We observe that there are some discrete noise pixels in detection results. In some cases, it is easy to utilize the information of pixels around noise pixels to filter them out. However, the cars, whose feature are similar with lanes, normally cause more noise pixels. These noise pixels can only be removed by RANSAC process. In following experiments, the improved detection results are deployed to compare the performances of different models trained by different parameters.

In order to obtain the proper proportion between classification loss and regression loss, the sample ratio is fixed to 2 and detection network is trained with proportions of $[2 : 1, 1 : 1, 1 : 1.2, 1 : 1.5, 1 : 1.7, 1 : 2]$. The comparison of detection results is shown in Fig. 4. When the classification loss has higher or equal proportion with the detection loss,

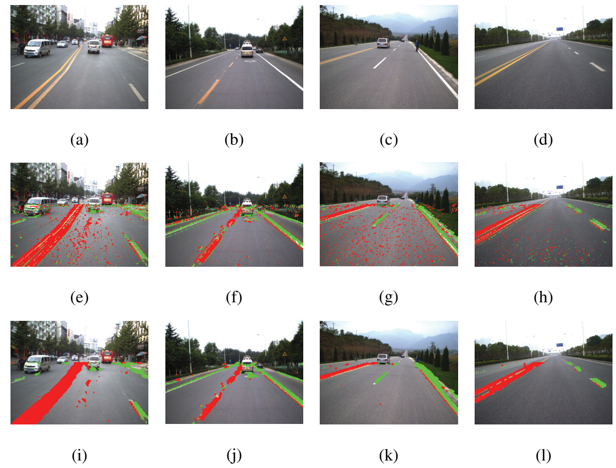


Fig. 3. The original detection results and improved detection results. Fig (a)-(d) in the first row are test images. Fig (e)-(h) in the second row are the detection results. Fig (i)-(l) in the third row are the improved detection results by surrounding pixels. We mark the detected yellow lane pixels with red pixels and white lane pixels with green pixels.

many white lane pixels are wrongly detected as yellow lanes. However, there are more other regions pixels detected as white lane pixels when β is more bigger than α. Therefore, the difference between β and α cannot be large and β should be a little bigger than α. We conclude that proportion $1 : 1.2$ is proper for detection network training.

Above experiment results are obtained by keeping sample ratio on 2, we examine different sample ratios with $\alpha : \beta = 1 : 1.2$. Fig. 5 displays the detection results for different sample ratios in $[1, 2, 3, 4, 5, 6, 10]$. When the sample ratio is 1, many region pixels are wrongly detected as yellow lane pixels. And the successfully detected lane pixels become less and less as the sample ratio increasing. There are almost no detected lane pixels in detection results of sample ratio '6' and '10'. Obviously, the sample ratio '2' is proper for detection task with enough detected lane pixels and less mistaken pixels.

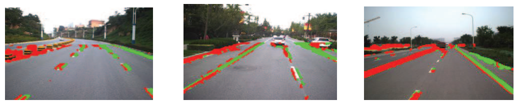
Experiment results show that the best model is obtained by setting sample ratio to '2', α to '1' and β to '1.2'. The good detection results show that the designed detection loss function can train the network to output lane categories and locations in different scenes.

C. Lane Marking

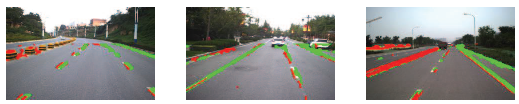
Due to low confidence, we remove the lane sets whose pixels number is less than 30. Moreover, the dash lanes are connected in images. The lanes in near-view with more detected pixels are easy to be detected accurately. Since the horizontal boundary of the road and sky has influence on the lane marking, it's necessary to screen out the marked lanes whose angles are smaller than 10 degrees or bigger than 170 degrees. The range of fitted lanes' angles is from 0 degree to 180 degrees.

The performance of detection model in one scene with high road reflection is not very good. The scene and detection

(a) Test images.



(b) α : 2, β : 1.



(c) α : 1, β : 1.

(d) α : 1, β : 1.2.

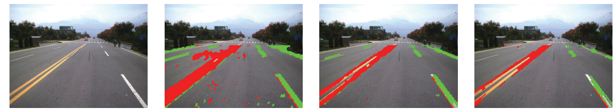
(e) α : 1, β : 1.5.

(f) α : 1, β : 1.7.

(g) α : 1, β : 2.

Fig. 4. The performance of models with different ratios between α and β.

result are displayed in Fig. 6. The strong reflection makes the feature difference between the background and lane regions very smaller. The detection network model detects a lot of noise pixels in the reflection zone. Modification process and RANSAC process cannot remove the large amount of noise pixels, either. Through checking detection train dataset, we found the number of images for high reflection is low. The high reflection images are hard to collect in normal road conditions. However, the performance of best model is good in the rest scenes. The lane marking results in another 28 road scenes are shown in Fig. 7. The lane marking results shows that the proposed approach has good performance in 28 different scenes. There is a little marking offset for the lane covered by people or cars, which cause more noise pixels. The marking of broken lanes also has slight deviation because of irregular

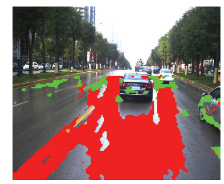


(a) Resized test image (b) Sample ratio : 1 (c) Sample ratio : 2 (d) Sample ratio : 3

(e) Sample ratio : 4 (f) Sample ratio : 5 (g) Sample ratio : 6 (h) Sample ratio : 10

Fig. 5. The detection results of different sample ratios.

(a) High reflection scene.



(b) Improved detection result

Fig. 6. The bad results in high reflection scene.

lane pixels distribution. However, these small deviations can be ignored in lane marking on whole image. The trained detection network model takes 54.85 ms in lane pixels extraction on a single GTX Force 1080Ti. The detection accuracy of the best model can reach 82.24% on detection test dataset.

IV. Conclusions

In this paper, a traffic lane detection method using fully convolutional neural network is proposed. A small and shallow neural network is built to extract suitable lane feature. The classification accuracy on each kind of lanes is larger than 97.5%. In lane detection, it is very difficult to obtain a good detection network model when detection network is trained on images of different road conditions. Although with not good detection results in one high reflection scene, the detection accuracy of the best model can still reach 82.24% on whole test dataset. Finally, the detected lane pixels are clustered and mark the full lane with RANSAC. In future, fine-tuning the obtained network model will be performed by images in different weather and illumination conditions to improve the detection performance. Moreover, lane marking algorithm will be improved to revise the deviation of fitted lanes.

Acknowledgment

This work was sponsored in part by the National Natural Science Foundation of China (61702418, 61602383 and 61772424), the Fundamental Natural Science Research Funds of Shaanxi Province (2018JQ6016 and 2017JQ6019), the Seed Foundation of Innovation and Creation for Graduate Students in Northwestern Polytechnical University (ZZ2018119).

Fig. 7. The lane marking results in different road scenes.

REFERENCES

[1] Huachun Tan, Yang Zhou, Yong Zhu, Danya Yao, and Jianqiang Wang, "Improved river flow and random sample consensus for curve lane detection," *Advances in Mechanical Engineering*, vol. 7, no. 7, pp. 1687814015593866, 2015.

[2] Yue Wang, Eam Khwang Teoh, and Dinggang Shen, "Lane detection and tracking using b-snake," *Image & Vision Computing*, vol. 22, no. 4, pp. 269–280, 2004.

[3] D. Schreiber, B. Alefs, and M. Clabian, "Single camera lane detection and tracking," in *Intelligent Transportation Systems, 2005. Proceedings*, 2005, pp. 302–307.

[4] Ruyi Jiang, Klette Reinhard, Vaudrey Tobi, and Shigang Wang, "Lane detection and tracking using a new lane model and distance transform," *Machine Vision & Applications*, vol. 22, no. 4, pp. 721–737, 2011.

[5] Amol Borkar, Monson Hayes, and Mark T. Smith, "Robust lane detection and tracking with ransac and kalman filter," in *IEEE International Conference on Image Processing*, 2010, pp. 3225–3228.

[6] Zhenqiang Ying and Ge Li, "Robust lane marking detection using boundary-based inverse perspective mapping," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2016, pp. 1921–1925.

[7] Jun Li, Mei Xue, Danil Prokhorov, and Dacheng Tao, "Deep neural network for structural prediction and lane detection in traffic scene," *IEEE Transactions on Neural Networks & Learning Systems*, vol. 28, no. 3, pp. 690–703, 2017.

[8] Dajun Ding, Jongsu Yoo, Jekyo Jung, and Sungho Jin, "Various lane marking detection and classification for vision-based navigation system?," in *IEEE International Conference on Consumer Electronics*, 2015, pp. 491–492.

[9] Heechul Jung, Junggon Min, and Junmo Kim, "An efficient lane detection algorithm for lane departure detection," vol. 36, no. 1, pp. 976–981, 2013.

[10] Avishek Parajuli, Mehmet Celenk, and H. Bryan Riley, "Robust lane detection in shadows and low illumination conditions using local gradient features," *Open Journal of Applied Sciences*, vol. 3, no. 1, pp. 68–74, 2013.

[11] Hunjae Yoo, Ukil Yang, and Kwanghoon Sohn, "Gradient-enhancing conversion for illumination-robust lane detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1083–1094, 2013.

[12] Jongin Son, Hunjae Yoo, Sanghoon Kim, and Kwanghoon Sohn, "Real-time illumination invariant lane detection for lane departure warning system," *Expert Systems with Applications*, vol. 42, no. 4, pp. 1816–1824, 2015.

[13] Upendra Suddamalla, Saikat Kundu, Sameer Farkade, and Apurba Das, "A novel algorithm of lane detection addressing varied scenarios of curved and dashed lanemarks," in *International Conference on Image Processing Theory, TOOLS and Applications*, 2016, pp. 87–92.

[14] Umar Ozgunalp and Naim Dahnoun, "Robust lane detection & tracking based on novel feature extraction and lane categorization," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2014, pp. 8129–8133.

[15] A Borkar, M Hayes, M. T Smith, and S Pankanti, "A layered approach to robust lane detection at night," in *Computational Intelligence in Vehicles and Vehicular Systems, 2009. CIVVS '09. IEEE Workshop on*, 2009, pp. 51–57.

[16] Martin A. Fischler and Robert C. Bolles, *Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography*, ACM, 1981.

[17] Yifei Wang, Naim Dahnoun, and Alin Achim, "A novel system for robust lane detection and tracking," *Signal Processing*, vol. 92, no. 2, pp. 319–334, 2012.

[18] Vitor S. Bottazzi, Paulo V. K. Borges, Bela Stantic, and Jun Jo, "Adaptive regions of interest based on hsv histograms for lane marks detection," *Advances in Intelligent Systems & Computing*, vol. 274, pp. 677–687, 2014.

[19] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 580–587.

[20] J. R. R. Uijlings, K. E. A. Van De Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *International Journal of Computer Vision*, vol. 104, no. 2, pp. 154–171, 2013.

[21] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, "Faster r-cnn: towards real-time object detection with region proposal networks," in *International Conference on Neural Information Processing Systems*, 2015, pp. 91–99.

[22] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng Yang Fu, and Alexander C. Berg, "Ssd: Single shot multibox detector," pp. 21–37, 2015.

[23] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi, "You only look once: Unified, real-time object detection," in *Computer Vision and Pattern Recognition*, 2016, pp. 779–788.

[24] Brody Huval, Tao Wang, Sameep Tandon, Jeff Kiske, Will Song, Joel Pazhayampallil, Mykhaylo Andriluka, Pranav Rajpurkar, Toki Migimatsu, and Royce Chengyue, "An empirical evaluation of deep learning on highway driving," *Computer Science*, 2015.

[25] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun, "R-fcn: Object detection via region-based fully convolutional networks," 2016.