Hindawi
Mathematical Problems in Engineering
Volume 2022, Article ID 7919875, 14 pages
https://doi.org/10.1155/2022/7919875

Research Article

A Robust Detection Method for Multilane Lines in Complex Traffic Scenes

Xiang Song,[1] **Xiaoyu Che**,[2] **Huilin Jiang**,[1] **Shun Yan**,[1] **Ling Li**,[1] **Chunxiao Ren**,[3] **and Hai Wang**[4]

[1] *School of Electronic Engineering, Nanjing Xiaozhuang University, Nanjing 211171, China*
[2] *National Engineering Research Center of Road Maintenance Technologies, Beijing 100095, China*
[3] *Research Institute of Highway Ministry of Transport, Beijing 100088, China*
[4] *School of Automotive and Traffic Engineering, Jiangsu University, Zhenjiang 212013, China*

Correspondence should be addressed to Hai Wang; wanghai1019@163.com

Received 31 January 2022; Accepted 19 February 2022; Published 8 March 2022

Academic Editor: Xinyuan Jiang

The robustness and stability of lane detection is vital for advanced driver assistance vehicle technology and even autonomous driving technology. To meet the challenges of real-time lane detection in complex traffic scenes, a simple but robust multilane detection method is proposed in this paper. The proposed method breaks down the lane detection task into two stages, that is, lane line detection algorithm based on instance segmentation and lane modeling algorithm based on adaptive perspective transform. Firstly, the lane line detection algorithm based on instance segmentation is decomposed into two tasks, and a multitask network based on MobileNet is designed. This algorithm includes two parts: lane line semantic segmentation branch and lane line Id embedding branch. The lane line semantic segmentation branch is mainly used to obtain the segmentation results of lane pixels and reconstruct the lane line binary image. The lane line Id embedding branch mainly determines which pixels belong to the same lane line, thereby classifying different lane lines into different categories and then clustering these different categories. Secondly, the adaptive perspective transformation model is adopted. In this model, the motion information is used to accurately convert the original image into a bird's-eye view image, and then the least-squares second-order polynomial fitting is performed on the lane line pixels. Finally, experiments on the CULane dataset show that the proposed method achieved similar or better performance compared with several state-of-the-art methods, the F_1 score of the proposed method in the normal test set and most challenge test sets is better than other algorithms, which verifies the effectiveness of the proposed method, and then the field experiments results show that the proposed method has good practical application value in various complex traffic scenes.

1. Introduction

Vehicle and road safety has been a key issue for the communities and governments [1]. With emerging new technologies and knowledge, advanced driver assistance systems (ADAS) have been proposed to reduce road accidents and improve vehicle safety [2]. In ADAS and even autonomous driving vehicles, the main technical bottleneck is the perception problem, which has two elements: road and lane perception and obstacle detection [3].

The robustness and stability of lane detection is vital for advanced driver assistance vehicle technology and even unmanned technology [4]. Firstly, lane detection and tracking aids in localizing the ego-vehicle motion, which is one of the very first and primary steps in most ADAS, such as lane departure warning (LDW) and lane change assistance. Furthermore, lane detection is also able to aid other ADAS modules such as vehicle detection and driver intention perception.

At present, a large number of research results have been achieved in the research of lane detection at home and abroad. The lane detection methods nowadays usually contain two main steps: (1) lane feature extraction and (2) lane modeling [3, 5, 6]. Generally, the lane detection methods can be divided into three categories: (1) traditional lane detection methods, (2) image processing method combined with deep learning, and (3) end-to-end lane line detection method.

The traditional lane line detection method can be divided into three steps [7, 8]. Firstly, the road image data is preprocessed to remove the noise and to obtain the lane line features; then, the lane line is detected from the preprocessed image by means of feature-based or model-based methods; finally, the detection results are fitted to convert the lane lines represented by image coordinates into world coordinates. Aly [9] proposed a real-time and robust model-based detection method for urban road lane lines. IPM is used to convert the front view into a top view, and then the selective directional Gaussian filter is used to filter to check the lines in the image and deal with other abnormal lines. Finally, the lane line is simulated in the front view by mapping. Bounini et al. [10] proposed a feature-based lane line detection method in virtual simulation environment. In the initialization stage, the method integrates Hough transform, Canny edge detection, and Kalman filter to greatly reduce the region of interest and predict the lane line in the future. Overall, the feature-based method has higher computational efficiency but poor robustness. Compared with the feature-based method, the model-based method is more stable, but it is difficult to implement, and the calculation is huge.

The lane line detection method combined with deep learning [11] can extract features from various types of complex environments, which breaks through the limitations of traditional image processing methods. Amayo et al. [12] designed a target classification method based on the extracted lane line geometric elements. Firstly, weakly supervised neural network is used to extract lane line elements under different conditions; then, the global energy optimization method is used to retrieve and cluster the lane geometry, and the corresponding semantic categories are given. Finally, the tracking effect of continuous frame images is optimized to improve the stability of classification effect. Song et al. [13] designed a lane line detection and classification system based on 3D vision. Firstly, the lane line is extracted from the ROI region by the Sobel filter; then, an adaptive lane line model is designed in Hough space to further extract lane lines; finally, the convolution neural network is used to obtain the specific classification results.

Due to the rapid development of deep learning [14–16], researchers prefer to use end-to-end methods to solve computer vision problems [17]. As shown in Figure 1, the initial picture is input into the model, and the desired result can be obtained through the end-to-end method. The method of lane line detection based on end-to-end deep learning technology can increase the accuracy of lane line recognition in a complex environment and simplify the steps of two coordinate system transformation, that is, the transformation between pixel coordinate system and world coordinate system, thus increasing the robustness of lane line detection. Li et al. [18] proposed an end-to-end system called line-CNN to realize the lane line detection method, which uses the straight-line anchor to locate the precise position of the lane line. In this paper, the semantic information of the whole lane line at the global level is considered. Van Gansbeke et al. [19] showed a method to realize lane line detection by solving the weighted least-squares problem, which is specifically divided into two parts. One part is to predict the weight of pixels of each lane line by using the depth convolution network, and the other part is to use the differentiable least-squares fitting module to return the parameters of the best fitting curve of lane line features. Generally, the end-to-end method does not need image preprocessing and manual feature extraction, and the experimental results have great advantages in accuracy and robustness.

Although many novel end-to-end methods have been proposed to deal with lane detection applications and have achieved very good performance, they are usually implemented in high-performance PC or embedded systems so that those methods can use a complicated algorithm and utilize big storage space. However, automotive companies are very sensitive to hardware cost, so they more prefer lane detection method that can be used in low cost and resource-limited platforms. Meanwhile, the complexity and uncertainty of road conditions has led to the fact that there is still room for improvement of existing local methods. The challenges include shadows cast from trees, other vehicles and buildings, invisible or defect lanes, highlight, strange lane shapes, and bad quality lines.

To meet the challenges of real-time lane detection in complex traffic scenes, a simple but robust multilane detection method is proposed for the resource-limited automotive embedded platform in this paper. The proposed method breaks down the lane detection task into two stages, that is, lane line detection algorithm based on instance segmentation and lane modeling algorithm based on adaptive perspective transform model. The main contributions can be listed as follows:

(1) The multilane line detection is considered as an instance segmentation problem in this paper, and a deep learning network is designed to solve the problems of uncertain number of lanes and lane change. The lane detection problem is decomposed into two tasks, that is, lane line semantic segmentation branch and lane line Id embedding branch, and a multitask network based on MobileNet [20] is designed to shorten the forward propagation calculation time without loss of detection accuracy. In MobileNet, a convolution method called depthwise separable convolution is used to replace traditional convolution methods to reduce the network weight parameters. By sharing the first four coding processes between the two tasks, the calculation speed and the accuracy of lane line segmentation are improved. Meanwhile, by fully considering the semantic characteristics of the encoding and decoding process, the strategy of fusing the corresponding encoded information during decoding is adopted to improve the accuracy of lane segmentation.

(2) An adaptive perspective transformation model is designed to overcome the disadvantage that the

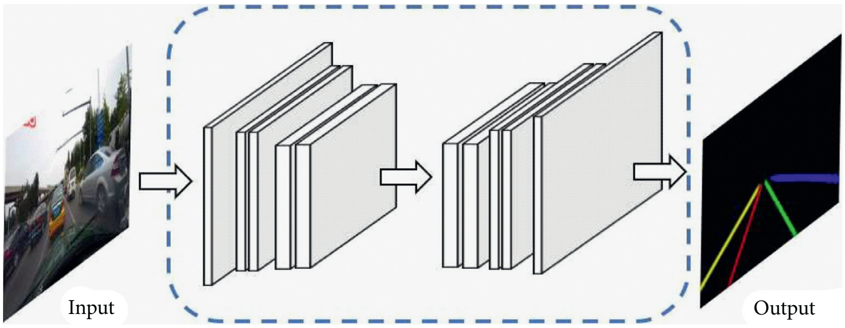


Figure 1: End-to-end lane line detection method.

conventional perspective transformation model is only suitable for smooth roads, and the image will be distorted when the vehicle is driving on the uneven road or the pitch angle of the camera changes due to turbulence. The adaptive perspective transformation model overcomes the deficiency of the transformation matrix with fixed parameters in the traditional model and can realize the accurate conversion from the camera image to the bird's-eye view under the camera motion state so as to improve the robustness of lane line fitting.

The remainder of the paper proceeds as follows. The lane detection method is given in Section 2. Section 3 presents the lane line detection algorithm based on instance segmentation. Section 4 is devoted to the adaptive perspective transform model. Section 5 shows the experimental results and analysis. Section 6 gives the conclusion.

2. Overview of the Proposed Multilane Line Detection Method

In this paper, the multilane line detection is considered as an instance segmentation problem. Each lane line is formed into a separate category, and then each lane line is fitted. In order to increase the running speed, improve the detection accuracy, and meet the requirements of real vehicle applications, the lane detection problem is decomposed into two tasks, and a multitask network based on MobileNet is designed for lane line segmentation. The multitask network includes two branches: lane line semantic segmentation branch and lane line Id embedding branch. The lane line semantic segmentation branch is mainly used to obtain the segmentation results of lane pixels and reconstruct the lane line binary image. The lane line Id embedding branch mainly determines which pixels belong to the same lane line, thereby classifying different lane lines into different categories and then clustering these different categories.

After completing the lane line instance segmentation, these pixels belonging to the lane line need to be parametric curve fitting. Curve fitting refers to the process of finding an appropriate function that fits a finite set of data points. The commonly used curve fitting models include Bayesian fitting [21], B-spline curve fitting [22], and least-square curve fitting [23]. Generally, in order to improve the accuracy of fitting, the fixed perspective transformation model [9, 24] is commonly used to convert the image into a bird's-eye view for curve fitting. However, the algorithm is only suitable for smooth roads. When the road is uneven or the vehicle is bumpy, the image will move and be distorted, resulting in large errors in the conversion into a bird's-eye view. To solve this problem, the adaptive perspective transformation model is designed. In this model, the motion information is used to accurately convert the original image into a bird's-eye view image, and then the least-squares second-order polynomial fitting algorithm is performed on the lane line pixels. Compared with the traditional fixed perspective transformation, this model has better robustness. The specific process of lane line detection is shown in Figure 2.

3. Lane Line Detection Algorithm Based on Instance Segmentation

In this paper, MobileNet is designed to train an instance segmentation model for lane line detection. Its advantage is that it can solve the problem of uncertain number of lanes and lane change. The instance segmentation is composed of two branches: (1) lane semantic segmentation, which is used to obtain the lane line binary image; (2) lane line Id allocation, which is used to determine the pixels that belong to the same lane line.

To improve the calculation speed and the accuracy of lane line segmentation, the two-branch algorithm, for instance, segmentation, shares the first four coding processes. To ensure the real-time performance of the multilane line detection algorithm, the input image is converted into a 512×256 pixel format for network learning. The encoding-decoding structure of the network is shown in Figure 3.

To meet the cost requirements of in-vehicle computing, MobileNet is adopted as an encoding structure to shorten the forward propagation calculation time without loss of detection accuracy. In the encoding process, the semantic feature information is less, and the lane line position is accurate, while the semantic feature information in the decoding process is richer, and the lane line position information is relatively rough. Therefore, similar to the ENet [25], the corresponding encoded information is fused during the decoding process to improve the accuracy of lane line segmentation.

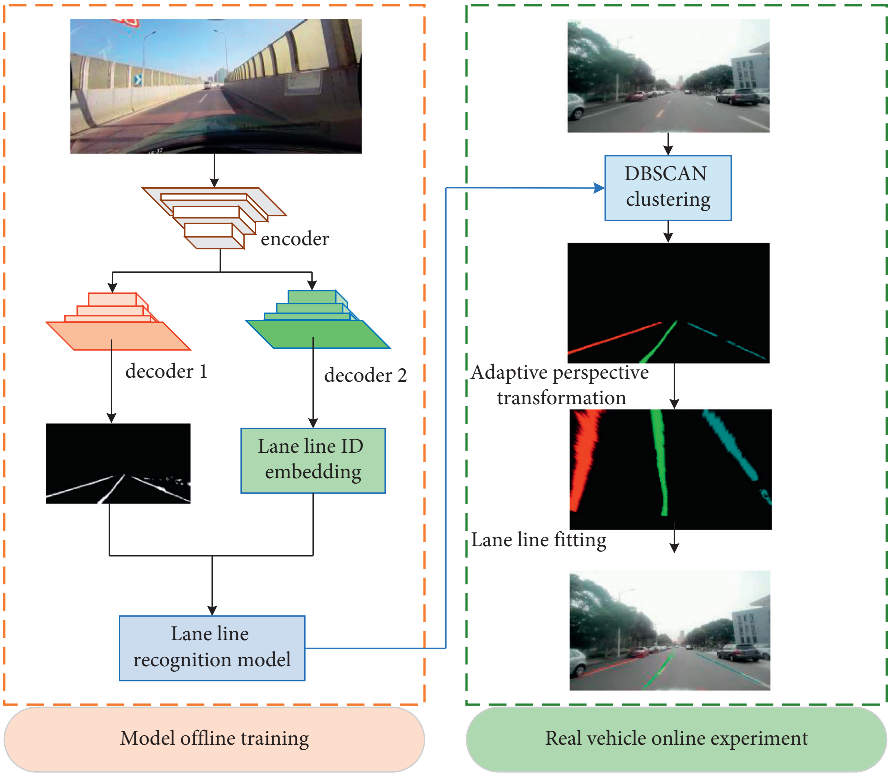


Figure 2: The flowchart of multilane line detection algorithm.

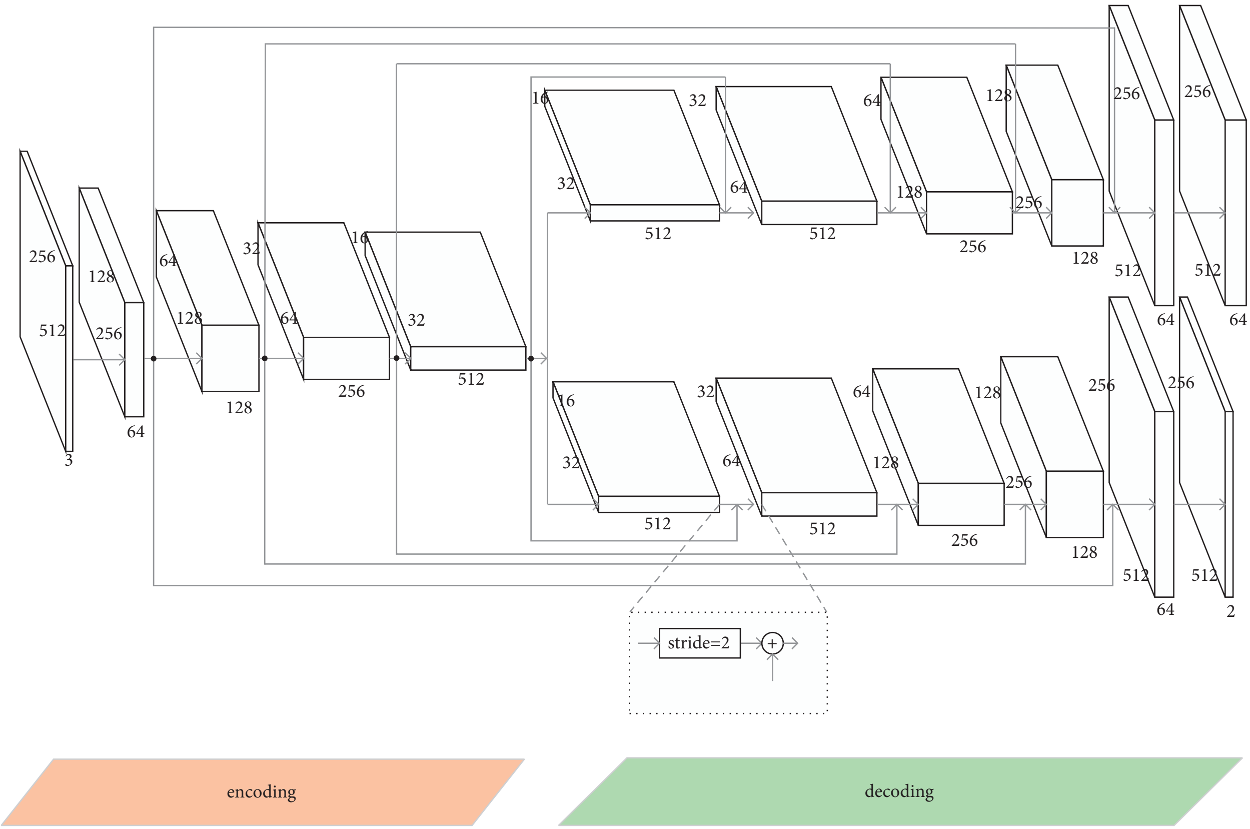


Figure 3: The encoding-decoding structure.

3.1. MobileNet. As a representative of lightweight convolutional networks, the core idea of MobileNet is to use a convolution method called depthwise separable convolution instead of traditional convolution methods to reduce the network weight parameters. The depthwise separable convolutions divide standard convolution into two steps. The first step is depthwise convolution, that is, channel-by-channel convolution. In depthwise convolution, one convolution kernel is only responsible for one channel, and one channel is "filtered" by only one convolution kernel. The second step is pointwise convolution, which is to "string" the results obtained by the first step. Assuming that, in standard convolution, the size of the input feature map is $D_F \times D_F$, the number of input channels is M, the number of output channels is N, and the size of the convolution kernel is $D_K \times D_K$; then the calculation amount of standard convolution is

$$D_F \times D_F \times D_K \times D_K \times M \times N. \tag{1}$$

In MobileNet, the depthwise separable convolution is obtained by the two-part standard convolution solution, and the convolution kernel size is $D_K \times D_K$ and 1×1, respectively. Then, the calculation amount of the depth separable convolution is

$$D_K \times D_K \times M \times D_F \times D_F + M \times N \times D_F \times D_F. \tag{2}$$

As an example, Figure 4 is a schematic diagram of a standard convolution where the input feature map size is 5×5, the number of input channels is 3, the number of output channels is 4, and the size of the convolution kernel is 1×1. Figure 5 is a schematic diagram of the depthwise separable convolution where the input feature map size is 5×5, the number of input channels is 3, the number of output channels is 4, and the size of the convolution kernel is 3×3 and 1×1, respectively.

From (1)and (2), it can be seen that the ratio of the parameter calculation amount of the depthwise separable convolution and the standard convolution under the same input and output feature map size is

$$\frac{D_K \times D_K \times M \times D_F \times D_F + M \times N \times D_F \times D_F}{D_F \times D_F \times D_K \times D_K \times M \times N} = \frac{1}{N} + \frac{1}{D_K^2}. \tag{3}$$

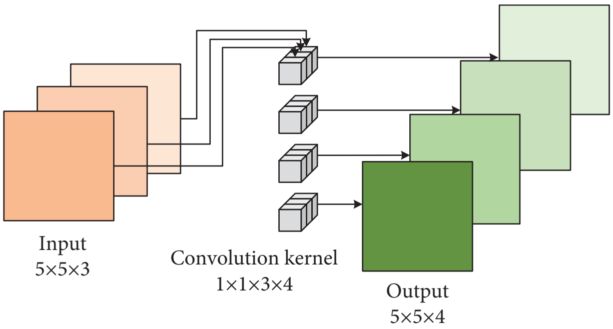


Figure 4: The standard convolution.

In order to compare the performance of convolutional neural networks using standard convolution and depthwise separable convolution, these two networks were used for training and testing on the ImageNet dataset in [20]. The results show that, in terms of accuracy, using standard convolution is 1.1% higher than using depth separable convolution, but in terms of calculation amount and parameter amount, the former is 8–9 times the latter. From [20], we can conclude that the use of deep separable convolution can greatly reduce the amount of calculation and the number of parameters while basically ensuring the accuracy. Correspondingly, it can reduce the difficulty of network model training, reduce training time, and reduce the performance requirements of hardware devices.

3.2. Lane Line Semantic Segmentation Branch. The purpose of lane line semantic segmentation is to obtain the segmentation result of lane pixels and reconstruct the binary image of the lane line, and then determine which pixels belong to the lane line. Due to the extreme imbalance between the lane pixels and the background pixels, the focal loss function [26] is used for model training, and the specific lane line pixel segmentation loss function formula is as follows:

$$L_{\text{seg}} = \begin{cases} \dfrac{-1}{n_c \times h \times w} \displaystyle\sum_{c=1}^{n_c} \sum_{y=1}^{h} \sum_{x=1}^{w} \left[\alpha \left(1 - \hat{s}(c, x, y)\right)^{\gamma} \log \hat{s}(c, x, y)\right] & \text{if } s(c, x, y) = 1, \\ \\ \dfrac{-1}{n_c \times h \times w} \displaystyle\sum_{c=1}^{n_c} \sum_{y=1}^{h} \sum_{x=1}^{w} \left[(1 - \alpha) \hat{s}(c, x, y)^{\gamma}\right] \log \left(1 - \hat{s}(c, x, y)\right) & \text{if } s(c, x, y) = 0, \end{cases} \tag{4}$$

where $s(c, x, y)$ indicates whether the pixel (x, y) belongs to category c, that is, $s(c, x, y) = \begin{cases} 1 & \text{if } (x, y) \in c \\ 0 & \text{if } (x, y) \notin c \end{cases}$; $\hat{s}(c, x, y)$ represents the probability that pixel (x, y) is predicted to be category c; n_c is the number of segmentation categories, $n_c = 2$; h and w are the length and width of the output

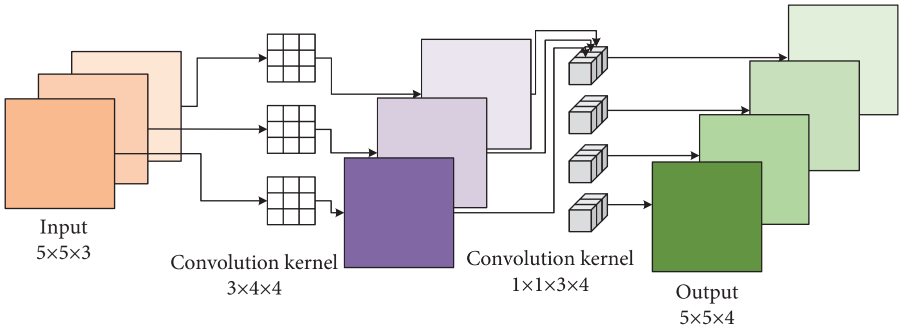


Figure 5: Depthwise separable convolution.

prediction image, respectively; γ represents the focus parameter. To increase the weight of the pixels that are difficult to classify, the weight of the pixels that are easy to classify becomes smaller so that the network can pay more attention to the pixels that are difficult to learn; in this paper, $\gamma = 2$. α refers to the weighting parameter of the lane line category. α is mainly used to solve the problem of the uneven number of pixels in each category. The mathematical calculation formula of α is

$$\alpha = \frac{1}{\ln(1.02 + n/m)}, \tag{5}$$

where n represents the number of lane line pixels; m indicates the total number of pixels.

3.3. Lane Line Id Embedding Branch. The lane line Id embedding branch mainly determines which pixels belong to the same lane line. First, the vector value corresponding to each pixel is output by the encoder. Then the pixels are clustered according to the feature that the vector distances of the pixel points of the same lane line are close, and the vector distances of the pixels of different lane lines are relatively far. Finally, the pixels belonging to the same lane line are determined. The clustering loss function L consists of two parts, L_1 and L_2. L_1 is used to cluster the pixels belonging to the same lane line to the center point according to the embedded pixel point vector value. L_2 is used to move pixels that do not belong to a lane line as far away as possible according to the embedded pixel vector value. There are $N(N-1)/2$ groups of different lane lines, and the clustering loss function L is

$$\begin{aligned}
L_1 &= \frac{1}{N}\sum_{1}^{N}\frac{1}{N_c}\sum_{1}^{N_c}\left[\left(\mu_c - x_i\right) - 0.6\right]^2,\\
L_2 &= \begin{cases} \dfrac{N(N-1)}{2}\displaystyle\sum_{1}^{N(N-1)/2}\left[6-\left(\mu_i-\mu_j\right)\right]^2 N \geq 2, & i \neq j,\\ 0 & N = 1,\end{cases}\\
L &= L_1 + L_2,
\end{aligned} \tag{6}$$

where N is the number of lane lines; N_c is the number of pixels of a lane line; μ_c is the mean vector value embedded in pixels of a certain lane line; x_i is the embedding vector of a certain pixel; μ_i and μ_j are the mean vectors embedded in the pixels of two different lane lines i and j.

In the inference stage, the DBSCAN clustering algorithm [27] is used to cluster each pixel. The DBSCAN method is a clustering algorithm based on high-density connected regions, which can divide regions with sufficiently high density into clusters and can find clusters of any shape in noisy data. The DBN method is used in this paper for clustering until all lane line pixels are assigned to the corresponding lanes. The cluster center is taken as the center of the circle, the radius is 0.28 m, and the minimum number of points in the domain is 180.

4. Adaptive Perspective Transform Model

After obtaining the clustered pixel point set of each lane line based on the lane line instance segmentation, the pixel points should be fitted. According to the imaging principle of the camera, the lane line will gradually converge into a point at a long distance, so the accuracy of the lane line fitting directly on the original image will be greatly reduced. In the commonly used improvement method, the picture is converted into a bird's-eye view through a perspective transformation model to make the lane lines parallel so that the fitting accuracy is improved. However, in the conventional perspective transformation model, the transformation matrix with fixed parameters is adopted, which is only suitable for smooth road conditions. If the vehicle is driving on an uneven road or the pitch angle of the camera changes due to turbulence, the image will be distorted, which will affect the generated bird's-eye view. Therefore, an adaptive perspective transformation model that can realize the accurate conversion of the camera image to the bird's-eye view under the camera motion state is adopted.

As shown in Figure 6, the origin of the world coordinate system is defined as the point where the car's center of mass is perpendicular to the ground, and the vertical distance between the camera installation position and the origin of the world coordinate system is h. According to the relationship between the world coordinate system $(O - \widehat{X}\widehat{Y}\widehat{Z})$, the camera coordinate system $(O_2 - \widehat{cr})$, and the image pixel

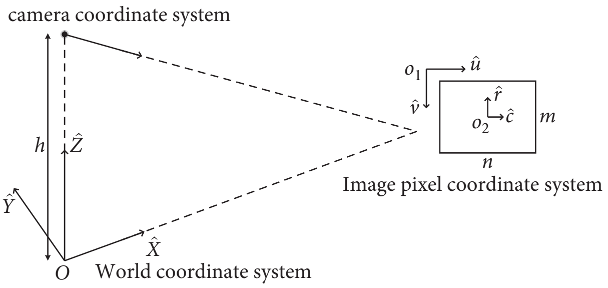


Figure 6: Relationship between world coordinate system, camera coordinate system, and image pixel coordinate system.

coordinate system ($O_1 - \hat{u}\hat{v}$), the position of the pixel in the camera coordinate system can be expressed as

$$c(u) = K\left(u - \frac{n}{2}\right),$$
$$r(v) = K\left(\frac{m}{2} - v\right), \tag{7}$$

where K is the conversion scale factor between the camera coordinate system and the image pixel coordinate system; m is the width of the image; n is the length of the image.

In order to better explain the parameters of the adaptive perspective transformation model, the lateral view structure of the adaptive perspective transformation model is established, as shown in Figure 7. The x-axis in the world coordinate system can be written as a function of pixel points v, θ_0, and θ_b:

$$X(v,b) = h\cot\left(\theta_0 + \theta_b + \theta(v)\right), \tag{8}$$

$$\theta(v) = \arctan\left(\frac{r(v)}{f_r}\right), \tag{9}$$

$$f_r = r(v)\cot\theta(v). \tag{10}$$

When $v = m$, $\theta(v)$ is α. Substituting formula (7) into formula (10), f_r can be obtained:

$$f_r = \frac{K(2-m)}{2}\cot\alpha. \tag{11}$$

Substituting formulas (9) and (11) into (8),

$$X(v,b) = h\frac{\tan\left(\theta_0 + \theta_b\right)(m - 2v/2 - m)\tan\alpha - 1}{\tan\left(\theta_0 + \theta_b\right) + m - 2v/2 - m\tan\alpha}, \tag{12}$$

where θ_0 is the tilt angle of the camera; θ_b is the change value of the pitch angle when the camera bumps; $\theta(v)$ is the angle value between the ground pixel and the camera; α is the half of the vertical field of view of the camera; f_r is the vertical focal length of the camera.

As shown in Figure 8, the vertical view structure of the adaptive perspective transformation model is established, and the $Y(u,v,b)$ axis in the world coordinate system can be expressed as

$$\frac{Y(u,v,b)}{X(v,b)} = -\frac{c(u)}{f_h}. \tag{13}$$

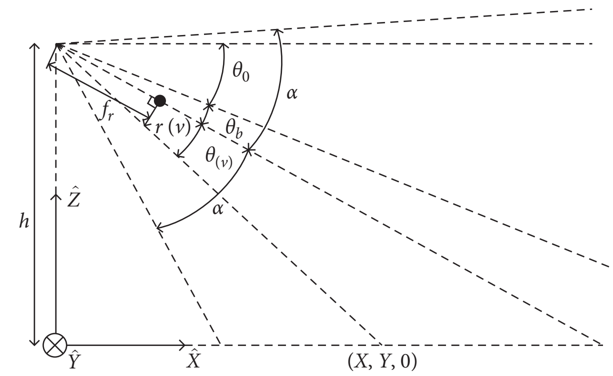


Figure 7: The lateral view structure of the adaptive perspective transformation model.

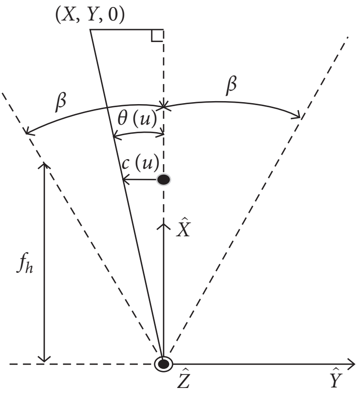


Figure 8: The vertical view structure of the adaptive perspective transformation model.

When $u = n$, $\theta(u)$ is β. f_h can be obtained from formula (13):

$$f_h = \frac{c(u=n)}{\tan\beta} = \frac{Kn}{2\tan\beta}. \tag{14}$$

Substituting formulas (12) and (14) into (13),

$$Y(u,v,b) = h\left(1 - \frac{2u}{n}\right)\tan\beta\frac{\tan\left(\theta_0 + \theta_b\right)(m - 2v/2 - m)\tan\alpha - 1}{\tan\left(\theta_0 + \theta_b\right) + (m - 2v/2 - m)\tan\alpha}, \tag{15}$$

where β is half of the horizontal field of view angle of the camera; f_h is the horizontal focal length of the camera.

Assuming $z = 0$, according to the relationship established among the world coordinate system, camera coordinate system, and pixel coordinate system, the coordinate changes of pixel coordinate (u, v) on the x-axis and y-axis in the world coordinate system are updated when the vehicle motion turbulence causes the change of camera pitch angle θ_b. Therefore, the influence of image motion distortion on aerial view image is reduced, and the adaptive perspective transformation model is more robust to camera motion.

In the transformed aerial view, the pixels of the lane line are fitted by the second-order polynomial of the least-square algorithm, and the fitted curve is regressed to the original image so that the quadratic polynomial of the lane line in a real road scene is obtained.

5. Experimental Results and Analysis

The CULane dataset [13] was used to train and validate the proposed multilane line detection model, and then field experiments were conducted to verify the performance of the proposed model.

5.1. Experimental Environment Configuration. The algorithm design, training, and testing are based on the deep learning framework PyTorch. The experimental configuration used in this experiment is shown in Table 1.

5.2. CULane Dataset. Although the multilane detection network based on deep learning has been able to extract lane line pixels effectively in simple traffic scenes with good weather conditions, it is still challenging for lane line detection in complex scenes. Therefore, it is very important for supervised learning that the lane dataset covers a wide range of traffic scenes and has high annotation quality.

The challenging CULane dataset is used to train and validate the model, as shown in Figure 9. CULane is a large scale challenging dataset for academic research on traffic lane detection. Only the two lane lines on the left and right of the current lane line are concerned by CULane, which are paid most attention to in real applications. Therefore, there are only four real lane lines labeled. The dataset is divided into 88880 for the training set, 9675 for the validation set, and 34680 for the test set. Table 2 shows the various scene information of the CULane dataset.

It can be seen from Table 1 that a total of 9 types of traffic scenes are included in CULane. Among them, the number of images in normal traffic scenes with clear lane lines accounted for 27.7% of the total dataset, while the number of images in complex road scenes with unclear lane lines or interference with lane lines due to various reasons accounted for 72.3% of the dataset. It shows that complex road scenes are more concerned in CULane, and it is also consistent with the proportion of various traffic scenes encountered in the actual driving process. Therefore, this dataset is more suitable for verifying the ability of the proposed method to detect lane lines in complex road scenes.

5.3. Model Training Process

5.3.1. Data Processing Method. In order to ensure the real-time performance of the multilane line detection algorithm, the CULane input image is converted into an 800 ∗ 288 pixel format for network learning. The input data is normalized so that the value of the pixel is limited to the interval [0,1], which improves the computational efficiency of the model fitting process. The data augmentation strategies are used, including brightness conversion, horizontal flip, and overall image translation of 0~2 pixels; an example is shown in Figure 10.

5.3.2. Model Training Parameters Setting. Taking into account the performance of the hardware platform used in this study, the training parameters are shown in Table 3. In Table 3, batch size represents the number of training groups input for each iteration of training; epoch represents the training period of the whole training set; the learning rate decays exponentially to prevent gradient dispersion. In addition, in order to accelerate the convergence of the network, the MobileNet pretraining model is used to initialize the weight parameters of the unmodified part of the feature extraction network. The mainstream Glorot uniform distribution method is used for the initialization of other parameters.

5.4. Training Result Analysis. To verify the performance of the proposed multilane line detection algorithm based on deep learning, F_1 is used to judge whether a lane marking is successfully detected. The lane markings are viewed as lines with widths equal to 30 pixels, and the intersection over union (IoU) between the ground truth and the prediction is calculated. Predictions whose IoUs are larger than 0.5 are viewed as a correctly predicted lane line.

$$\begin{aligned}\text{precision} &= \frac{\text{TP}}{\text{TP} + \text{FP}},\\ \text{recall} &= \frac{\text{TP}}{\text{TP} + \text{FN}},\\ F_1 &= \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}},\end{aligned} \tag{16}$$

where TP (true positives) is the number of lanes lines correctly predicted; FP (false positives) is the number of wrongly predicted lane lines; FN (false negatives) is the number of missed ground-truth lane lines.

Based on the evaluation index of F_1, the effectiveness of the proposed algorithm and other multilane detection algorithms is analyzed. The experimental results are shown in Table 4.

From Table 4, it can be seen that the F_1 score of the proposed method in the normal test set and most challenge test sets is better than other algorithms, which verifies the effectiveness of the proposed method.

5.5. Examples of Multilane Line Detection Results. Figure 11 shows the multilane line detection results of the CULane dataset, and the curves of different colors represent the detected different lane lines. As can be seen from Figure 11, the proposed method can accurately detect multilane lines in various scenarios.

5.6. Real Vehicle Experimental Verification. To verify the positioning performance of the proposed strategy, experiments were conducted on the intelligent driving vehicle platform. The platform is refitted by Chery pure electric vehicle [31], as shown in Figure 12.

5.6.1. Camera Parameters. The HD industrial camera with a USB driver interface is selected as the visual acquisition device. The specific parameters are shown in Table 5.

Table 1: Experimental environment configuration parameters.

Hardware environment	GPU	GeForce GTX 2080Ti dual card processor
	Memory	32G
	Hard disk	2T
	Operating system	Ubuntu18.04
Software environment	Deep learning framework	PyTorch 1.2.0
	Programming language	Python 3.6.2
	CUDA	10.1
	CUDNN	7.6

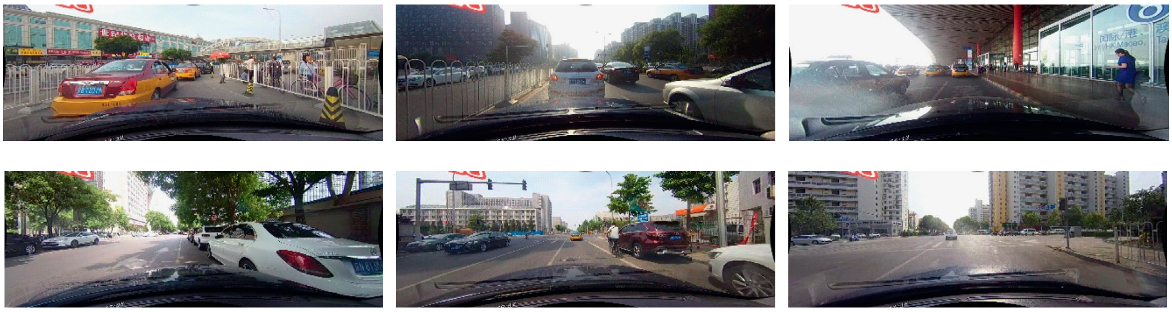


Figure 9: CULane dataset.

Table 2: The various scene information of the CULane dataset.

Scene	Scene description	Image proportion (%)
Normal	The driving vision is good, and the lane markings are clearly visible	27.7
Crowd	A large number of vehicles are moving slowly on the road, and the lane line is blocked by vehicles	23.4
Night	Visibility is low at night, lane lines are blurred, and different lights interfere with each other	20.3
No line	No lane markings, usually narrow roads, and many vehicles parked on the side of the road	11.7
Shadow	Under the sky bridge or viaduct, the light is dark, the shade of the trees or the shadow of the building	2.7
Strong light	Various strong lights cause roadside reflections, and the lane lines on the road are not clear	1.4
Curve	Curved lane line ahead	1.2
Crossroad	Interference of zebra crossing markings at intersections, no lane markings in the middle of the intersection	9.0
Arrow	There are interference marks between lane lines such as going straight, turning, and turning around	2.6

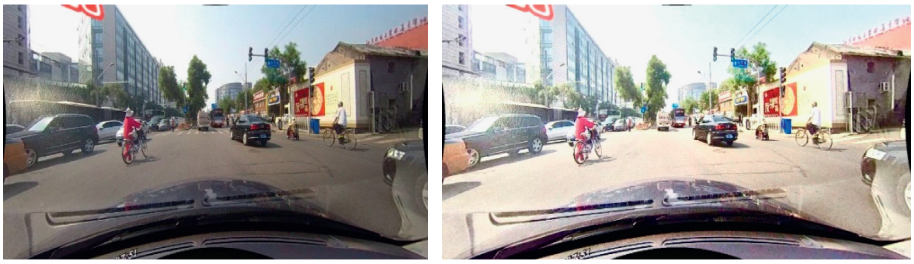


Figure 10: Data augmentation.

Table 3: Model training parameters.

Parameter	Data
The size of the input image	800 ∗ 288
Epoch	100
Batch size	16
Optimizer	Adam
Initial learning rate	1e-4

Table 4: Comparative experimental results of each improved part

Method	Scene (%)							
	Normal	Crowd	Night	No line	Shadow	Hlight	Curve	Arrow
SCNN [28]	90.6	69.6	66.1	43.4	66.9	58.5	64.4	84.1
GCJ [29]	89.7	**76.5**	68.7	35.1	65.5	**67.4**	63.2	82.2
SAD [30]	90.7	70.0	66.3	43.5	67.0	59.9	65.7	84.4
Our	**91.2**	72.8	**69.7**	**45.1**	**68.8**	64.4	**66.3**	**85.2**

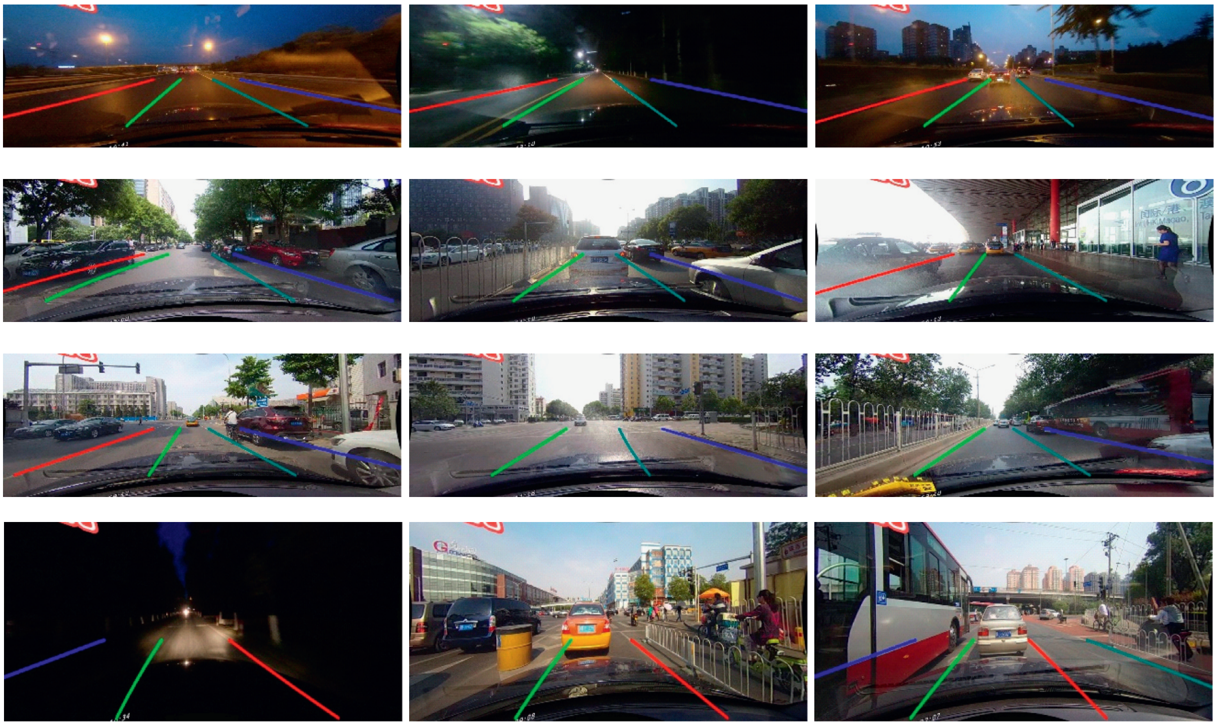


Figure 11: Examples of multilane line detection results.

Figure 12: The intelligent driving vehicle platform.

Table 5: Main parameter table of Camera.

Parameter	Data
Image format	RGB
Pixel size	$2\mu m * 2\,\mu m$
Frame rate	30 frame/s
Resolution	$1280 * 720$

The HD industrial camera can maintain a good shooting effect without distortion under strong light, weak light, or no light, and the image acquisition requirements under various driving conditions can be met.

5.6.2. Industrial Computer. Advantech MIC-7700 industrial computer is selected as the data computing platform, as shown in Figure 13. MIC-7700 is a compact and fan-free computing platform designed based on the industrial market. It can work in an environment of 0~50 degrees, can reduce the external vibration and impact on the inside of the chassis, and can be used all day in bad weather.

5.6.3. Software Platform. The real vehicle data acquisition and experiment in this paper are carried out in ROS (robot operating system). Ubuntu 16.04 is used as the installation and deployment operating system of ROS. The version of ROS is kinetic.

To achieve multitask road multitarget detection and the integration of the drivable area segmentation model on the ROS platform, as shown in Figure 14, the topic structure diagram of multitask target detection and segmentation node and network connection mode is established.

Firstly, node/lane_node is created, which subscribes to the image Topic/usb_cam/image_raw of real-time road conditions driven by the HD camera installed on the vehicle in real time, and converts the image format into the format recognized by the OpenCV tool library by calling the

Figure 13: MIC-7700 industrial computer.

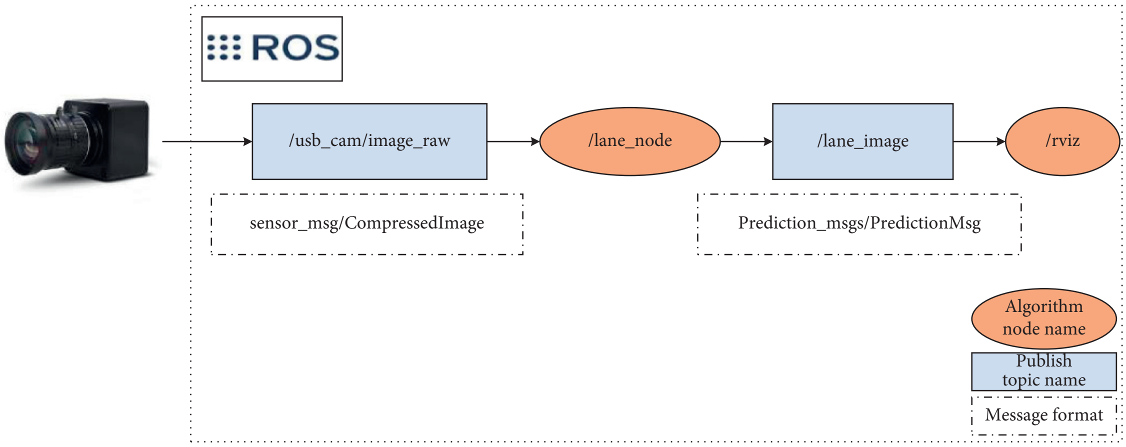
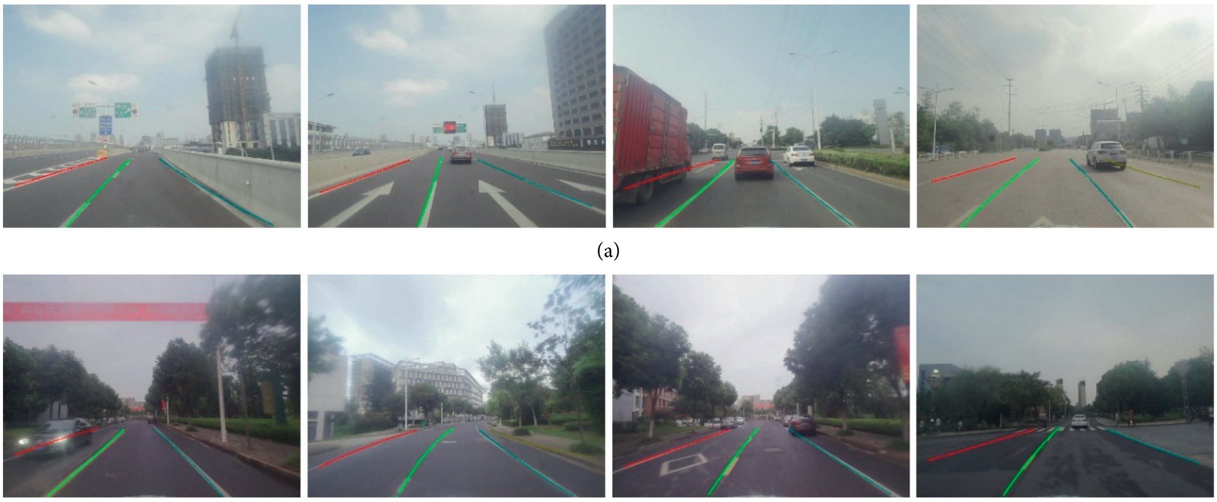


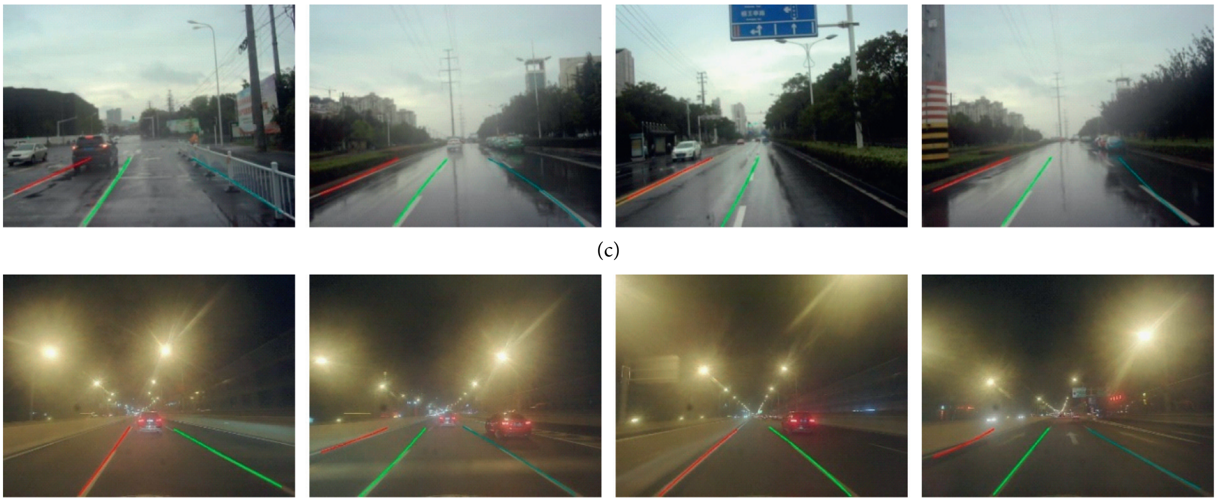
Figure 14: Node and topic structure diagram of ROS.



(a)

(b)

Figure 15: Continued.



(c)

(d)

Figure 15: Lane line detection results in different scenarios. (a) Sunny day. (b) Overcast. (c) Rainy day. (d) Night.

cv_bridge module in the ROS library. Then the image data is reasoned according to the multitask road target detection and segmentation model, and finally, the results are published in real time as topic/lane_image. ROS graphical tool RViz can display the multitask road multitarget detection and drivable area segmentation prediction results by subscribing to the topic.

5.6.4. Real Vehicle Test of Multilane Line Detection Algorithm. Figure 15 shows the multilane line detection effect of the algorithm proposed in this paper for the data collected by the onboard vision system of the intelligent driving vehicle platform. It can be seen that the proposed method can achieve better detection results in various complex scenarios. The reason is that the lane line detection problem is transformed into an instance segmentation problem, which can solve any number of lane line detection and lane change problems. Even if the lane line is blocked by vehicles or shadows or the detected lane line is a dashed line, the position of the lane line can be accurately predicted.

6. Conclusion and Discussion

Aiming at the difficulty of lane line detection in complex urban traffic scenes, a new method of lane line detection based on instance segmentation and adaptive perspective transformation is proposed in this paper. A branch multitask lane line instance segmentation network is designed for lane line instance segmentation, including the lane line semantic segmentation part and the lane line Id embedding part, which can solve the problems of multilane and lane change detection while satisfying real-time performance. The CULane dataset is used for training. At the same time, in order to deal with the change of camera inclination angle caused by vehicle turbulence, the example segmentation results are adaptively perspective transformed to obtain the lane line pixel set from the perspective of aerial view so as to realize the fitting of lane line pixels. The CULane test results show that the lane line detection effect F_1 in normal scenes is 91.2%. Integrating the model into the ROS platform realizes the real-time detection of multilane lines in various complex traffic scenes, which has good practical application value. We also conclude that the proposed strategies can be easily embedded into other advanced driver assistance approaches with slight modifications.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflicts of interest.

Authors' Contributions

Xiang Song and Hai Wang contributed to methodology; Xiang Song, Xiaoyu Che, and Huilin Jiang contributed to software; Xiang Song, Ling Li, Chunxiao Ren, and Hai Wang contributed to validation; Xiang Song, Shun Yan, and Hai Wang contributed to investigation; Xiang Song was responsible for original draft preparation; Xiang Song, Xiaoyu Che, Chunxiao Ren, and Hai Wang responsible for review and editing; Xiang Song, Ling Li, and Huilin Jiang contributed to funding acquisition. All authors have read and agreed to the published version of the paper.

Acknowledgments

This research was funded by the National Natural Science Foundation of China (Grant no. 61801227), the Future Network Scientific Research Fund Project (Grants nos. FNSRFP-2021-YB-29 and FNSRFP-2021-YB-28), the Qing Lan Project of Jiangsu (Grant no. QLGC-2020), and the

Natural Science Foundation of the Jiangsu Higher Education Institutions of China (Grant no. 20KJB130005).

References

[1] Z Liu, Y Cai, H Wang et al., "Robust target recognition and tracking of self-driving cars with radar and camera information fusion under severe weather conditions," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–14, 2021.

[2] D. Shin, B. Kim, K. Yi, A. Carvalho, and F. Borrelli, "Human-centered risk assessment of an automated vehicle using vehicular wireless communication," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 2, pp. 667–681, 2019.

[3] A. Bar Hillel, R. Lerner, D. Levi, and G. Raz, "Recent progress in road and lane detection: a survey," *Machine Vision and Applications*, vol. 25, no. 3, pp. 727–745, 2014.

[4] H. Liu, X. Wu, and W. Zhang, "Lane detection in structural road by using linear model method [J]," *International Journal of Research in Engineering and Science*, vol. 4, no. 8, pp. 20–24, 2016.

[5] W. Chen, W. Wang, K. Wang, Z. Li, H. Li, and S. Liu, "Lane departure warning systems and lane line detection methods based on image processing and semantic segmentation: a review," *Journal of Traffic and Transportation Engineering*, vol. 7, no. 06, pp. 25–51, 2020.

[6] Q. Li, J. Zhou, B. Li, Y. Guo, and J. Xiao, "Robust lane-detection method for low-speed environments," *Sensors*, vol. 18, no. 12, p. 4274, 2018.

[7] W. Li, F. Qu, Y. Wang, L. Wang, and Y. Chen, "A robust lane detection method based on hyperbolic model," *Soft Computing*, vol. 23, no. 19, pp. 9161–9174, 2019.

[8] H Xiong, D Yu, J Liu et al., "Fast and robust approaches for lane detection using multi-camera fusion in complex scenes," *IET Intelligent Transport Systems*, vol. 14, no. 12, pp. 1582–1593, 2020.

[9] M. Aly, "Real time detection of lane markers in urban streets," in *Proceedings of the 2008 IEEE Intelligent Vehicles Symposium (IV)*, pp. 7–12, IEEE, Eindhoven, Netherlands, July 2008.

[10] F. Bounini, D. Gingras, V. Lapointe, and H. Pollart, "Autonomous vehicle and real time road lanes detection and tracking," in *Proceedings of the 2015 IEEE Vehicle Power and Propulsion Conference (VPPC)*, pp. 1–6, IEEE, Montreal, QC, Canada, October 2015.

[11] J. Tang, S. Li, and P. Liu, "A review of lane detection methods based on deep learning," *Pattern Recognition*, vol. 111, Article ID 107623, 2021.

[12] P. Amayo, T. Bruls, and P. Newman, "Semantic classification of road markings from geometric primitives," in *Proceedings of the 2018 IEEE International Conference on Intelligent Transportation Systems (ITSC)*, pp. 387–393, IEEE, Maui, HI, USA, November 2018.

[13] W. Song, Y. Yang, M. Fu, Y. Li, and M. Wang, "Lane detection and classification for forward collision warning system based on stereo vision," *IEEE Sensors Journal*, vol. 18, no. 12, pp. 5151–5163, 2018.

[14] Y. Cai, L. Dai, H. Wang, L. Chen, and Y. Li, "DLnet with training task conversion stream for precise semantic segmentation in actual traffic scene," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2021.

[15] Y. Cai, L. Dai, H. Wang et al., "Pedestrian motion trajectory prediction in intelligent driving from far shot first-person perspective video," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–16, 2021.

[16] Y. Cai, T. Luan, H. Gao et al., "YOLOv4-5D: an effective and efficient object detector for autonomous driving," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–13, 2021.

[17] R. Muthalagu, A. Bolimera, and V. Kalaichelvi, "Lane detection technique based on perspective transformation and histogram analysis for self-driving cars," *Computers & Electrical Engineering*, vol. 85, Article ID 106653, 2020.

[18] X. Li, J. Li, X. Hu, and J. Yang, "Line-CNN: end-to-end traffic line detection with line proposal unit," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 1, pp. 248–258, 2020.

[19] W. Van Gansbeke, B. De Brabandere, D. Neven, M. Proesmans, and L. Van Gool, "End-to-end lane detection through differentiable least-squares fitting," in *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pp. 905–913, Seoul, Korea (South), October 2019.

[20] Y Sun, S Ma, S Sun et al., "Partial discharge pattern recognition of transformers based on MobileNets convolutional neural network," *Applied Sciences*, vol. 11, no. 15, p. 6984, 2021.

[21] J. Cheng, W Ding, X. Ku, and J. Sun, "Road extraction from high-resolution SAR images via automatic local detecting and human-guided global tracking," *International Journal of Antennas and Propagation*, vol. 201210 pages, Article ID 989823, 2012.

[22] S. Li, J. Xu, W. Wei, and H. Qi, "Curve lane detection based on the binary particle swarm optimization," in *Proceedings of the 2017 29th Chinese Control And Decision Conference (CCDC)*, pp. 75–80, IEEE, Chongqing, China, May 2017.

[23] C. Jin, X. Wang, Z. Miao, and S. Ma, "Road curvature estimation using a new lane detection method," in *Proceedings of the 2017 Chinese Automation Congress (CAC)*, pp. 3597–3601, IEEE, Jinan, China, October 2017.

[24] J. Jeong and A. Kim, "Adaptive inverse perspective mapping for lane map generation with SLAM," in *Proceedings of the 2016 13th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, pp. 38–41, IEEE, Xian, China, Auguest 2016.

[25] B. Lourenco, V. Santos, M. Oliveira, and T. Almeida, "Performance analysis on deep learning semantic segmentation with multivariate training procedures," in *Proceedings of the 2020 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, Ponta Delgada, Portugal, April 2020.

[26] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 318–327, 2017.

[27] T. N. Tran, K. Drab, and M. Daszykowski, "Revised DBSCAN algorithm to cluster data with dense adjacent clusters," *Chemometrics and Intelligent Laboratory Systems*, vol. 120, pp. 92–96, 2013.

[28] Z. Shan, D. Huang, and Z. Geng, "End-to-End automatic driving based on spatial CNN," *Journal of Changchun University of Science and Technology*, vol. 44, no. 3, pp. 102–108, 2021.

[29] J. Zhang, Y. Xu, B. Ni, and Z. Duan, "Geometric constrained joint lane segmentation and lane boundary detection," in *Proceedings of the European Conference on Computer Vision*

(ECCV), pp. 486–502, Springer, Munich, Germany, September 2018.

[30] X. Zhang, H. Huang, W. Meng, and D. Luo, "Improved lane detection method based on convolutional neural network using self-attention distillation," *Sensors and Materials*, vol. 32, no. 12, p. 4505, 2020.

[31] Y. Yu, H. Wang, Y. Cai, L. Chen, and J. Chen, "A robust lane detection method for low memory and computation resource platform," in *Proceedings of the 2019 IEEE 9th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*, pp. 854–860, IEEE, Suzhou, China, Auguest 2019.