Sensors and Materials, Vol. 32, No. 12 (2020) 4505–4516
MYU Tokyo

S & M 2425

Improved Lane Detection Method Based on Convolutional Neural Network Using Self-attention Distillation

Xinyu Zhang, He Huang,* Weiming Meng, and Dean Luo

School of Geomatics and Urban Spatial Information, Beijing University of Civil Engineering and Architecture,
No. 15, Yongyuan Road, Huangcun Town, Daxing District, Beijing 102616, China

(Received September 29, 2020; accepted December 9, 2020)

Keywords: high-precision map, semantic segmentation, lane detection, DC-VGG-SAD, detection accuracy, detection speed

With the rapid development of autopilot technology and various types of sensor, high-precision maps containing a large amount of information for assisting driving have been proposed. The standard lane line detection algorithm relies on the robust estimation of visible lane line markers from a camera image using vision and image processing algorithms. Although the recognition and detection technology for road marking lines is relatively mature, some problems still exist, such as poor detection accuracy and unsatisfactory real-time performance. To solve the problems of the poor robustness and low running speed of the current lane detection methods in complex environments, in this study, we improve current lane detection methods from the perspective of semantic segmentation and propose a DC-VGG-SAD network (VGG: visual geometry group), in which dilated convolution (DC) is used to reduce the complexity of the network to ensure detection accuracy. Furthermore, adding self-attention distillation (SAD) makes the information transmission faster. The proposed network was experimentally evaluated using two large-scale datasets. It was found that when dealing with lane lines in complex environments, the network offers higher detection accuracy and detection speed than most current mainstream networks.

1. Introduction

With the rapid development of high-precision optical and electronic sensors, and highly efficient and highly effective computer vision and machine learning algorithms, real-time driving scene understanding has become increasingly realistic. As one of the key elements of self-driving technology, high-precision maps are attracting wide attention.(1) Compared with the "qualitative" description of traditional maps, high-precision maps are more of a "quantitative" description. They not only have more accurate coordinate information, but also have more accurate and richer road semantic information, which can provide accurate positioning, auxiliary environment perception, and decision planning for self-driving cars while improving the safety of autonomous driving.

*Corresponding author: e-mail: huanghe@bucea.edu.cn
https://doi.org/10.18494/SAM.2020.3128

ISSN 0914-4935 © MYU K.K.
https://myukk.org/

Lane markings are an important traffic safety feature with the functions of separating road areas, stipulating the direction of travel, and providing guidance information for pedestrians. In addition, as one of the core elements of an auxiliary environment awareness module, lane detection technology is crucial in the autopilot field(2) and is a powerful tool not only for building high-precision maps but also for verifying their accuracy.(3) Owing to its low cost and strong real-time performance, vision-based lane detection has become the key to full autopilot driving.(4) Most traditional lane detection methods use the characteristics of the lane(5) but need to find specific lane features manually, such as lane edge features(6) and color.(7) The low-level features extracted by traditional methods can be combined with the Hough transform,(8) Kalman filter,(9) or particle filter(10) to detect lanes. Traditional methods are easy to use but are frequently disturbed by noise sources such as object occlusion, low light intensity, and incomplete lane markings. This means that the algorithm cannot achieve the required detection accuracy in a complex environment or poor weather and is not robust.(11) When there are no markings on lanes, neither the feature-based nor model-based traditional lane detection methods can effectively segment the lane structure.

Many lane detection methods use deep learning to solve these problems. Lane detection methods based on deep learning or the strong feature extraction ability of convolutional neural networks (CNNs) are used to obtain the lane features.(12) By dividing the lane sign in an image into small image blocks and tagging the bounding boxes of the blocks, the lane line is transformed into a target detection problem. He *et al.* designed a type of network called a dual-view convolutional neutral network (DVCNN), which introduced the lane sign images on a perspective plane and a top plane into the network as a patch, separated the lane from other types of lane sign such as arrows, and then obtained the lane through a weighted cap filter and a global optimization function.(13) Lee *et al.* proposed a vanishing point guided network (VPGNet) based on a CNN with a multitask structure, which used manually calibrated vanishing points as constraints. They gridded the dataset to realize a special labeling method and obtained good detection results. However, using this type of target detection method to solve the problem of lane detection is complex and requires a large amount of labeled information support, resulting in a low operation speed in the later stages.(14)

Another approach is to use the method of image segmentation to separate the lane as a foreground from other regions and to solve the problem of lane detection from the perspective of semantic segmentation. Pan *et al.* proposed an image semantic segmentation method called a spatial CNN (SCNN). They used a visual geometry group (VGG)(15) network as the basic framework and transformed the traditional convolutional connection based on a layer-to-layer approach to a new convolutional operation based on a slice-to-slice approach. This method resolved the situation when a lane is blocked.(16) Kim and Park proposed an end-to-end transfer learning method, which combined a CNN with the random sample consensus (RANSAC) algorithm to detect the lanes to the left and right of the current lane.(17) Neven *et al.* proposed a network called LaneNet based on the SegNet(18) network infrastructure, which is composed of an encoder–decoder structure.(19) The main feature of the structure is that it contains two decoders, one to detect the segmented branch of the channel and the other to segment the embedded branch of the road region. However, the CNN in most of the above methods contains a large number of parameters, which leads to a lower operation speed, a more complex network structure, and a larger number of downsampling layers, which in turn lead to a reduction in spatial resolution and loss of detail in the output image.

Hence, in this paper, we propose a DC-VGG-SAD network, which is based on a full convolutional network (FCN)(20) with an optimization algorithm. This proposed network improves the lane detection method of a VGG network and uses dilated convolution (DC)(21) to remove part of the downsampling layer in the network so that the output image of the network has a larger receptive field. The complexity of the network can be reduced, ensuring detection accuracy. Moreover, adding self-attention distillation (SAD)(22) between the encoder and decoder of the network speeds up the information transmission. In this study, experiments were carried out on a network based on two large network datasets: TuSimple(23) and CULane.(15)

2. Network Architecture

2.1 DC-VGG network

An FCN is different from the traditional CNN in that it replaces fully connected layers with a convolution layer, so it can input images of any size, making it suitable for semantic segmentation. As an encoder structure, the VGG-16 network can classify and analyze the local pixel values of images and obtain high-order semantic information. In the encoder stage, the last three layers of the VGG-16 network structure are fully connected layers, but too many fully connected layers will lead to a large number of parameters, thereby reducing the network computing efficiency. Therefore, the FCN converts the three fully connected layers into a convolutional layer. In this manner, the entire network is realized by a convolutional layer. The advantage of using a convolutional layer instead of fully connected layers is that it can retain the position information of the image and reduce the amount of calculation of parameters. The number of layers of the decoder corresponds to the number of network layers of the encoder, meaning that the deconvolution of the decoder is extended to four layers. According to the characteristics of the jump structure, the characteristic image output by the encoder is introduced into the decoder to reduce the amount of detailed information lost in the process of upsampling.

A dilation rate parameter is added in DC, which represents the number of points in the convolution kernel and increases the reception field of the model without reducing the size of the feature graph. The feature mapping of the network is magnified after the 10th convolutional layer, and the continuous downsampling operation distorts the network output image, so that the 11th to 13th convolutional layers are transformed into an extended convolution layer and the step size is doubled. Through the DC layer, the convolutional layer can achieve a larger receptive field, which makes the spatial information richer and the prediction results of pixels more precise. In addition, to ensure a finer segmentation effect of the network, a jump structure is used, i.e., the n-layer feature map is upsampled 2^n times to the size of the original image. The details of the structure are shown in Fig. 1. This type of jump structure can obtain finer effects while taking into account global information.

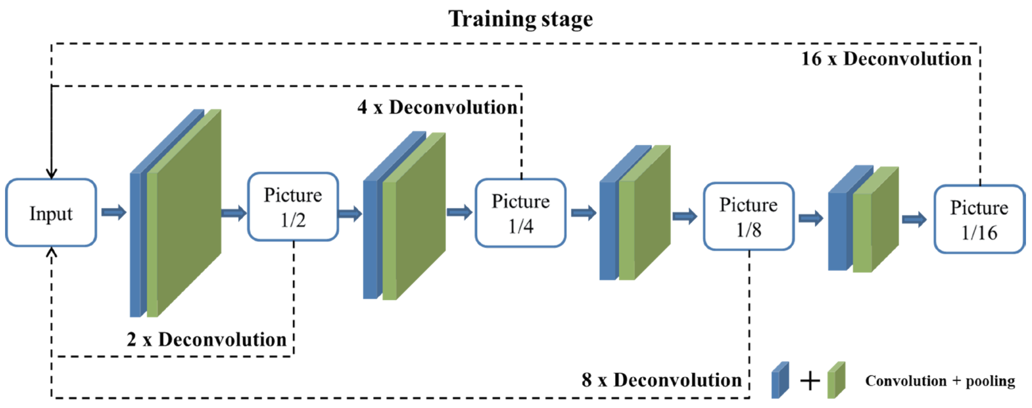


Fig. 1. (Color online) Jump structure used in training stage.

2.2 SAD mechanism embedding

As a simple yet novel approach, SAD allows a lane detection network to reinforce the representation learning of itself without the need for additional labels and external supervision. In addition, it does not increase the inference time of the base model. In a real-world traffic environment, lanes are not always continuous and sometimes may even be missing. The characteristics of SAD can overcome these problems. As the name implies, SAD allows a network to exploit attention maps derived from its own layers as the distillation targets for its lower layers. Such an attention distillation mechanism is used to complement the usual segmentation-based supervised learning.

Hence, we use a SAD connection between the encoder and the decoder. The convolutional operation is limited by the size of the convolutional kernel, and its receptive field can only cover a local area. Although we increase the receptive field by increasing the expansion convolution, the computational complexity of the entire network is still very large. SAD is one method that can be used to solve this problem. Using the structural characteristics of SAD, global context information can be introduced into the feature graph, which can also improve the transmission of spatial information. SAD allows us to train small networks with excellent visual attention on a par with that of very deep networks.

In summary, we enhance the representation learning of CNN-based lane detection models by SAD. SAD is only used in the training phase and incurs no computational cost during its deployment. We verify the usefulness of SAD in boosting the performance of lane detection networks. Our lightweight model, DC-VGG-SAD, achieves state-of-the-art lane detection performance on TuSimple and CULane. It can also serve as a strong backbone to facilitate future research on lane detection. Figure 2 shows the overall network architecture proposed in this study. After the operation of each convolution layer, pooled layer, and so forth, the output image is displayed in the form of pixels.

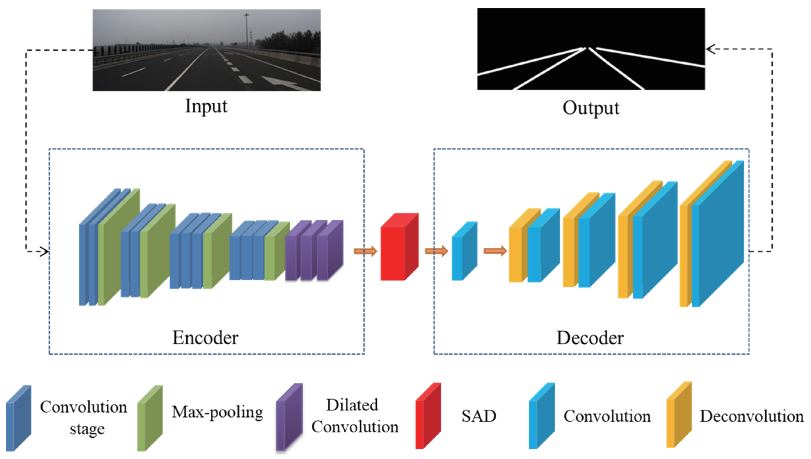


Fig. 2. (Color online) Overall architecture of network.

2.3 Loss function

"Learning" in a neural network refers to the process of automatically obtaining the optimal weight parameters from the training data. The purpose of learning is to find the weight parameters that can minimize the value of a loss function. In general, many neural network models dealing with classification problems use cross entropy as the loss function. If only the lane line is detected, the other areas outside the lane line in the image can be set as the background, which we regard as a simple dichotomy problem. However, there are many more background pixels than driveway pixels. Therefore, according to the discriminative loss based on the one-shot method proposed by De Brabandere *et al.*,[24] the role of this loss function is to make the pixel values of the same lane as close to the clustering center as possible while the pixel values of different lanes are as far away as possible from the corresponding clustering center. This is described in detail by Eqs. (1)–(3):

$$L_{var} = \frac{1}{C}\sum_{C=1}^{C} \frac{1}{N_C}\sum_{i=1}^{N_C} \left[\left\|\mu_C - x_i\right\| - \delta_v\right]_+^2, \tag{1}$$

$$L_{dist} = \frac{1}{C(C-1)}\sum_{C_A=1}^{C}\sum_{C_B=1}^{C} \frac{1}{N_C}\sum_{i=1}^{N_C} \left[\delta_d - \left\|\mu_{C_A} - \mu_{C_B}\right\|\right]_+^2, \tag{2}$$

$$LOSS_{ins} = L_{var} + L_{dist}, \tag{3}$$

where N_C is the number of lane lines, μ_C is the average embedding vector of a lane pixel set C relative to the clustering center, $\|\cdots\|$ represents distance, L_{var} and L_{dist} are variance and distance losses, respectively, and δ_v and δ_d are the differences between the two loss values. Through the action of these two loss functions, the pixel classification results of different lane lines are further optimized, thus improving the aggregation effect of pixels belonging to the same lane.

3. Experimental Analysis

3.1 Lane datasets

The proposed network structure was verified by performing many experiments on two widely used lane datasets: TuSimple and CULane.

The TuSimple dataset is used for the lane detection of structured roads, with the data collected from the San Diego Freeway in California in the United States, mostly under stable light conditions. In contrast, the images in the CULane dataset were collected in Beijing, one of the largest and most crowded cities in the world, and the dataset provides many challenging traffic scenarios for lane detection. Table 1 shows the detailed data of the two datasets.

The number of frames in the TuSimple dataset is much less than that in the CULane dataset, but the frames were taken on a highway and the environment is relatively simple. The frames in the CULane dataset were taken from complex road scenes, and we mainly use this dataset as a test set to verify the stability of the network.

3.2 Evaluation indexes of accuracy

Different datasets will have differences in the parameters and pixel size of image data due to differences in the collection vehicles and acquisition equipment. Thus, the accuracy evaluation indexes are also different after processing different datasets. In this paper, we use the official evaluation index standards of the TuSimple and CULane datasets.

(a) TuSimple dataset

The main evaluation indexes are accuracy, false positive, and false negative, as shown in Eqs. (4)–(6), respectively. Table 2 gives the meanings of the symbols in the equations.

$$Accuracy = \frac{\sum_{clip} C_{clip}}{\sum_{clip} S_{clip}} \tag{4}$$

$$FalsePositive = \frac{F_{pred}}{N_{pred}} \tag{5}$$

$$FalseNegative = \frac{M_{pred}}{N_{gt}} \tag{6}$$

Table 1
Dataset descriptions.

Dataset	Frames	Train	Validation	Test	Resolution	Scenarios	Environment
TuSimple	6408	3268	358	2782	1280 × 720	1	Highway
CULane	133235	88880	9675	34680	1640 × 590	9	Urban and highway

Table 2
Notation.

Variable	Definition
C_{clip}	Number of correct points detected
S_{clip}	Number of truth points
N_t	Number of lanes
F_{pred}	Number of lanes with error detected
M_{pred}	Number of lane lines missed
N_{gt}	Number of lanes in actual label
N_{pred}	Actual number of lanes

(b) CULane dataset

To judge whether a lane in the CULane dataset was successfully detected, we regard the lane as a line with a width of 30 pixels, and we calculate the intersection of the actual and predicted values of the lane divided by the union of the corresponding actual and predicted values. The result of this calculation is called the intersection–union ratio or Intersection over Union (IoU). Predictions greater than a specific threshold are regarded as true positives. Assuming that the threshold is strictly set to 0.5, the final F_1 score is used as the final evaluation index.

$$F_1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{7}$$

$$Precision = \frac{TP}{TP + FP} \tag{8}$$

$$Recall = \frac{TP}{TP + FN} \tag{9}$$

Equations (7)–(9) show how the accuracy is determined. *TP* is true positive, that is, the prediction is true and the reality is also true; *FP* is false positive, that is, the prediction is true but the reality is false; *FN* is false negative, that is, the prediction is false but the reality is true.

3.3 Experimental setup

In this study, the hardware environment used in the experiment included an Nvidia 1080Ti GPU equipped with 8 GB of video memory. The algorithm was developed using the PyCharm development tool and TensorFlow deep learning framework platform in a Windows 10 environment. In the process of training, the network model in this study did not use any pre-training model for optimization nor did it employ commonly used data enhancement methods such as random clipping, rotation, scale scaling, affine transformation, and random noise superposition. In this manner, the effects of different methods on the final lane detection results could be fully tested.

The gradient optimization strategy of adaptive moment estimation (Adam)(25) was used to train the accurate model rapidly and improve the convergence speed of the network's function.

The initial learning rate of the training set was 0.001 and the weight attenuation coefficient was 0.1. Because the video memory capacity of the GPU is only 8 GB, we set the batch size of the dataset to 2. All input images were rescaled from 1280 × 720 resolution to 512 × 320 resolution. Using the label coordinates provided by the dataset, we used the CVPolylines function in OpenCV to generate the lane line according to the method given in the official document and generated two types of grayscale image with a width of 30 pixels: a binary map with a white pixel value of 255 and another area with a black pixel value of 0. To distinguish different lane lines, different lane lines with different pixel values were marked and the background of the image was black.

3.4 Analysis of results

3.4.1 Analysis of effectiveness of DC

DC enlarges the receptive field of the convolutional kernel while keeping the number of parameters constant and ensuring that the size of the output feature map remains unchanged. Hence, to ensure the same receptive field, DC is added to the network to reduce the large number of parameters compared with that in the normal network, such that the amount of calculation greatly decreases and the running time shortens. When testing the TuSimple dataset, we used three evaluation indexes: accuracy, missed detection rate, and false detection rate. Because the false positive and false negative values are small and difficult to observe, to better visualize them, the values of these two indicators were subtracted from 1, and the resulting values were visualized.

As shown in Fig. 3, the accuracy improved from 94.39 to 94.78 when DC was added to the original network. When the SAD module was added to the network, the accuracy significantly improved to 95.67, which demonstrates that the addition of these two modules enables the

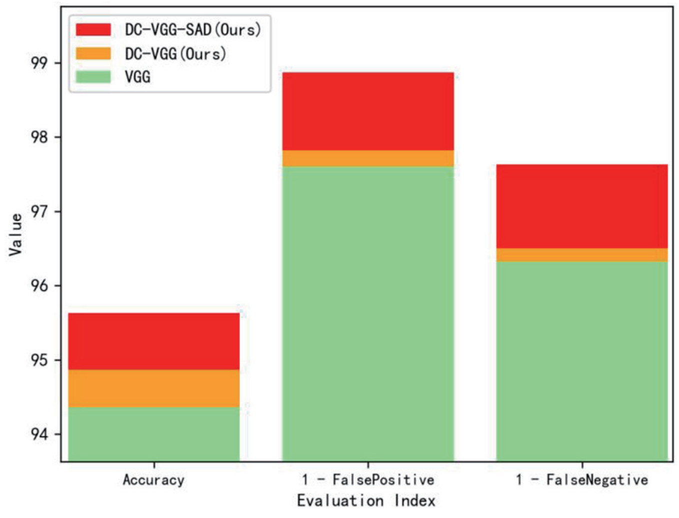


Fig. 3. (Color online) Accuracy evaluation of network.

network to better detect lane lines. It can also be seen from Fig. 4 that the addition of the extended convolutional structure can make up for the incompleteness of lane detection caused by occlusion and incomplete lanes.

The bottom three rows of Fig. 4 show results for the original network, the network with DC, and the network with DC and the SAD module from top to bottom. When the original images in the first row are detected, the results for the original network include missed and false detections. In the right column, for example, the picture in the first row is the original image, and the sky blue and white lines on the left in the second row are examples of false detection. There is a break in the middle of the yellow line in the image in the third row of the same column, which is a missed detection. In addition, adding the expanded convolutional network to the original network significantly improves the detection, making up for some of the incomplete lane detections caused by occlusion and incomplete lane lines, but there is still obvious roughness at the edges of the lane lines. After adding the SAD module, this problem is clearly reduced.

3.4.2 Analysis of effectiveness of SAD

Figure 4 shows that the SAD module helps optimize the lane shape. The use of SAD adds weights to the feature map, which enables the network model to better understand the spatial structure relationship, particularly at the edges of the lane lines. The results show that the edges of the lane lines are smoother, which can meet the requirements for lane detection in this study. For the TuSimple dataset, while ensuring detection accuracy, the detection speed of the network

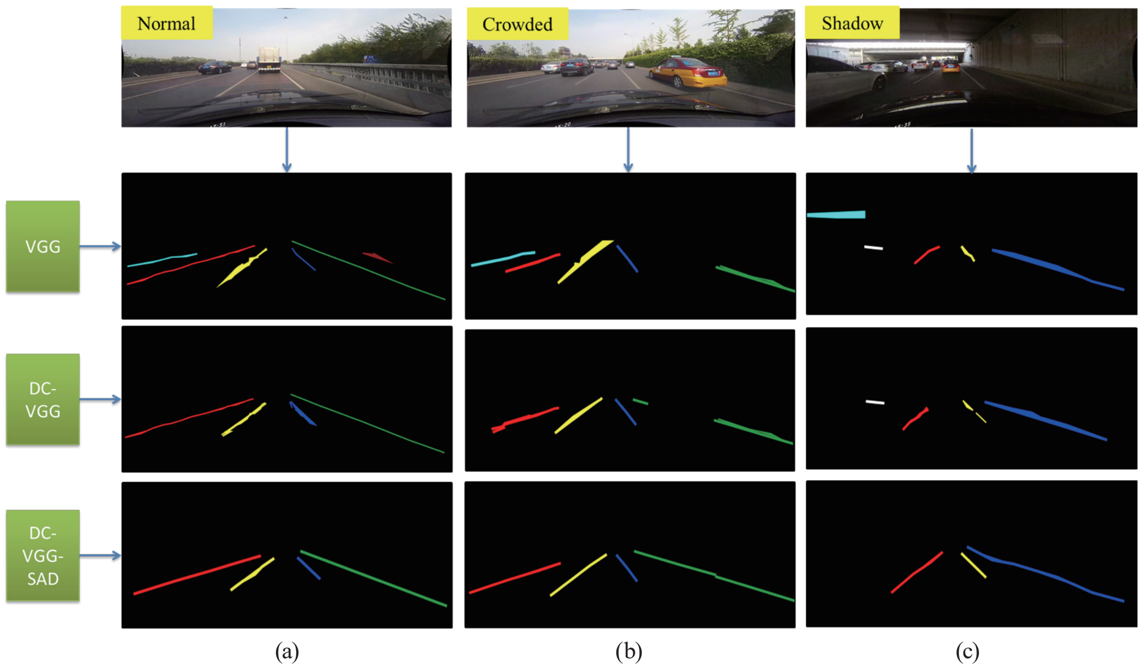


Fig. 4. (Color online) Effect of convolutional network on lane detection in different scenes. (a) Normal road environment; (b) crowded road environment; and (c) shadow road environment.

Table 3
Comparison of accuracies and running speeds of different networks.

Method	Accuracy	Runtime (ms)
VGG	94.39	20.7
SAD	93.64	13.2
SCNN	96.58	134.2
LaneNet	96.38	20.1
DC-VGG (ours)	94.78	12.7
DC-VGG-SAD (ours)	95.67	10.5

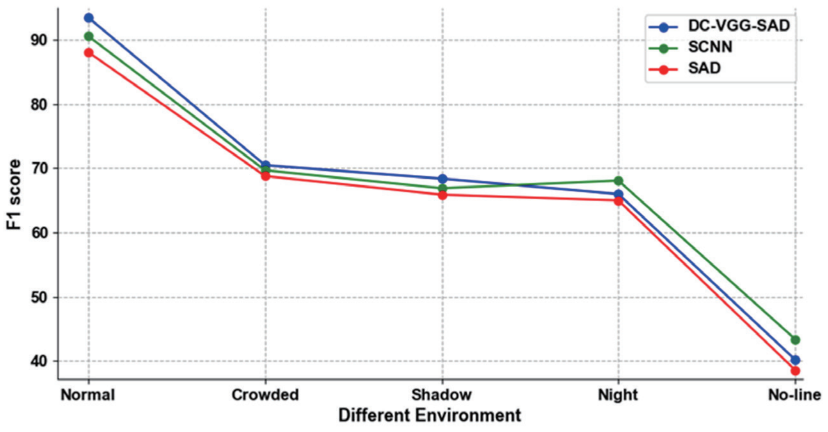


Fig. 5. (Color online) Comparison of lane detection accuracies of different networks in different environments.

proposed in this paper is higher than those of most mainstream convolutional networks, and the detection speed of the improved network is twice that of the original network. The results for the accuracy and speed of different methods can be seen in Table 3.

We also evaluated the proposed network through the CULane dataset. Figure 5 shows the lane detection accuracy for five different scenarios in CULane, and the proposed convolutional network is compared with the current high-quality network, with the F1 score used as the evaluation index. The data detection accuracy of the proposed DC-VGG-SAD network is slightly higher than those of other networks in normal driving scenarios, but at night or when there is no line, the detection accuracy is not as good as that of the SCNN network. Hence, the next direction in future research will be to solve the problem of accurate detection of lane lines in more complex environments.

4. Conclusion and Future Work

We proposed a lane detection network model based on improved semantic segmentation, which is mainly based on the VGG-16 and SAD mechanisms, to optimize the encoder–decoder in the network structure. In the encoder stage, the DC structure is used to replace the last three fully connected layers in the VGG-16 network. The SAD module stage is added between the encoder and the decoder, and a deconvolutional network with the same number of layers is set up in the decoder stage to convert the feature graph into an attention-weighted feature graph and accumulate the corresponding layers in the decoder process. The experimental results show that the detection speed of the modified network is twice that of the original network, and the detection accuracy is improved from 94.39 to 95.67. The data detection accuracy of the DC-VGG-SAD network proposed in this study is slightly higher than those of other networks in normal driving scenarios, which demonstrates that the proposed method can improve detection without the use of a complex post-processing mechanism. The experimental results also show that the accuracy of the original network is significantly improved.

In future work, the detection and classification results will be employed in a simple forward collision warning strategy, which may be very helpful for assisted driving in some simple structured environments. We also intend to fuse the proposed method with an electronic map or GIS to realize a better result. By matching the observed road states with a priori electronic map information, the detection and classification results can also be used to locate a self-driving vehicle accurately in the lateral direction. Moreover, the forward collision warning strategy also needs to be improved for use in daily driving.

Acknowledgments

This work was supported by the National Key R&D Program of China (No. 2017YFB0503700). We also thank LetPub (www.letpub.com) for its linguistic assistance during the preparation of this manuscript.

References

1 E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda: IEEE Access **8** (2020) 58443. https://doi.org/10.1109/ACCESS.2020.2983149

2 V. Kogan, I. Shimshoni, and D. Levi: IEEE Intelligent Vehicles Symp. (IV) (2016) 889. https://doi.org/10.1109/IVS.2016.7535493

3 N. Homayounfar, WC. Ma, SK. Lakshmikanth, and R. Urtasun: IEEE/CVF Conf. Computer Vision and Pattern Recognition (2018) 3417. https://doi.org/10.1109/CVPR.2018.00360

4 O. O. Khalifa, A.-H. A. Hashim, and A. A. M. Assidiq: IEEE Int. Conf. Semantic Computing (2009) 636. https://doi.org/10.1109/ICSC.2009.113

5 W. Song, Y. Yang, M. Fu, Y. Li, and M. Wang: IEEE Sens. J. **18** (2018) 5151. https://doi.org/10.1109/JSEN.2018.2832291

6 C. Lee and J. Moon: IEEE Trans. Intell. Transp. Syst. **19** (2018) 4043. https://doi.org/10.1109/TITS.2018.2791572

7 K. Chiu and S. Lin: IEEE Proc. Intelligent Vehicles Symp. (2005) 706. https://doi.org/10.1109/IVS.2005.1505186

8 R. Duda and P. Hart: CACM **15** (1972) 11. https://doi.org/10.1145/361237.361242

9 A. Borkar, M. Hayes, and M. T. Smith: IEEE ICIP (2009) 3261. https://doi.org/10.1109/ICIP.2009.5413980

10 Y. He, H. Wang, and B. Zhang: IEEE Trans. Intell. Transp. Syst. **5** (2004) 309. https://doi.org/10.1109/TITS.2004.838221

11 Y. Xing, C. Lv, L. Chen, H. Wang, H. Wang, D. Cao, E. Velenis, and F. Wang: IEEE EEE/CAA J. Autom. Sinica **5** (2018) 645. https://doi.org/10.1109/JAS.2018.7511063

12 R. Girshick, J. Donahue, T. Darrell, and J. Malik: IEEE Conf. Computer Vision and Pattern Recognition (2014) 1311. http://arxiv.org/abs/1311.2524

13 B. He, R. Ai, Y. Yan, and X. Lang: IEEE Intelligent Vehicles Symp. (IV) (2016) 1041. https://doi.org/10.1109/IVS.2016.7535517

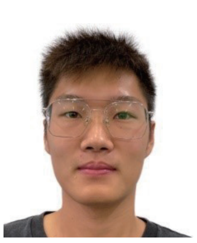
14 S. Lee, J. Kim, J. S. Yoon, S. Shin, O. Bailo, N. Kim, T. Lee, H. S. Hong, S. Han, and I. S. Kweon: IEEE Int. Conf. Computer Vision (ICCV) (2017) 1710. http://arxiv.org/abs/1710.06288

15 Simonyan, Karen, and A. Zisserman: ArXiv [Cs] (2015) 1409. http://arxiv.org/abs/1409.1556
16 X. Pan, J. Shi, P. Luo, X. Wang, and X. Tang: ArXiv [Cs] (2017) 1712. http://arxiv.org/abs/1712.06080
17 J. Kim, and C. Park: IEEE Trans. Pattern Anal. Mach. Intell. (CVPRW) (2017) 1194. https://doi.org/10.1109/CVPRW.2017.158
18 V. Badrinarayanan, A. Kendall, and R. Cipolla: IEEE Trans. Pattern Anal. Mach. Intell. **39** (2017) 2481. https://doi.org/10.1109/TPAMI.2016.2644615
19 D. Neven, B. D. Brabandere, S. Georgoulis, M. Proesmans, and L. V. Gool: IEEE Intelligent Vehicles Symp. (IV) **39** (2018) 1802. http://arxiv.org/abs/1802.05591
20 R. Girshick, J. Donahue, T. Darrell, and J. Malik: IEEE Conf. Computer Vision and Pattern Recognition (2014) 580. http://arxiv.org/abs/1311.2524
21 F. Yu and V. Koltun: CoRR (2016) 1511. http://arxiv.org/abs/1511.07122
22 Y. Hou, M. Zheng, C. Liu, and C. C. Loy: IEEE/CVF Int. Conf. Computer Vision (ICCV) (2019) 1013. http://arxiv.org/abs/1908.00821
23 TuSimple: http://github.com/TuSimple/tusimple-benchmark/issues/3 (accessed September 2018).
24 D. Brabandere, Bert, D. Neven, and L. V. Gool: ArXiv:1708.**02551** (2017) 1708. http://arxiv.org/abs/1708.02551
25 D. P. Kingma and J. Ba.: ArXiv:1412.6980 (2017) 1412. http://arxiv.org/abs/1412.6980

About the Authors

Xinyu Zhang received his B.S. degree from Beijing University of Civil Engineering and Architecture, China, in 2018. He is now studying for his M.S. degree at Beijing University of Civil Engineering and Architecture. His research interests are in high-precision navigation maps and artificial intelligence.

He Huang received his B.S. degree from Wuhan University, China, in 2000 and his M.S. and Ph.D. degrees from Sungkyunkwan University, South Korea, in 2004 and 2010, respectively. Since 2010, he has been a lecturer and associate professor at Beijing University of Civil Engineering and Architecture, China. His research interests are in autonomous driving, high-precision navigation maps, and visual navigation and positioning.



Weiming Meng received his B.S. degree from Beijing University of Civil Engineering and Architecture, China, in 2019. He is now studying for his M.S. degree at Beijing University of Civil Engineering and Architecture. His research interests are in multisensor fusion.

Dean Luo received his B.S. degree from Wuhan University, China, in 1990 and his M.S. and Ph.D. degrees from Southwest Jiaotong University, China, in 1997 and 2002, respectively. Since 2004, he has been a lecturer and professor at Beijing University of Civil Engineering and Architecture, China. His research interests are in GNSS, geodetic technology, and deformation-monitoring technology.