S & M 2780

Lane Line Detection Based on Improved Semantic Segmentation in Complex Road Environment

Chaowei Ma, Dean Luo, and He Huang*

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 July 16, 2021; accepted October 27, 2021; online published November 19, 2021)

Keywords: smart city, autonomous driving, complex road environment, lane line detection, semantic segmentation

With the concepts of smart city and smart travel and the rapid development of modern sensors, artificial intelligence, and other modern technologies, automatic driving technology that can effectively solve road congestion and ensure driving safety has become the main direction of future industry development. Accurate lane line technology is a fundamental technology for realizing autonomous driving. However, in actual road environments, lane lines are often detected with a low accuracy because of various factors, including light intensity changes and lane line obstruction, which greatly affect the safety of autonomous driving. To address the current challenges in lane line detection, in this study, we propose a lane line detection model based on improved semantic segmentation for complex road scenarios, such as lane line occlusion, mutilation, and shadowing. The Visual Geometry Group–Special Convolutional Neural Network (VGG-SS) proposed in this paper, which is based on the VGG-16 network, introduces a self-attentive distillation model and a spatial convolutional neural network (SCNN) model. Empirical results show that the proposed model outperforms the current semantic segmentation models, achieving better detection effects and a higher $F1$ value of 82.6 in complex road scenarios. The results prove that the proposed method can effectively improve the detection accuracy of lane lines.

1. Introduction

In recent years, autonomous driving technology has become a research hotspot in the field of intelligent transportation systems and has attracted considerable attention. The development of the self-driving car industry can not only effectively solve the problem of road congestion but also provide behavioral decisions for self-driving vehicles and guarantee the safety of vehicles, providing an important guarantee for achieving intelligent travel; examples include lane departure warning, lane keep assist, automatic lane change assist, and so forth.

Among the technologies for autonomous vehicle driving, environmental perception of road surface information has been considered an important aspect. As an important road part, a lane

*Corresponding author: e-mail: huanghe@bucea.edu.cn

https://doi.org/10.18494/SAM.2021.3544

line contains semantic information about road areas, specifies travel directions, and provides guidance information; therefore, the lane line detection technology based on a low-cost visual perception model has evolved. Owing to the rapid development of deep learning and artificial intelligence, the lane line detection technology can provide collision warning, lane departure warning, and auxiliary environment perception information to autonomous vehicles, as well as assist the autonomous driving system to realize the role of lane path planning, thus improving the safety of autonomous driving.

The current methods for lane line detection mainly include traditional methods based on feature detection and model building and deep-learning-based lane line detection methods. The problem of road or lane perception is a crucial enabler for advanced driver assistance systems.(1) The tensor-voting-based road lane recognition algorithm with road lane geometric constraints was presented by Wei *et al.*(2) Tapia-Espinoza and Torres-Torriti proposed an approach for lane segmentation and tracking that is robust to varying shadows and occlusions.(3) Fritsch *et al.* introduced a novel open-access dataset and benchmark for road area and ego-lane detection.(4) A Catmull-Rom spline-based lane model that describes the perspective effect of parallel lines was proposed for a generic lane boundary by Yue *et al.*(5) In feature-detection-based lane line detection methods, salient features such as lane line direction and length have been commonly used to obtain lane-line-related information.(6) Yoo *et al.* proposed a gradient enhancement conversion method based on linear discriminant analysis to generate new grayscale images from RGB color images and then used adaptive Canny edge detection, Hough transformation, and curve model fitting methods to obtain lane line information.(7) Lin *et al.* first grayed out the original image, set a double region of interest by a perspective transformation method, and performed a coarse feature detection, and then detected lane lines using the Hough transform.(8) Gaikwad and Lokhande used a segmented linear stretching function to improve the contrast of the region of interest and then employed the Hough transform to detect lane lines separately.(9) However, the lane line detection based on a grayscale difference in pixel point edge information has certain limitations and is affected by many factors, including shadows and light intensity variations.

Grayscale value refers to color intensity; grayscale denotes no color, and RGB color components are all equal. After graying, the dimensionality of the matrix decreases, the speed of the operation increases significantly, and the gradient information is still retained. If the image is not grayed out, the color information is still retained, so the object can be recognized on the basis of color information. In addition, when an image is grayed out, the color information is lost, so these types of detection algorithm do not have a generalization capability. However, if features with color information in road images could be detected effectively, the lane line information could also be obtained. Mammeri *et al.* proposed a lane line detection system combining the most stable extremal region and Hough transform, which uses matching features such as the color and shape of lane lines, to detect lane lines.(10) Sotelo *et al.* developed a road segmentation algorithm based on an HIS color space and a two-dimensional constrained space for obtaining the lane line information.(11) Ozgunalp and Dahnoun proposed a feature-map-based lane detection algorithm that uses an inverse perspective transformation method.(12) To improve the feature map signal-to-noise ratio, the feature map was matched by inverse perspective transformation, and lanes were detected by Hough transformation. Although the detection results of lane lines obtained using RGB images are better than those obtained using grayscale images, these methods cannot perceive local features of image data. Kumar *et al.* used the Kalman-filter-based tracking method to detect lane lines to solve the problem of low robustness of detection algorithms in illuminated scenes.(13) Chi *et al.* used the road vanishing point estimation method to detect lane lines, but their model-based method is computationally expensive.(14) In addition, the model-based detection method is computationally intensive and can perform well only in specific environments, which poses certain limitations.

Both feature- and model-based traditional lane line detection methods are susceptible to external environmental factors, and their robustness is extremely low when the lane lines are broken, obscured, or unpainted,(15,16) which can result in incorrect or even impossible lane line detection. To solve the problem of low accuracy of lane line detection in complex road environments, convolutional neural networks have been widely used for lane line detection owing to their powerful feature detection capability.(17,18)

In 2015, He *et al.* proposed the use of the SPP-net to improve the detection speed.(19) Ren *et al.* proposed the Fast-RCNN network, which was trained using a multitask loss function, allowing all layers to be updated while reducing the number of parameters in the fully connected neural network, and the detection performance was improved.(20) He *et al.* designed a new dual-view convolutional neural network strategy and used a weighted cap filter to obtain lane line information.(21) Aly used Gaussian filtering and detected street lanes using line detection and a new RANSAC spline fitting technique.(22) Kim and Park combined convolutional neural networks with the RANSAC algorithm and proposed a continuous end-to-end migration learning method that can detect both left and right lane lines of the current lane.(23) Neven *et al.* transformed the lane line detection problem into an instance segmentation problem that distinguishes the lane lines and their background using a binary classification principle.(24)

Considering that lane lines are striped targets with a strong structural continuity, the previously proposed networks that do not use the spatial relationship fully in lane line detection cannot achieve the required detection accuracy in complex road scenarios. To solve the problem of low lane line detection accuracy of the existing methods in complex road scenes, in this paper, we propose an end-to-end semantic segmentation network model based on the Visual Geometry Group–Spatial Convolutional Neural Network (VGG-SS), which represents an optimized VGG-16 network and improves the lane line detection accuracy by embedding a self-attentive distillation model between the encoder and the decoder, and a spatial convolution neural network (SCNN) model in the top implicit layer. The proposed model is trained with the CULane dataset using the designed reasonable hyperparameters and training strategies.

In this paper, we propose a lane line detection method based on improved semantic segmentation, which solves the problem of low detection accuracy because of damaged and obscured lane lines in complex road scenes.

2. Lane Line Detection in Complex Road Scenes

The semantic segmentation model for lane line detection is constructed using an encoder-decoder structure with reference to the U-net network structure,(25) and the spatial information acquisition of lane lines with a long-distance structure is improved by the detailed design of each part of the model feature encoder and via the introduction of additional operations, such as self-attention distillation (SAD), spatial convolution, and fusion upsampling.

2.1 Improved VGG-16 network

The classification performance of VGG-16 as a base network is very good; the network structure of VGG-16 is very regular and relatively easy to modify. The model trained on ImageNet has been published and can be fine-tuned on this basis for other datasets and has good adaptability to other datasets. There are many network structures using VGG-16 as a base network in the field of target detection, and the same effect is also very good. These advantages of VGG-16 made us choose it as the model for detection.

VGG-16 is a classical convolutional network model for image classification tasks, which was proposed in the ImageNet image classification and localization challenge in 2014.(26) The structure of VGG-16 is shown in Fig 1. The original VGG-16 network consists of 13 convolutional layers, five pooling layers, and three fully connected layers. Compared with the AlexNet network,(27) the VGG-16 network has a simple structure and possesses fewer hyperparameters, and its convolutional layers all use the same convolutional kernel parameters.

In the VGG-16 network, the convolutional kernel size is 3 × 3, so finer detailed features can be obtained. Also, the pooling layer of the VGG-16 network uses a maximum pooling kernel with a size of 2 × 2, so better results can be achieved when capturing local information of features, such as image edges and texture.

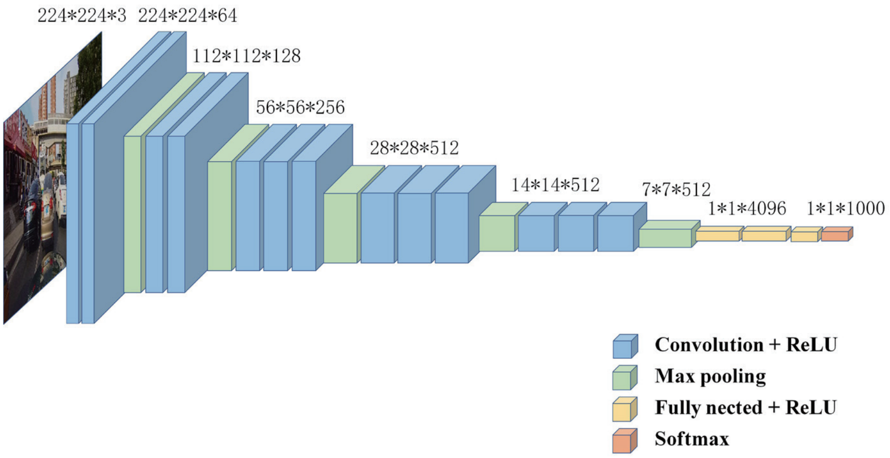


Fig. 1. (Color online) VGG-16 network structure.

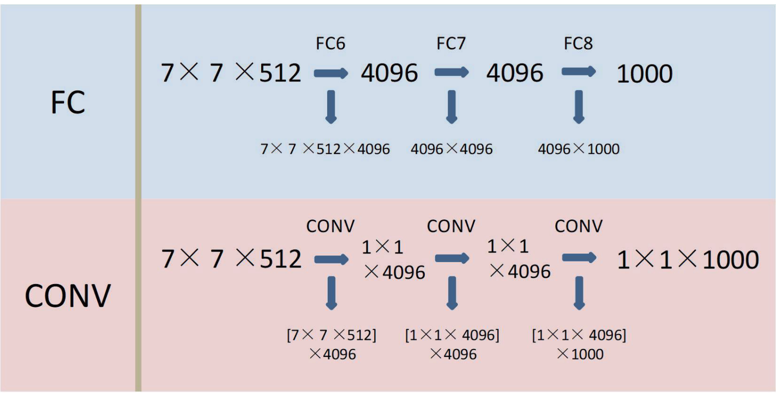


Fig. 2. (Color online) Replacement of fully connected layers with convolutional layers in VGG-16 network.

The last three fully connected layers in the VGG-16 network structure contain a large number of parameters, which greatly affect the computational efficiency of the network. In addition, the input images must be of a specific size, which does not facilitate subsequent image input work. To solve these two problems, all three fully connected layers in the VGG-16 network are replaced with convolutional layers, as shown in Fig. 2.

2.2 Dilated convolution (DC) and jump structure

The addition of a pooling layer in the convolutional network will result in the loss of feature information in the training process, which decreases accuracy. However, if the pooling layer is removed and the convolutional kernel is expanded, the training accuracy will increase. Therefore, in this work, the DC is used.[28] The schematic diagrams of the void convolution are shown in Fig. 3, where red dots denote the convolutional kernel, and the light green color represents the perceptual field in the original input.

In Fig. 3(a), a 3 × 3 convolution with a dilation rate of one is presented, where the receptive field is 3 × 3. When the dilation rate is two, as shown in Fig. 3(b), although the number of convolutional nuclei is fixed, the receptive field increases to 7 × 7. If the dilation rate continues to increase, then in a scenario where the number of convolutional nuclei is fixed, the receptive field can increase to 15 × 15 under the dilation rate of four, as shown in Fig. 3(c). Thus, the convolutional kernel receptive field grows exponentially with the dilation rate.

2.3 SCNN model

To address the problem that, in complex road scenarios, the number of lane line pixels in an image is much less than that of background pixels, which leads to inefficient feature information transfer between alternate convolutional layers, the SAD method has been proposed in Ref. 29. This method allows enhancing the model's performance without increasing its training time, which represents a self-representation learning ability. Similar to the self-attention mechanism,[30] the SAD allows a network to use the attention map of its own layer as a learning

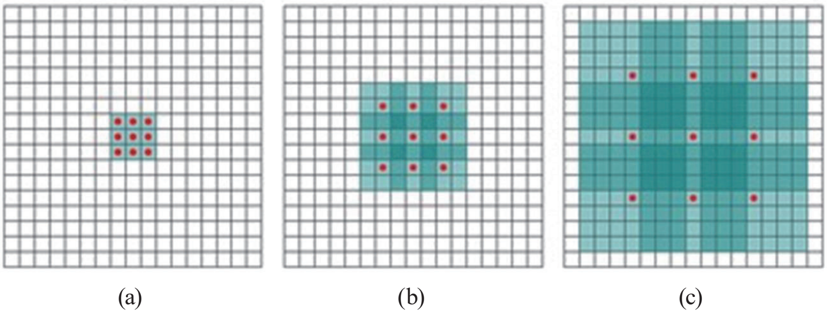


Fig. 3. (Color online) Schematic diagram of void convolution. (a) 1-DC, (b) 2-DC, and (c) 4-DC.

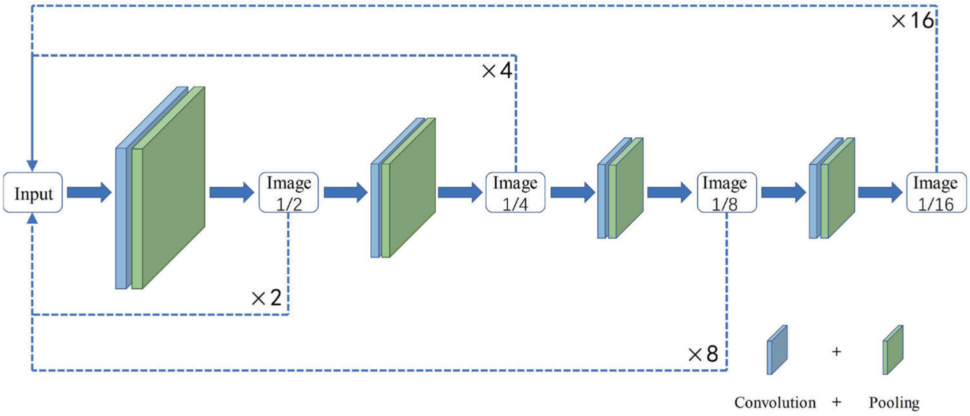


Fig. 4. (Color online) Jump connection. A brief demonstration of how jump connection is structured and how it works.

target for its lower layers, and this attention detection mechanism has often been used to complement segmentation-based supervised learning, as shown in Fig. 4.

In actual road scenes, complex scenarios with broken and discontinuous lane lines often occur, causing the disadvantage of low efficiency of feature information transfer during the network training process, which can be compensated for by the SAD. The SAD has been mainly added during the network training process to provide information for deeper feature maps by learning the low-level feature maps so that the network can obtain richer contextual information. Therefore, the VGG-16 model adds the SAD model after the 13th convolutional layer located between the encoder and the decoder so that the spatial information can also be better transferred.

Although the improved VGG-16 model has a powerful feature detection capability, lane line detection can still be difficult when there are continuous-shaped targets with long distances. To improve the efficiency and accuracy of lane line detection, a new method has been proposed in Ref. 31; this method uses a SCNN by fully mining the spatial relationships of rows and columns in an image to obtain semantic information on targets with strong spatial relationships but weak shape coherence in appearance, such as obscured or even missing lane lines. The structure of this network is shown in Fig. 5.

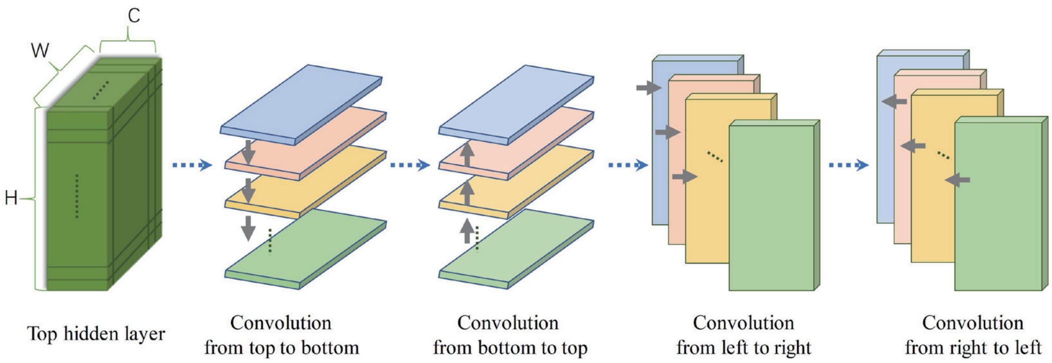

C
W
H
Top hidden layer
Convolution from top to bottom
Convolution from bottom to top
Convolution from left to right
Convolution from right to left

Fig. 5. (Color online) Structure of spatial convolution model.

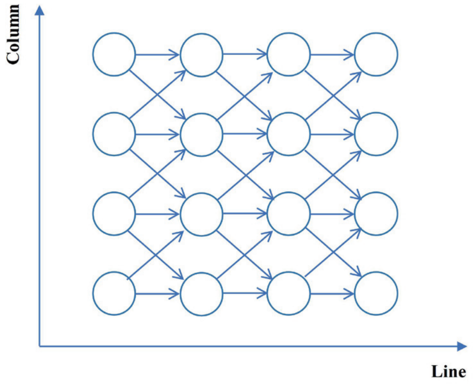


Fig. 6. (Color online) Information transfer chart.

Compared with the traditional networks in which feature pixel information is passed from all directions, thus causing a data redundancy problem, the SCNN model passes information in a sequential way, as shown in Fig. 6. In the SCNN model, each pixel is passed to the next layer in rows or columns, thus saving much computation time and increasing the computation efficiency. Therefore, the SCNN model can be easily integrated into any part of a network model.

2.4 Proposed network model

The network used in this study is based on the VGG-16 model, and additional models are introduced, such as expanded convolution and self-attentive distillation, to propose the VGG-SS network. The VGG-SS network has 16 convolutional layers and five pooling layers, all using ReLU as the activation function. The encoder mainly parses and classifies the lane line pixels. In the decoder part, the number of deconvolution layers corresponds to that of convolution layers in the encoder part, i.e., the decoder's deconvolution is expanded into five stages, with each stage consisting of a deconvolutional layer and a full convolutional layer. The feature map output by the encoder is passed to the decoder according to the characteristics of the jump structure to strengthen the detailed information lost in the upsampling process. The VGG-SS structure is shown in Fig. 7.

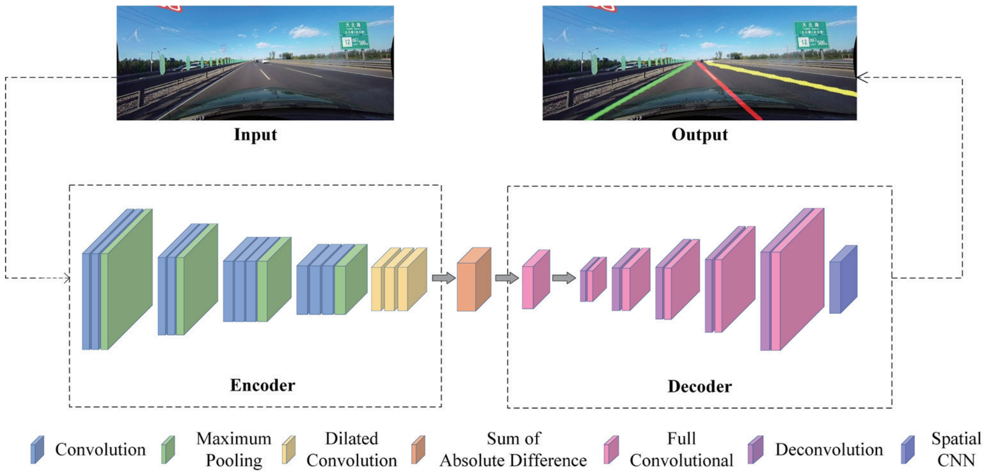


Fig. 7. (Color online) Structure of proposed VGG-SS network model.

3. Experiments and Analysis of Results

3.1 Dataset and parameter settings

The CULane dataset containing a total of 133235 images was used in the experiments. The parameters of the CULane dataset are given in Table 1. The CULane dataset includes data from nine scenes, including common, crowded, and night scenes. The proportion of each of the scenes is shown in Fig. 8.

The experiments were conducted on a computer with the NVIDIA 1080Ti graphics processor, 8 GB video memory, and Windows 10 operating system. The network was developed and trained using the Python3 language and TensorFlow deep learning framework platform under Windows 10.

During the VGG-SS network training, Adam’s gradient optimization strategy was used to decrease the training time and improve the convergence speed. The training parameters of the VGG-SS network are given in Table 2.

In this study, we use the cross-entropy loss function (Cross Entropy) based on binary classification to distinguish lane lines and backgrounds, and we set the distance *d*. When the distance between two different categories of pixel sets is greater than the threshold *d*, the model will not be updated. In this study, the initial learning rate of training is 0.001 and the weight decay factor is 0.0001. Since the memory capacity of the GPU used in this study is only 8 GB, the batch size of the dataset is set to 10. In other words, 10 images are input into the network for training each time, and the polynomial learning rate decay strategy is used for network training.

3.2 Detection accuracy evaluation indexes

Different datasets have different parameters and pixel sizes owing to differences in vehicle type and acquisition equipment, so different accuracy evaluation indexes should be used for different datasets.

Table 1
Dataset parameters

Dataset	Total number of images	Training set	Validation set	Test set	Pixels	Environment	Source	Evaluation indicator
TuSimple	6408	3268	358	2782	1280 × 720	Expressway	Santiago	Accuracy
CULane	133235	88880	9675	34680	1640 × 590	Urban Expressway rural	Beijing	F1Score/IOU

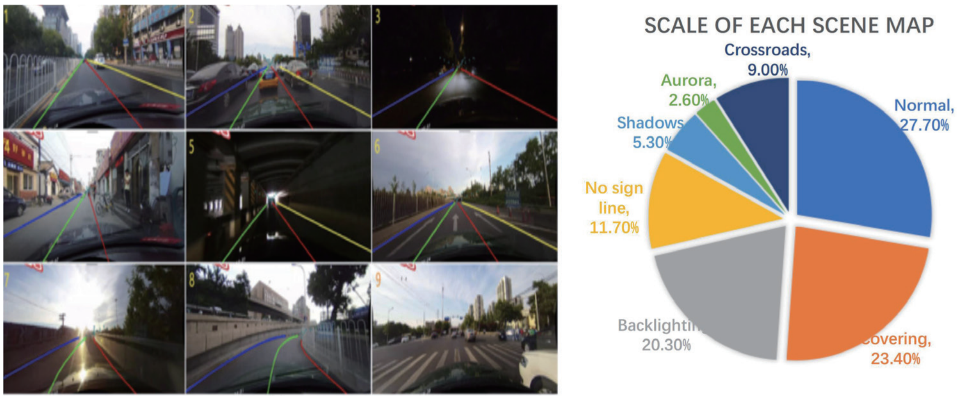


Fig. 8. (Color online) Sampling map of the dataset and scale map of each scene.

Table 2
Training parameters of VGG-SS network.

Parameter	Value
Dataset	CULane
Input image size	800 × 290 pixels
Initial learning rate	0.001
Power	0.9
Batch size	10
Weight decay	0.0001
Maximum iteration number	60000

To evaluate the lane line detection accuracy of the proposed model, each lane line was regarded as a line with a width of 30 pixels. In the evaluation process, Intersection over Union (IOU) between the predicted result and the true value was calculated. The IOU represents the ratio between the intersection part and the merge part of the predicted result and the true value. The threshold value was set to 0.6, and when IOU $\geq$ 0.6, the detection was adjudged correct. When the set threshold value was exceeded, the detection was judged to be correct and regarded as true positive (TP); otherwise, the detection was regarded as false positive (FP).

The $F1$ score is a statistical measure of the accuracy of a binary classification model, which takes into account the precision and recall of the classification model. The precision, recall, and $F1$ measure are respectively expressed as

$$Precision = \frac{TP}{TP + FP}, \tag{1}$$

$$Recall = \frac{TP}{TP + FN}, \tag{2}$$

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}, \tag{3}$$

where *TP* stands for true positive (prediction is positive and the actual value is also positive); *TN* stands for true negative (prediction is negative and the actual value is also negative); *FP* stands for false positive (prediction is positive, but the actual value is negative); and *FN* stands for false negative (prediction is negative, but the actual value is positive). The closer the precision is to the recall, the larger the *F*1 value will be, and a larger *F*1 value indicates a higher precision.

3.3 Experimental and comparison results

(1) DC-VGG-SAD model detection results

The test set selected included a total of 5000 images of five road types in the CULane dataset, namely, normal road scenes, congested scenes, scenes with blocked lane lines, scenes without light at night, and scenes without painted lane lines; 1000 images were selected for each road type for testing.

The DC-VGG-SAD convolutional network was compared with the existing high-quality networks by adding the DC and SAD models in terms of detection accuracy. The experimental results are shown in Fig. 9, where it can be seen that under road conditions with fewer vehicles and more vehicles, the detection accuracy of the DC-VGG-SAD network was slightly higher

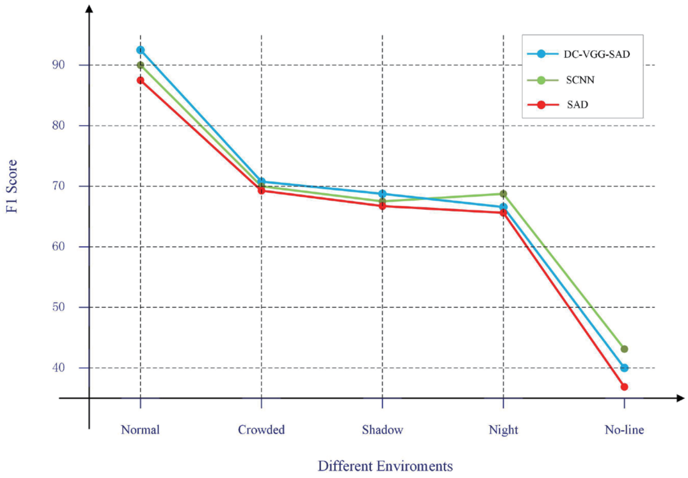


Fig. 9. (Color online) Comparison of F1 scores of different network models on CULane dataset.

than those of the other networks; compared with that of the single SAD network, the detection capability of the DC-VGG-SAD network was improved, indicating its advantages in lane line detection. However, at night and in road environments without lane lines, when completely obscured, the lane lines cannot be effectively identified, and the detection accuracy of the DC-VGG-SAD network was lower than that of the SCNN network. However, when partially obscured, the lane lines can be partially identified effectively, and therefore, the lane lines can be extracted and predicted.

(2) VGG-SS model detection results

In the experiment, three road environments were selected for the test set, namely, the normal road environment, the congested road environment, and the road environment with blurred lane lines under shaded light. As shown in Fig. 10, the trained network model was used to detect the lane lines in the first row of the original image data separately. The results showed that the VGG-16 network detection performed well in the normal road environment, but when the conditions of the road environment changed from simple to complex, especially when the road was obscured by vehicles, the VGG-16 network performance decreased, the VGG-16 network could not accurately detect the lane line pixels, and the detection results will appear to identify only part of the lane line, not all the lane lines. The detection effect was significantly improved when the expanded convolution and SAD models were added to the VGG-16 network, which avoided the incomplete detection of lane lines due to occlusion and lane line mutilation, but there was obvious unsmoothness in the lane line edge area.

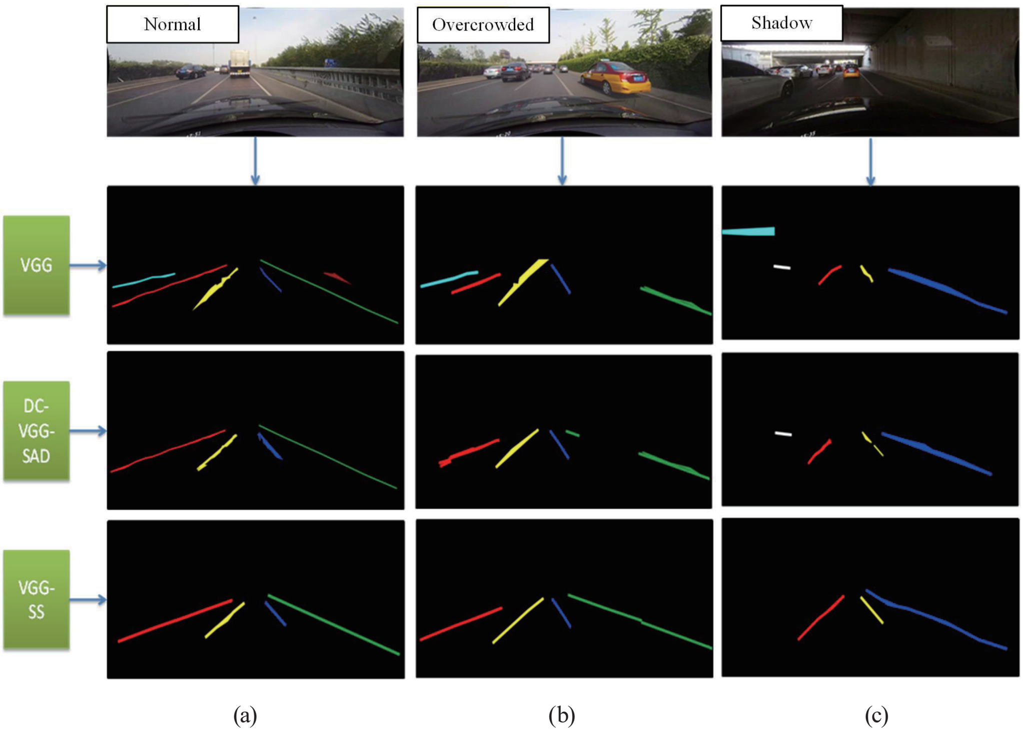


Fig. 10. (Color online) Performance of convolutional network in lane line detection under different scenarios: (a) normal, (b) crowded, and (c) shadow.

(3) VGG-SS lane line detection results

The lane line detection results of the proposed VGG-SS network on the CULane dataset are shown in Figs. 11–15, where (a) indicates the original figure and (b) indicates the detection result of the lane line. As shown by the purple dashed circle in Fig. 11, the part that did not belong to the lane line was not detected, which ensured low false detection and leakage rates, and also guaranteed the lane line detection accuracy.

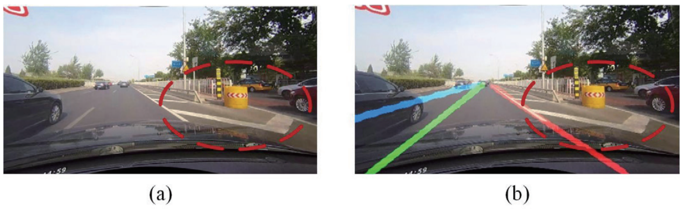


Fig. 11. (Color online) VGG-SS network lane line detection results.

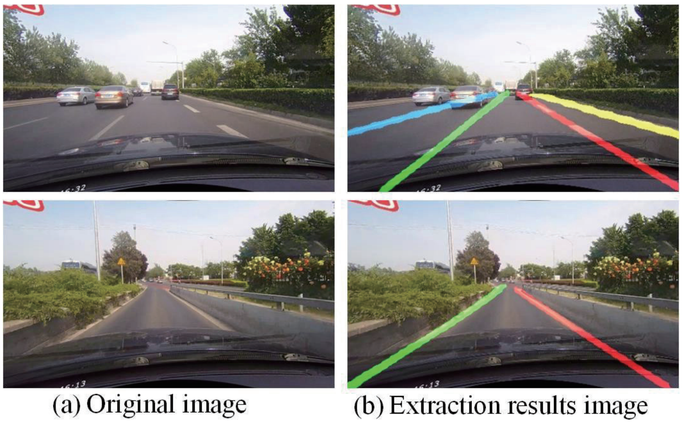


Fig. 12. (Color online) Lane line detection results of VGG-SS network for normal road scenarios.

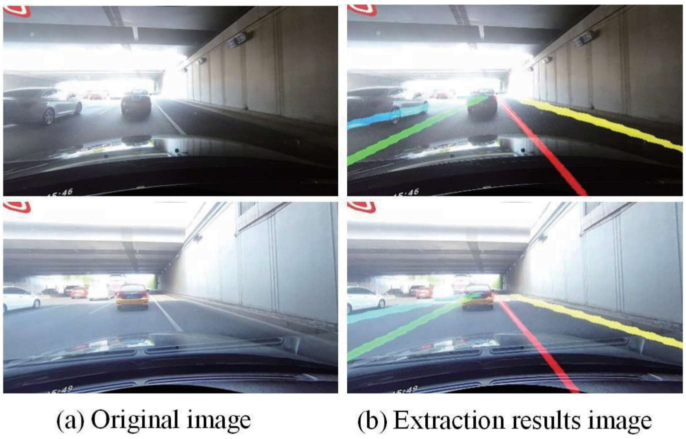
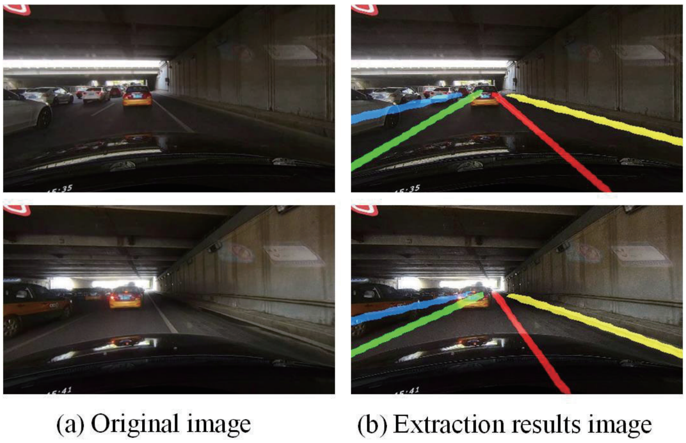
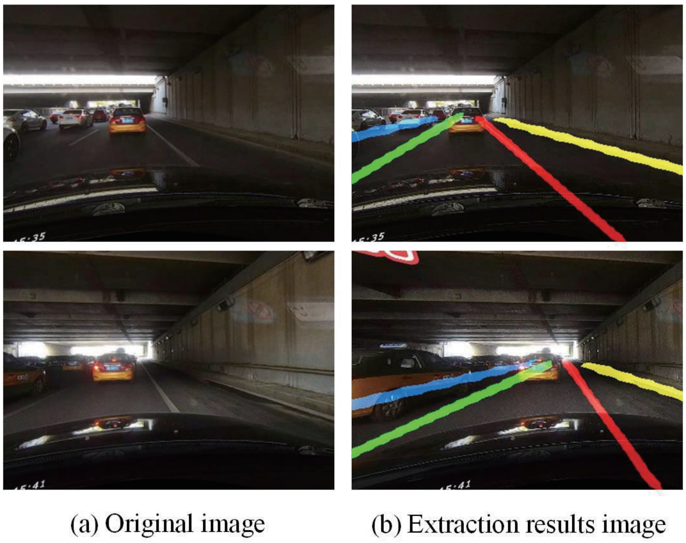


Fig. 13. (Color online) Lane line detection results of VGG-SS network for backlit scenes in road images.



(a) Original image (b) Extraction results image

Fig. 14. (Color online) Lane line detection results of VGG-SS network for shaded scenes in road images.



(a) Original image (b) Extraction results image

Fig. 15. (Color online) Lane line detection results of VGG-SS network for obscured road scenarios.

(4) Comparative experiments

The improved network VGG-SS was compared with the existing VGG-16, SAD, and SCNN network models. The test dataset used in the comparison experiment was the same as that used in the previous experiments, and the obtained detection results are given in Table 3.

As shown in Table 3, the $F1$ score of the proposed VGG-SS network on the CULane dataset was much higher than those of the existing networks, such as VGG-16 and SAD. In simple road scenarios, the $F1$ score of the VGG-SS was 94.7, which was higher than the $F1$ score of the SCNN with a strong spatial information transfer capability. However, in complex road scenarios, when the lane lines were broken or blocked, the detection accuracy of the VGG-SS network decreased significantly, but it was still higher than those of the VGG-16, SAD, and SCNN neural models. The comparison results prove that the proposed method can improve the detection accuracy of lane lines in complex road scenarios.

Table 3
Comparison of accuracy and operating speed of different network models.

Network model	$F1$ score (simple scenes)	$F1$ score (complex scenes)	Running time (ms)
VGG-16	80.8	61.4	20.7
SAD	87.3	64.1	13.2
SCNN	91.6	76.6	134.2
DC-VGG-SAD (ours)	91.7	66.7	17.7
VGG-SS (ours)	94.7	82.6	147.5

4. Conclusion

To solve the problem of low detection accuracy of the existing lane line detection methods in complex road scenes, where lane lines are often damaged and obscured, in this study, we propose a lane line detection method based on improved semantic segmentation. The SAD and SCNN models are used to optimize the encoder-decoder network structure based on the improved VGG-16. The proposed VGG-SS model is trained on the CULane dataset and then compared with state-of-the-art semantic segmentation network models. Experimental results show that the lane line detection accuracy of the VGG-SS model is average when the lane lines are obscured and shaded, and higher when there are fewer road vehicles and the lane lines are clearly visible, but its detection accuracy can still reach 82.6%, which is significantly higher than those of the other semantic segmentation models. This proves that the proposed method can improve the detection accuracy of lane lines in both simple and complex road scenarios.

The fully self-driving car system should not only detect and extract information related to lane lines, but also provide insights into the detection of various lane signs, such as steering, speed limit, and crosswalks. In this paper, the VGG-SS network is constructed on the basis of the VGG-16 network, and several models are introduced and modified. The lane line detection in complex road conditions achieves good results. However, this experiment is only applied to pictures, and lane line extraction on video and moving images is still something that should be studied in the future. Subsequent work can combine convolutional neural networks with different tasks to form a complete multitask neural network, providing important technical support for self-driving cars.

Supporting Data

The data that support the findings of this study are available in Github at https://github.com/Mcwbiubiubiu/Lane-line-detection.git. These data were derived from the following resources available in the public domain: http://mmlab.ie.cuhk.edu.hk/

References

1 A. B. Hillel, R. Lerner, D. Levi, and G. Raz: Mach. Vision Appl. **25** (2014) 727. https://doi.org/10.1007/s00138-011-0404-2

2 H. Wei, Y. Chen, and Z. Peng: Proc. Institution of Mechanical Engineers, Part D: J. Autom. Eng. **4** (2019) 765. https://doi.org/10.1177/0954407017752179

3 R. Tapia-Espinoza and M. Torres-Torriti: Sensors **3** (2013) 3270. https://doi.org/10.3390/s130303270

4 J. Fritsch, T. Kuehnl, and A. Geiger: 16th Int IEEE Conf. Intell. Transp. Syst. (ITSC, IEEE, 2013) 1693. https://doi.org/10.1109/ITSC.2013.6728473

5 W. Yue, D. Shen, and E. K. Teoh: Pattern Recognit. Lett. **8** (2000) 677. https://doi.org/10.1016/S0167-8655(00)00021-0

6 W. Yue, E. K. Teoh, and D. Shen: Lane Image Vision Comput. **22** (2004) 269. https://doi.org/10.1016/j.imavis.2003.10.003

7 H. Yoo, U. Yang, and K. Sohn: IEEE Trans. Intell. Transp. Syst. **3** (2013) 1083. https://doi.org/10.1109/TITS.2013.2252427

8 H. Lin, S. Ko, S. Wang, Y. Kim, and H. Kim: IEEE 2007 Int. Conf. Control, Automation and Systems (2007). https://doi.org/10.1109/ICCAS.2007.4407061

9 V. Gaikwad and S. Lokhande: IEEE Trans. Intell. Transp. Syst. **2** (2015) 910. https://doi.org/10.1109/TITS.2014.2347400

10 A. Mammeri, A. Boukerche, and Z. Tang: Comput. Commun. **73** (2015) 229. https://doi.org/10.1016/j.comcom.2015.08.010

11 M. N. Sotelo, F. J. Rodríguez, L. Magdalena, L. M. Bergasa, and L. Boquete: Auton. Robot. **1** (2004) 95. https://doi.org/10.1023/B:AURO.0000008673.96984.28

12 U. Ozgunalp and N. Dahnoun: 2015 IEEE Global Conf. Signal and Information Process (GlobalSIP, IEEE, 2015). https://doi.org/10.1109/GlobalSIP.2015.7418332

13 A. M. Kumar, P. Simon, and R. Kavitha: Artif. Intell. Evol. Comput. Eng. Syst. (Springer, Singap, 2017). https://doi.org/10.1007/978-981-10-3174-8_5

14 F. H. Chi, C. L. Huo, Y. H. Yu, and T. Y. Sun: 2012 IEEE Int. Conf. Fuzzy Syst. (2012) 1. https://doi.org/10.1109/FUZZ-IEEE.2012.6250840

15 J. Piao and H. Shin: IET Image Process **12** (2017) 1210. https://doi.org/10.1049/iet-ipr.2016.0506

16 Y. Xing, C. Lv, L. Chen, H. Wang, H. Wang, D. Cao, E. Velenis, and F.-Y. Wang: IEEE/CAA J. Autom. Sinica **3** (2018) 645. https://doi.org/10.1109/JAS.2018.7511063

17 S. Lee, J. Kim, J. S. Yoon, S. Shin, O. Bailo, N. Kim, T.-H. Lee, H. S. Hong, S.-H. Han, and I. S. Kweon: 2017 IEEE Int. Conf. Comput. Vision (ICCV, 2017) 1965. https://doi.org/10.1109/ICCV.2017.215

18 Y. Y. Ye, X. L. Hao, and H. J. Chen: IET Intell. Transp. Syst. **6** (2018) 513. https://doi.org/10.1049/iet-its.2017.0143

19 K. He, X. Zhang, S. Ren, and J. Sun: IEEE Trans. Pattern Anal. Mach. Intell. **9** (2014) 1904. https://doi.org/10.1007/978-3-319-10578-9_23

20 S. Ren, K. He, R. Girshick, and J. Sun: IEEE Trans. Pattern Anal. Mach. Intell. **6** (2017) 1137. https://doi.org/10.1109/TPAMI.2016.2577031

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

22 M. Aly: IEEE Intell. Vehicles Symp. (2014). https://doi.org/10.1109/IVS.2008.4621152

23 J. Kim and C. Park: 2017 IEEE Conf. Comp. Vision and Pattern Recognit. Workshops (CVPRW, 2017) 1194. https://doi.org/10.1109/CVPRW.2017.158

24 D. Neven, B. De Brabandere, S. Georgoulis, M. Proesmans, and L. van Gool: 2018 IEEE Intell. Vehicles Symp. (IV) (2018) 286. https://doi.org/10.1109/IVS.2018.8500547

25 O. Ronneberger, P. Fischer, and T. Brox: Int. Conf. Med. Image Comput. and Comput.-Assisted Intervention (2015). https://doi.org/10.1007/978-3-319-24574-4_28

26 K. Simonyan and A. Zisserman: Comput. Sci. (2014). https://arxiv.org/abs/1409.1556

27 A. Krizhevsky, I. Sutskever, and G. E. Hinton: Adv. Neural Process Syst. (2012) 1097. https://doi.org/10.1145/3065386

28 F. Yu and V. Koltun: ICLR (2016). https://arxiv.org/abs/1511.07122

29 Y. Hou, Z. Ma, C. Liu, and C. C. Loy: ICCV (2019). https://arxiv.org/abs/1908.00821

30 A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin: Computation and Language (2017). https://arxiv.org/abs/1706.03762v1

31 X. Pan, J. Shi, P. Luo, X. Wang, and X. Tang: AAAI (2018). https://arxiv.org/abs/1712.06080v1

About the Authors

Chaowei Ma received his B.S. degree from Beijing University of Civil Engineering and Architecture, China, in 2020. 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.

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.

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.