International Journal of
Geo-Information

Article

A Novel Lane Line Detection Algorithm for Driverless Geographic Information Perception Using Mixed-Attention Mechanism ResNet and Row Anchor Classification

Yongchao Song [1,†], Tao Huang [1,2,†], Xin Fu [3,4,*], Yahong Jiang [4], Jindong Xu [1], Jindong Zhao [1], Weiqing Yan [1] and Xuan Wang [1,†]

[1] School of Computer and Control Engineering, Yantai University, Yantai 264005, China
[2] School of Information Science and Engineering, Shandong Normal University, Jinan 250014, China
[3] Engineering Research Center of Highway Infrastructure Digitalization, Ministry of Education, Xi'an 710064, China
[4] College of Transportation Engineering, Chang'an University, Xi'an 710064, China
* Correspondence: fuxin@chd.edu.cn
† These authors contributed equally to this work.

Citation: Song, Y.; Huang, T.; Fu, X.; Jiang, Y.; Xu, J.; Zhao, J.;Yan, W.; Wang, X. A Novel Lane Line Detection Algorithm for Driverless Geographic Information Perception Using Mixed-Attention Mechanism ResNet and Row Anchor Classification. *ISPRS Int. J. Geo-Inf.* **2023**, *12*, 132. https://doi.org/10.3390/ijgi12030132

Academic Editors: Wolfgang Kainz and Maria Antonia Brovelli

Received: 24 November 2022
Revised: 27 February 2023
Accepted: 16 March 2023
Published: 20 March 2023

Abstract: Lane line detection is a fundamental and critical task for geographic information perception of driverless and advanced assisted driving. However, the traditional lane line detection method relies on manual adjustment of parameters, and has poor universality, a heavy workload, and poor robustness. Most deep learning-based methods make it difficult to effectively balance accuracy and efficiency. To improve the comprehensive perception ability of lane line geographic information in a natural traffic environment, a lane line detection algorithm based on a mixed-attention mechanism residual network (ResNet) and row anchor classification is proposed. A mixed-attention mechanism is added after the backbone network convolution, normalization and activation layers, respectively, so that the model can focus more on important lane line features to improve the pertinence and efficiency of feature extraction. In addition, to achieve faster detection speed and solve the problem of no vision, the method of lane line location selection and classification based on the row direction is used to detect whether there are lane lines in each candidate point according to the row anchor, reducing the high computational complexity caused by segmentation on a pixel-by-pixel basis of traditional semantic segmentation. Based on TuSimple and CurveLane datasets, multi-scene, multi-environment, multi-linear road image datasets and video sequences are integrated and self-built, and several experiments are designed and tested to verify the effectiveness of the proposed method. The test accuracy of the mixed-attention mechanism network model reached 95.96%, and the average time efficiency is nearly 180 FPS, which can achieve a high level of accuracy and real-time detection process. Therefore, the proposed method can meet the safety perception effect of lane line geographic information in natural traffic environments, and achieve an effective balance between the accuracy and efficiency of actual road application scenarios.

Keywords: road geographic information; safe driving; lane line detection; mixed-attention mechanism; row anchor

1. Introduction

In China, with the rapid increase in the number of cars and the vigorous development of road construction, drivers have to face more complex road conditions in the driving process. They urgently need road geometric geographic information that can help drive safe [1]. Based on this, detecting lane line geographic information is a crucial step to achieve road safety driving [2].

The existing lane line detection methods can be summarized into two types, one is the traditional image processing method [3–5], and the other is the method based on deep learning [6–8]. The traditional lane line detection method divides the lane line area by means of edge detection and filtering, and then combines Hough transform [9], RANSAC [10] and other algorithms. However, these methods require manual adjustment of filter operators according to the actual scene, which has a heavy workload and poor robustness. When the driving environment changes obviously, the lane line detection effect is not good [11].

With the rapid development of deep learning in computer vision, lane line detection methods based on deep learning have gradually become the mainstream. Lane line detection methods based on deep learning can be divided into three categories: segmentation, point-based and curve-based modeling methods. The methods based on point detection [12,13] usually use the R-CNN framework [14,15] to detect lane line by detecting a series of dense pixels, which is inefficient. Based on the segmentation methods [16,17], the lane line pixels are segmented one by one by using multi feature clues such as foreground texture, and these pixels are decoded into lane line instances through heuristic methods, with high calculation cost. However, as an essential component of automatic driving and advanced assistant driving, lane line detection tasks are frequently executed during operation, requiring extremely low calculation costs. Based on curved modeling methods, inefficient cyclic feature aggregation is often used under sheltered or unfavorable weather/lighting conditions, but the calculation cost is too high. A few methods [18,19] model the lane line as a polynomial curved to simulate the geometric characteristics, but abstract polynomial coefficients are difficult to learn. The research shows that the performance of lane line detection based on curved modeling is significantly behind the well-designed methods based on segmentation and point detection.

Based on the above, we propose a novel lane line detection algorithm based on mixed-attention mechanism residual network (ResNet) and row anchor classification. ResNet [20] is the backbone network to improve the reasoning effect of the model. A method of adding mixed-attention mechanisms after convolutional, normalization and activation layers in backbone networks is proposed, so that the model can focus more on important lane line features. The method of lane line position selection and classification based on row direction is adopted to obtain faster detection speed and solve the problem of no view. Finally, an effective balance between the accuracy and efficiency of lane line geographic information detection is achieved in the actual road application scenario.

In summary, the main contributions of this paper are as follows:

- A mixed-attention mechanism is proposed to be added after the backbone network convolution, normalization and activation layers, so that the model can focus more on important lane line features to improve the pertinence and efficiency of feature extraction.
- The method of lane line location selection and classification based on the row direction is adopted, and the row index is carried out according to the row anchor points to detect whether there is lane line in each candidate point, so as to achieve faster detection speed and solve the problem of no field of vision.
- The road video test sequences of multi-scene, multi-environment and multi-line type are built by ourselves to demonstrate the effectiveness and universality of the lane geographic information detection method, so as to lay a foundation for practical application.

2. Related Work

Currently, the methods of lane line detection are mainly divided into traditional and deep learning methods. The traditional lane line detection methods are divided into model-based methods and feature-based methods. The lane line detection methods based on deep learning are emerging recently, with relatively good detection effects, and gradually become the mainstream.

2.1. Traditional Lane Line Detection Methods

2.1.1. Feature-Based Lane Line Detection

According to the difference between the lane line and other pavement features, the feature-based method uses the edge [21], texture [22] and gradient [23] of the lane line and pavement to judge, so as to realize the function of lane line detection.

Yoo et al. [24] proposed a lane line detection method based on linear discriminant analysis and gradient enhancement conversion, which converted the original RGB image of the road into a new gray image, and can dynamically adjust the gray conversion vector to cope with a variety of light changes. Wei et al. [25] proposed a lane line detection method based on constrained Hough transform double-edge extraction, which used lane line width and color features to extract lane line regions, used a Canny edge detector to obtain lane line edges, and finally recognized lines through modified Hough transform. Mammeri et al. [26] proposed a lane line detection system based on the combination of maximum stable extreme value region (MSER) and Hough transform. They used MSER technology to locate the area of interest, and introduced a three-stage fine calculation method to enhance the results of MSER and filter out unwanted information.

2.1.2. Model-Based Lane Line Detection

The model-based lane line detection method uses the extracted lane line features to match the lane line model to achieve detection. Still, it needs to constantly adjust the model and parameters to ensure the detection effect of various road conditions.

In the past decades, different lane line models have been proposed, including simple straight line models and complex models such as parabola, hyperbola, B-spline and clothoid curve. To describe the constant change between straight and circular curves, Zhao et al. [27] used a high-order lane line model, but with the increase in model order, it became more sensitive to noise. The spline-based lane line model describes a wider range of lane structures through a group of different control points forming arbitrary shapes. He et al. [28] proposed curved lane line detection based on Catmull-Rom spline curve, and used the Hough transform and Catmull-Rom to fit the near and far lane lines. Lee et al. [29] proposed a cascaded particle filter algorithm based on model decomposition to detect lane lines. The lane model was divided into two parts: the straight and curved model, which can detect curves and straights.

In consequence, traditional lane detection methods need to extract features manually, which is affected by different detection scenarios. Moreover, these methods require manual modification of experimental parameters, with low robustness.

2.2. Deep Learning Lane Line Detection Methods

Nowadays, the lane line detection method based on deep learning has become the mainstream [30–32]. Different models can be selected, and different parameters can be set, which can adapt to various complex environments. Compared with traditional lane line detection methods, it has better robustness.

However, the current CNN-based methods still have defects in meeting the challenges of the above traditional methods. The typical method [16] first generated segmentation results, then used post-processing methods to generate lane lines, such as segmentation clustering and curved fitting. These post-processing methods were inefficient, and ignored the global environment when learning to segment lanes. In order to improve the efficiency of the algorithm, the literature [19] transferred the pipeline method in target detection to lane detection without the above segmentation process and post-processing. However, it relied on complex anchor design selection and additional non-maximum suppression, making it slower than most lane line detectors. Xu et al. [13] proposed a new lane sensitive architecture search framework called CurveLane-NAS, which combined multi-level prediction heads with multi-level feature fusion. For a long lane line, it was divided into the whole straight line part and the local bending part to capture features. The detection accuracy was high, but the efficiency needed to be improved. Liu et al. [18] proposed an end-to-end method to directly output the parameters of the lane model, and used Transformer model and self Attention mechanism to obtain the unique structure of the lane line. This method had higher learning efficiency, but the detection accuracy needed to be improved.

In a nutshell, the traditional lane line detection methods use edge detection, model matching, and so forth, to segment the lane area, which requires manual adjustment of the filter operator, with a heavy workload and poor robustness. Moreover, most of the existing deep learning lane line detection methods are difficult to achieve an effective balance between detection accuracy and efficiency.

3. Lane Line Detection Algorithm

We propose a new lane line detection algorithm that mainly consists three modules: feature extraction module based on mixed-attention mechanism ResNet, auxiliary segmentation module and classification module based on row anchor. With the cooperation of these modules, the lane line detection function can be realized with high performance and efficiency. The network architecture of the entire lane line detection algorithm is shown in Figure 1.

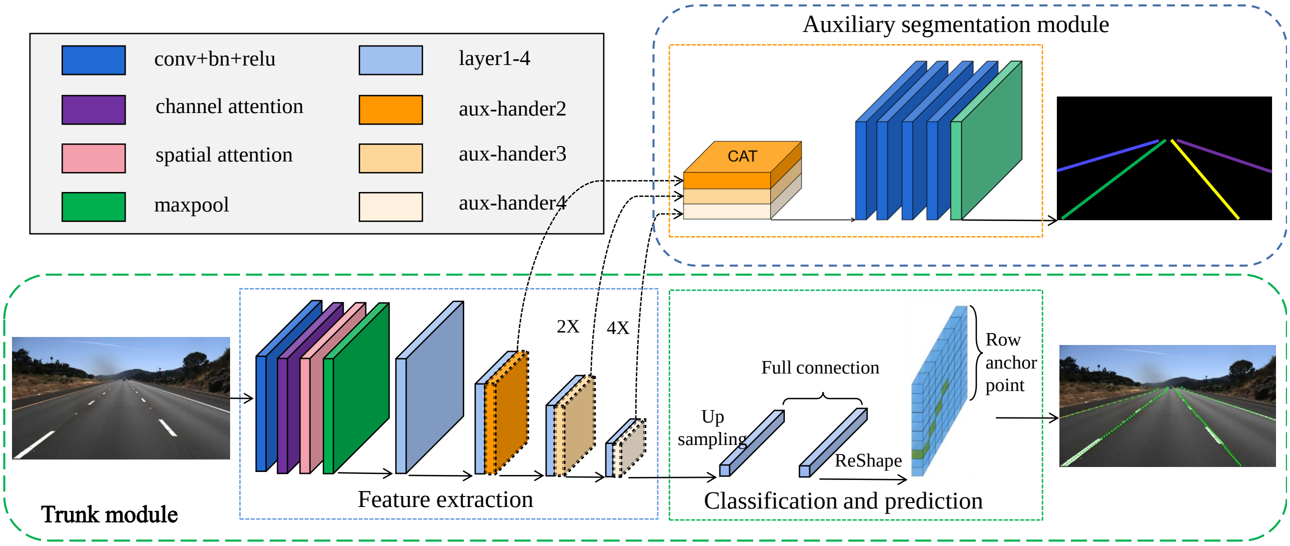


Figure 1. Overall network architecture.

3.1. Feature Extraction Based on Mixed-Attention Mechanism ResNet

Traditional CNN realizes feature extraction by stacking convolution layer and subsampling layer. However, when stacked to a certain depth, gradient disappearance, gradient explosion and degradation will occur, which will affect the speed and accuracy of lane detection. Therefore, this paper uses ResNet residual network as the backbone network, adds an identity mapping, and directly transmits the current output to the next layer of the network to solve the problem of gradient disappearance in the deep network and improve the model reasoning effect.

At the same time, to improve the effectiveness and pertinence of feature extraction, this paper adds channel attention mechanism and spatial attention mechanism after convolution, normalization and activation layer on the basis of ResNet backbone network.It improves the reasoning ability of the model by introducing the mixed-attention mechanism, so that the model can focus more on essential features. Lane line detection is to find the lane line from the pavement and detect it, so the lane line is an important feature, while the pavement is a negligible part. The spatial attention mechanism and the channel attention mechanism make the model more aware of the “where” (influenced by spatial attention) and “what to do” (influenced by channel attention). The feature extraction module of the mixed-attention mechanism ResNet is shown in Figure 2, in which aux-hander 2–4 layers are only used in the auxiliary segmentation module during training, and their inputs are the outputs of ResNet Layer 2–4 layers, respectively.

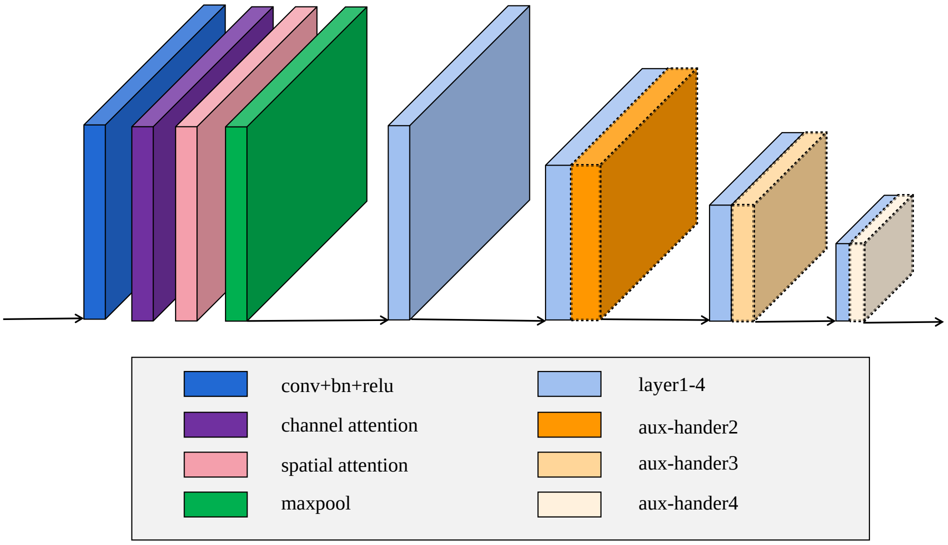


Figure 2. Mixed-attention mechanism module.

As shown in Figure 2, features are extracted from the input image through multiple convolutions, pooling and activation functions, and the most important features are extracted through the mixed-attention mechanism. Specifically, the lane features with higher effectiveness and pertinence are extracted from the input images after the mixed-attention mechanism ResNet. The original input image is downsampled four times to extract relatively shallow features. Through global features and larger field of perception, the global information is increased and the network reasoning ability in the absence of vision is improved. Since the ResNet networks used in this paper are mainly ResNet-18 and ResNet-34, the basic block residual blocks are used in layers 1–4 of the module.

Aux-hander 2–4 layers are mainly used in training by a segmentation module. As shown in Figure 1, semantic segmentation contact the extracted three-layers shallow features, and the spliced features are upsampled. This part uses cross-entropy as segmentation loss to enhance visual features during training.

3.2. Auxiliary Segmentation Module

The auxiliary segmentation module uses semantic segmentation to tensor concatenation for the extracted three shallow features, and upsampling the spliced features. This part uses cross-entropy as segmentation loss, which is only used in the training process to enhance visual features. The input information corresponding to the auxiliary segmentation is the output information of each residual block of the mixed-attention mechanism ResNet backbone network, which realizes the auxiliary segmentation method of using multi-scale features to model local features, as shown in Figure 3.

Specifically, the input of aux-hander 2–4 is the data generated by the residual block 2–4 layer of the mixed-attention mechanism ResNet. After convolution and normalization activation operations, the image is up-sampled and participated in segmentation. The specific introduction of each layer is shown in Figure 4, where (a) aux-hander 2, (b) aux-hander 3 and (c) aux-hander 4 are the detailed parameters of the corresponding layers, including the number of network input channels, the number of output channels and the size of the self-convolution kernel.

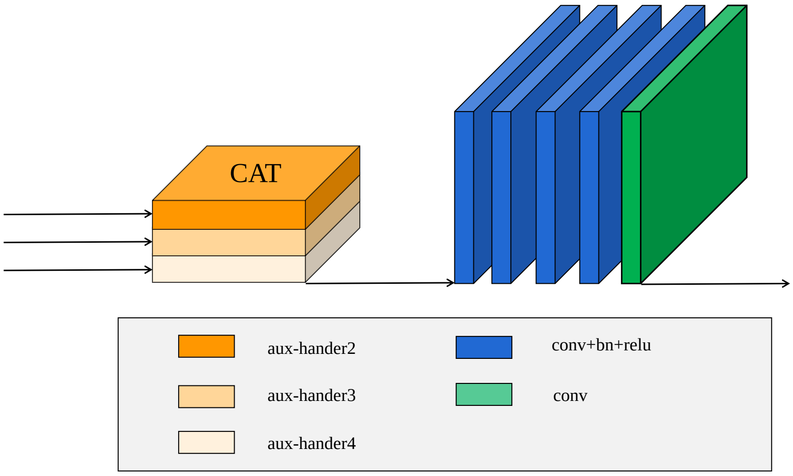


Figure 3. Auxiliary segmentation module.

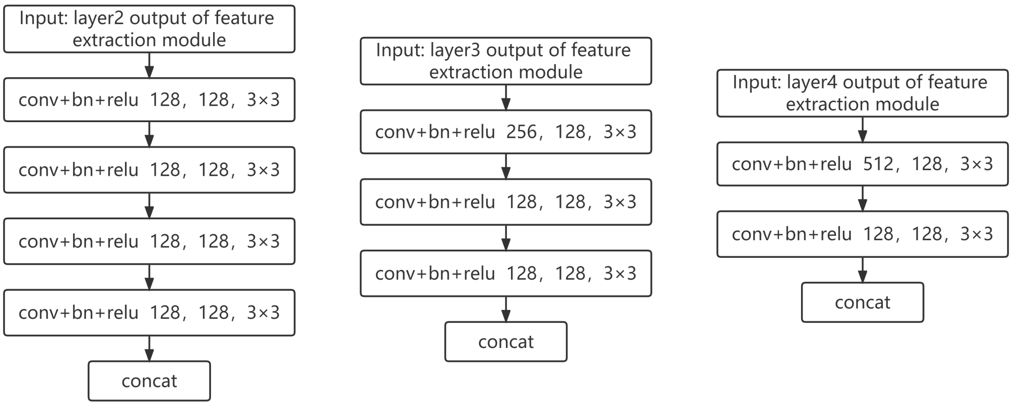

Input: layer2 output of feature extraction module
conv+bn+relu 128, 128, 3×3
conv+bn+relu 128, 128, 3×3
conv+bn+relu 128, 128, 3×3
conv+bn+relu 128, 128, 3×3
concat
Input: layer3 output of feature extraction module
conv+bn+relu 256, 128, 3×3
conv+bn+relu 128, 128, 3×3
conv+bn+relu 128, 128, 3×3
concat
Input: layer4 output of feature extraction module
conv+bn+relu 512, 128, 3×3
conv+bn+relu 128, 128, 3×3
concat

Figure 4. Aux-hander schematic of the two to four layers.

3.3. Lane Line Location Selection and Classification Based on the Row Direction

At this stage, thinking of how to achieve faster detection speed and solve the problem of no vision while maintaining the accuracy of lane line detection is a significant challenge. Based on this, this paper adopts the location selection and classification method of lane lines based on row direction [33]. In terms of global features, row indexes are carried out according to row anchor points to detect whether there are lane lines in each candidate point, which reduces the high computational complexity caused by pixel-by-pixel segmentation of traditional semantic segmentation.

The selection and classification method based on the position in the row direction converts the lane line detection to the selection of a specific row anchor point on a predefined row, as shown in Figure 5. First, the image is gridded and divided into several adjacent feature blocks by row. Then, the lane lines in each feature block are clustered to determine whether there are lane lines in each feature block and marked. As it does not need to detect every point pixel by pixel, as with traditional semantic segmentation, detection is only based on feature blocks, which greatly reduces the amount of computation and is more suitable for the real-time requirements of practical applications.

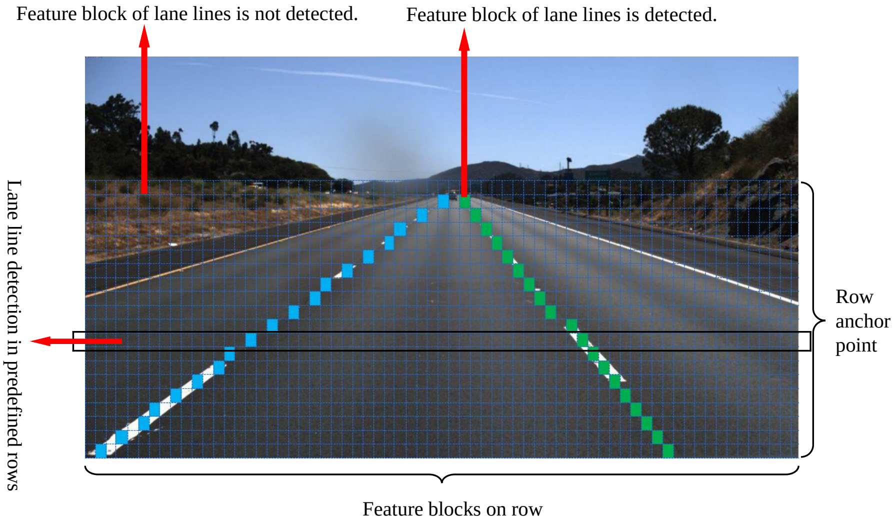

Feature block of lane lines is not detected.
Feature block of lane lines is detected.
Lane line detection in predefined rows
Row anchor point
Feature blocks on row

Figure 5. Schematic diagram of selection and classification methods based on row direction [33].

The image is resized into 288 × 800 pixels, and the height and width of the image are represented by H and W, respectively. The number of predefined lines is set to h. The number of feature blocks divided on each row is w, and the number of lane lines is assumed to be C. Assuming that X is the global image feature, $f_{i,j}$ is the classifier for the position of lane i and the anchor point of row j. The lane prediction result is:

$$P_{i,j,:} = f_{i,j}(X), \quad i \in [1, C], j \in [1, h] \tag{1}$$

where $P_{i,j,:}$ is a $w+1$ dimensional vector, which is expressed as the probability of selecting $w+1$ grid cells for the lane i and the row j of anchor points. Assuming that $T_{i,j,:}$ is labeled in the correct position, the formula can be optimized as follows:

$$L_{\text{cls}} = \sum_{i=1}^{C} \sum_{j=1}^{h} L_{CE}\left(P_{i,j,:}, T_{i,j,:}\right) \tag{2}$$

where L_{CE} represents cross-entropy loss. It can be seen that for the selection and classification method of the lane line position based on the row direction, the calculation amount is $h \times (w+1) \times C$, while for the segmentation method, the calculation amount is $H \times W \times (C+1)$, and h and w are much smaller than H and W. Therefore, the calculation amount of this method is small and the detection speed is fast.

In order to solve the problem of no vision, this paper defines the lane structure loss function with the help of context information and global information from the image:

$$L_{\text{str}} = L_{\text{sim}} + \lambda \, L_{\text{shp}} \tag{3}$$

where L_{sim} is the similarity loss, L_{shp} is the shape loss, and λ is the loss coefficient.

Since the lane lines are continuous and adjacent, after the feature blocks are divided according to rows, the lane lines at the adjacent row anchor points are also adjacent. Therefore, by constraining the distribution of each lane line on the feature block, the continuity between lane lines can be realized. The similarity loss function is defined as shown in Equation (4). The smoothness is defined by L1 normal form to constrain the distance between two adjacent points.

$$L_{\text{sim}} = \sum_{i=1}^{C} \sum_{j=1}^{h-1} \left\| \left(P_{i,j,:}, P_{i,j+1,:}\right) \right\|_1 \tag{4}$$

where $\|\cdot\|_1$ represents L1 normal form.

Shape loss refers to the definition of lane shape by the second-order difference between adjacent rows. Since lane lines are mainly straight, the second-order difference is set as 0, and the difference between the second-order difference and 0 is constrained to make the predicted lane lines straight.

First, the Softmax function is used to obtain the possibility of the lane at different locations, then the approximate value of the lane is replaced by the expected value predicted by the lane line. The probability of the lane at each location is defined as:

$$Prob_{i,j,:} = Softmax\left(P_{i,j,1:w}\right) \tag{5}$$

Since there is no counting in the background grid, the position range of the row is from 1 to w. Combining Equation (5), the probability of the lane at each position can be calculated. Then, the expected $Loc_{i,j}$ of the position is expressed in Equation (6), where $Prob_{i,j,:}$ is the probability of the lane at each feature block.

$$Loc_{i,j} = \sum_{k=1}^{w} k \cdot Prob_{i,j,k} \tag{6}$$

Finally, the constraint of the second-order difference of lane shape is expressed by Equation (7). The second-order difference is used because it is less constrained than the first-order difference. In this way, even if the lane is not straight, the actual effect will not be affected due to the loss function problem.

$$L_{\text{shp}} = \sum_{i=1}^{C} \sum_{j=1}^{h-2} \left\| \left(Loc_{i,j} - Loc_{i,j+1}\right) - \left(Loc_{i,j+1} - Loc_{i,j+2}\right) \right\|_1 \tag{7}$$

4. Experiment Description

In this part, we carry out the preparatory work before the experiment and design several experiments to evaluate the performance of the lane line detection method proposed in this paper. It mainly includes dataset preparation and training, evaluation index analysis, training set determination, and backbone network depth selection.

4.1. Datasets Preparation and Training

The composition of the training dataset is based on the TuSimple dataset [34] and CurveLane dataset [13]. In TuSimple, the scene is relatively simple, such as the highway scene, and most of them are straight roads. CurveLane is a curved dataset built by Huawei, China. It contains a variety of road scenes, with many cases of poor lighting, road damage and vehicle occlusion, and most of them are curves. Details are shown in Table 1.

Table 1. Dataset description.

Dataset	Training	Testing	Resolution	Scenes
Tusimple	3268	2782	1280 × 720	Highways (mainly straight)
Curvelane	100,000	30,000	2560 × 1440	Various complex scenes (curves mainly)

The experiment found that if only the TuSimple was used as the training set, the trained model had poor adaptability and could only achieve relatively stable detection on a straight-line road. However, if only the CurveLane was used, the scenes were relatively complex. When encountering some simple linear scenes, the trained model could not adapt to them, and the actual detection results are unsatisfactory. Therefore, in this paper we decided to combine the two datasets for training, which cannot only cope well with some complex scenes, but can also ensure the detection of simple scenes. At the same time, combining the datasets of curves and straights can enhance the adaptation effect of the trained model with better robustness.

Three training sets are constructed. For convenience, the three datasets are named DataSet-1, DataSet-2 and DataSet-3. DataSet-1 consists of 15,000 images in the CurveLane, DataSet-2 consists of 15,000 images in the CurveLane and 2000 images in the TuSimple, and DataSet-3 consists of 15,000 images in CurveLane and 3500 images in TuSimple. The details are shown in Table 2.

Table 2. Training set construction description.

Dataset	Training	Resolution	Scenes
DataSet-1	15,000	2560 × 1440	Various complex scenes (curves mainly)
DataSet-2	17,000	2560 × 1440, 1280 × 720	Various complex scenes (curves mainly)
DataSet-3	18,500	2560 × 1440, 1280 × 720	Various complex scenes (curves mainly)

Since the annotation format of CurveLane is different from that of TuSimple, the annotation format of the two datasets is incompatible in the process of model training, so it is necessary to convert the annotation of CurveLane into TuSimple format so that they can be combined. Specifically, you first need to select the desired images and their annotations from the CurveLane, and then create a text document to select the image range for the dataset annotation format conversion. The same json file as the TuSimple annotation can then be generated. Finally, since the transformed CurveLane and Tusimple do not provide segmentation annotation, the json annotation of the two datasets should be used to create segmentation data and generate the train_gt.txt file based on training. Once this has been completed, you can train by using either of these datasets or combining them.

The main evaluation indexes of the model effect in this paper are Accuracy, FP and FN. FP refers to the negative samples predicted by the model as positive, and FN refers to the positive samples predicted by the model as negative. The accuracy is calculated by Equation (8).

$$\text{Accuracy} = \frac{\sum_{\text{clip}} C_{\text{clip}}}{\sum_{\text{clip}} S_{\text{clip}}} \tag{8}$$

where C_{clip} refers to the number of lane line points correctly predicted, and S_{clip} refers to the total number of actual lane line points.

The datasets DataSet-1, DataSet-2 and DataSet-3 are taken as the training sets, Tusimple as the test set, and ResNet-18 as the deep learning backbone network. The corresponding model training and testing are carried out, and the obtained evaluation index Accuracy, FP and FN are shown in Table 3.

Table 3. Experimental results on three datasets.

Dataset	Testing	Backbone	Accuracy	FP	FN
DataSet-1	TuSimple	RestNet-18	85.97	0.422	0.317
DataSet-2	TuSimple	RestNet-18	94.02	0.215	0.069
DataSet-3	TuSimple	RestNet-18	95.26	0.196	0.046

As can be seen from Table 3, on the premise of limited device data-carrying capacity, when a higher proportion of linear lanes are added to the complex scene training dataset, the accuracy of the training model tested on the linear lane is significantly improved. Therefore, when deploying and applying in actual scenarios, for the same deep learning network model, adding a certain proportion of a linear lane dataset to the complex scenario will provide better robustness to the structured linear road easily encountered by unmanned drivers. In the self-made image and video detection experiment part in Section 5.2, visualization results will be used to further confirm the conclusions obtained in this part of the experiment.

4.2. Backbone Network Depth Selection

In the previous work, a training dataset with relatively strong adaptability was designed. Next, by modifying the depth of the backbone network, the model was retrained and compared with the impact of using different depths of the backbone network on the accuracy and speed of the model test under the same training dataset. In the experiment, DataSet-1 is used as the training dataset, and ResNet-18 and ResNet-34 are used as the backbone network to conduct the training and testing of the network model. The specific test accuracy and speed are shown in Tables 4 and 5.

Table 4. Experimental results on DataSet-1 with different depth backbones.

Dataset	Backbone	Accuracy	FP	FN
DataSet-1	RestNet-18	85.97	0.422	0.317
DataSet-1	RestNet-34	84.50	0.474	0.373

Table 5. Experimental results of speeds for different depth backbone networks.

Input	Backbone	FPS	Running Time (ms)
288 × 800	RestNet-18	172.38	5.8
288 × 800	RestNet-34	81.431	12.28

The FPS and RunTime tests based on the two deep backbone network models are conducted under NVIDIA GeForce GTX 1060 with Max-Q Design. As can be seen from Tables 4 and 5, when the network depth is deepened, the model is more complex, which not only affects the reasoning speed of the model, but also decreases the test accuracy. Although the effect of ResNet-34 is better than ResNet-18 in theory, due to the problem of datasets, the improvement of model complexity reduces the generalization ability of the model, resulting in overfitting. This shows that the ResNet-18 backbone network model has a better effect, and the method of deepening the network does not apply to the training dataset constructed in this paper.

5. Results and Discussion

This section describes the experimental results and performance analysis of the proposed method. It mainly includes the mixed-attention mechanism experiment and datasets' qualitative detection experiments. To further evaluate the performance of our approach, we test multiple integrated and self-built image/video sequences of onboard cameras on urban, suburban, and rural roads and highways under different environmental conditions, road alignments, and traffic scenarios.

5.1. Mixed-Attention Mechanism Experiment

Based on the conclusions of the above experiments, DataSet-3 is used as the training set and ResNet-18 as the backbone network. This section mainly verifies the lane line detection effect under the mixed-attention mechanism. It adds the mixed-attention mechanism to the ResNet-18 backbone network of the feature extraction module for further comparative experiments. The mixed-attention mechanism is added in front of the first-level residual block of ResNet-18, so that the proposed model pays more attention to important lane line feature information and makes greater use of computing resources. The specific test data of this experiment are shown in Table 6.

Table 6. Experimental results on DataSet-3 under the attention mechanism.

Backbone	Accuracy	FP	FN
RestNet-18	95.26	0.196	0.046
RestNet-18-Attention	95.96	0.196	0.045

It can be seen from Table 6 that after the mixed-attention mechanism is added to the ResNet backbone network, the test accuracy of the network model is increased by 0.7%, which proves the effectiveness of the mixed-attention mechanism ResNet proposed in this paper. Therefore, the final network model of this paper is ResNet-18 with the mixed-attention mechanism as the backbone network, and DataSet-3 as the training dataset.

5.2. Lane Line Detection Function Test

5.2.1. Integrated and Self-Built Image Sets Detection

In order to prove the effectiveness of the training dataset construction and the robustness of the network model design in this paper, Tusimple, Dataset-1 and DataSet-3 are used as the training sets, ResNet-18 is used as the backbone network, and other parameters are set as the same. The network model is trained separately and the visualization qualitative test is carried out on the self-built image sets.

We selected cities, suburbs, villages and highways with different environmental conditions, different road alignments and different traffic scenes, and integrated and self-built four image sets, named SBI-1, SBI-2, SBI-3 and SBI-4, respectively. Among them, the SBI-1 image set mainly covers the scenes of highways, urban straight roads, and simple curves. The SBI-2 image set mainly covers the scenes with lighting at night, reflections at night, and no lighting at night. The SBI-3 image set mainly covers the scenes with pavement damage and pavement shadow, and the SBI-4 image set mainly covers the scenes with slight and severe occlusion.

In the experiment, the newly constructed DataSet-1, DataSet-3 and TuSimple datasets were trained to obtain different network models, and tested on the three image sets. The comparison of the visual results of a typical scene are shown in Figures 6–9. Among them, Figure 6 is a visual comparison under the SBI-1, Figure 7 is a visual comparison under the SBI-2, Figure 8 is a visual comparison under the SBI-3, and Figure 9 is a visual comparison under the SBI-4.

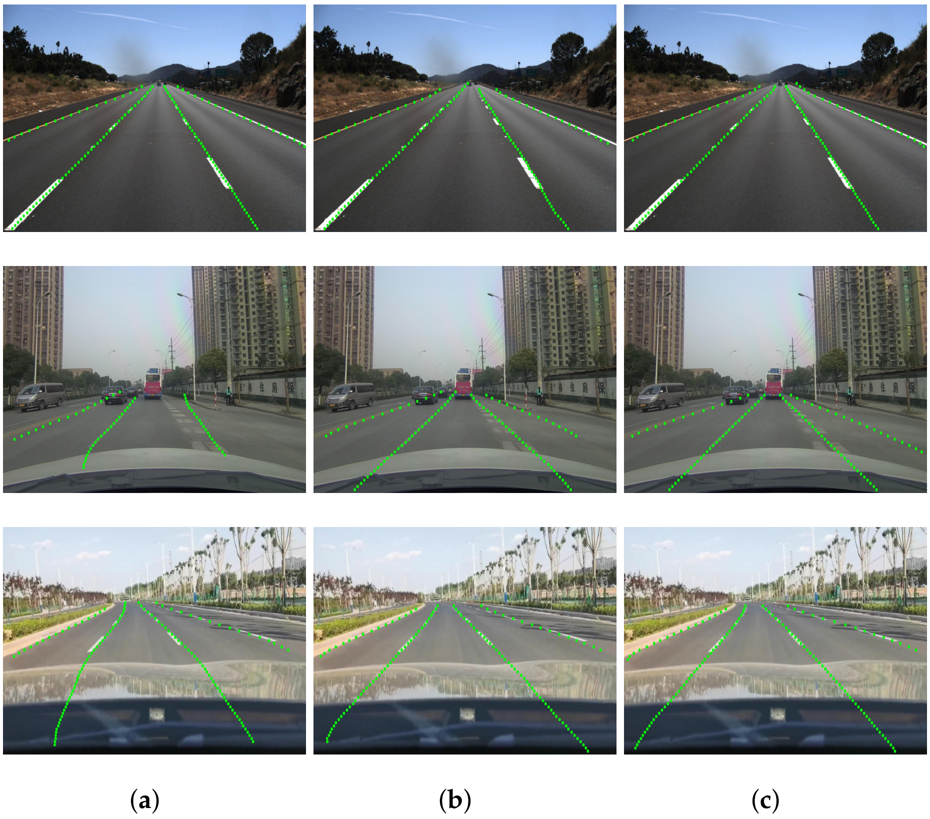


Figure 6. Experimental results on SBI-1 image set. (**a**) TuSimple; (**b**) DataSet-1; (**c**) DataSet-3.

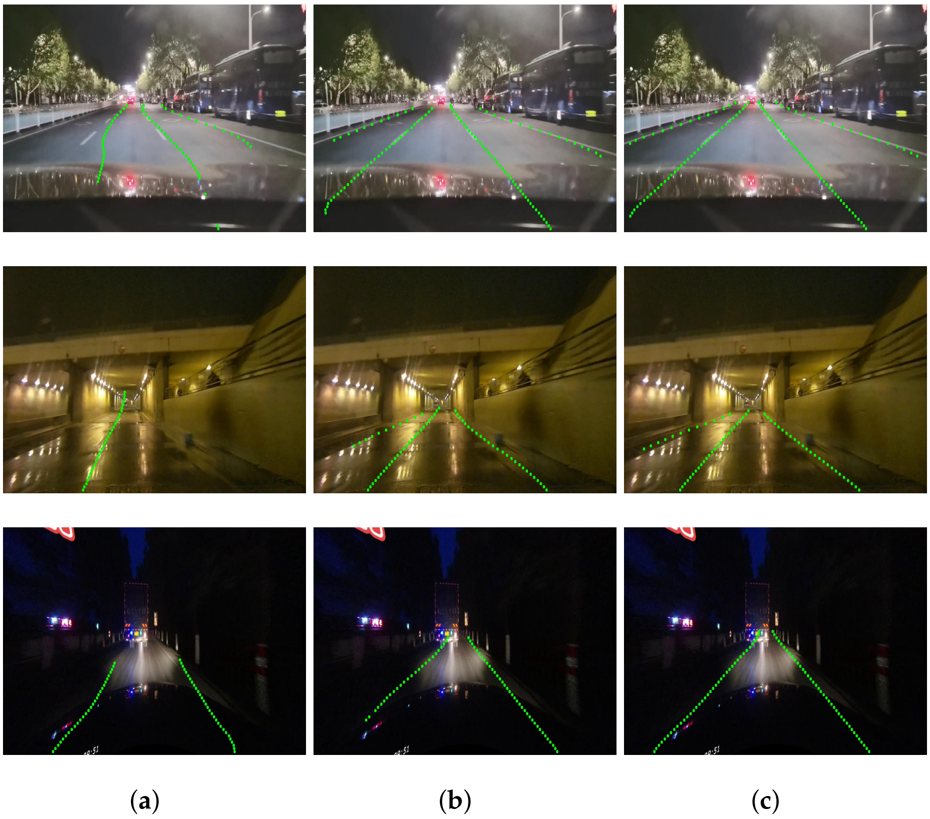


Figure 7. Night experiment results on SBI-2 image set. (**a**) TuSimple; (**b**) DataSet-1; (**c**) DataSet-3.

As can be seen from Figure 6, when only the Tusimple is used to train the model, the test results only have good adaptability to the highway scene, and the effect is poor for some simple urban straight and curved scenes. However, the model trained by DataSet-1 is not as effective as TuSimple and DataSet-3 on highway scenes. In the urban straight and simple curved scenes, the detection results of the DataSet-1 and DataSet-3 training models lead TuSimple by a great advantage. Among them, the model trained by DataSet-1 may misjudge a straight line as a curve, because most of the dataset is curved scenes. By comparison, the training model of the DataSet-3 performs best on highway, urban straight, and simple curved scenes. Therefore, for the highway, urban straight and simple curved scenarios, the comprehensive performance of the model trained by DataSet-3 is the best.

It can be seen from Figure 7 that the model trained by the Tusimple basically loses its function and cannot effectively detect lane lines in the scenes with lighting at night, night reflection, and no lighting at night. The models trained by DataSet-1 and DataSet-3 have achieved good results under the SBI-2, but the DataSet-1 is not as effective as DataSet-3 in solving the problem of no vision. Therefore, for scenes with lighting at night, reflective at night, and no lighting at night, the comprehensive performance of the model trained by DataSet-3 is the best.

As seen from Figure 8, in the scene with obvious illumination changes such as pavement damage and pavement shadow, the model trained by Tusimple is invalid, and the model trained by DataSet-3 has the best test results under SBI-3. DataSet-1 has the condition of missing lane detection and has weak adaptability to complex lighting conditions. Therefore, the comprehensive performance of the model trained by DataSet-3 is the best for the scene of pavement damage, pavement shadow and other obvious illumination changes.

It can be seen from Figure 9 that the model trained by Tusimple has the worst adaptability and the detection fails under the scenarios of slight or severe occlusion. The model trained by DataSet-1 is missing part of the lane lines, and DataSet-1 is not as effective as DataSet-3 in solving the problem of no field of view, and the shadow affects the lane line detection effect. Therefore, for scenarios such as slight occlusion and severe occlusion, the comprehensive performance of the model trained by the DataSet-3 is the best.

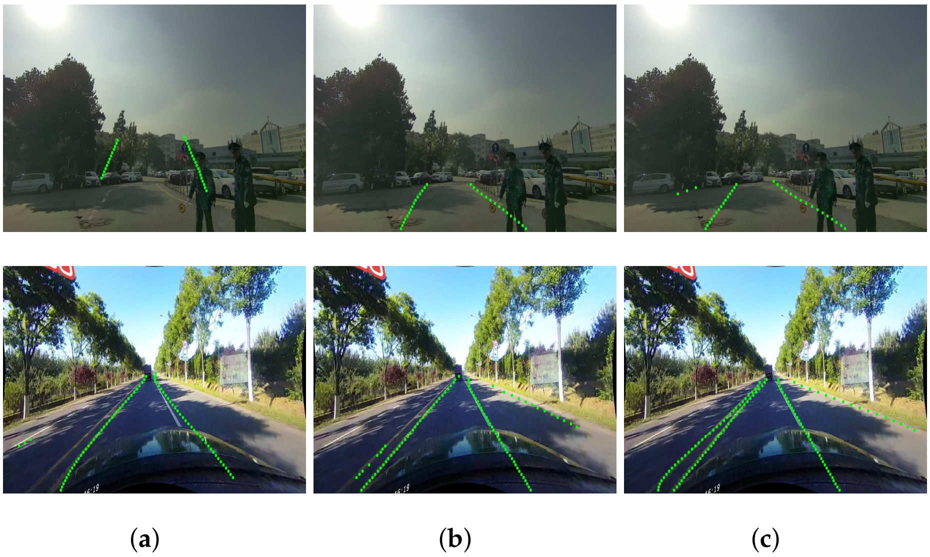

(**a**) (**b**) (**c**)

Figure 8. The experimental results of obvious illumination changes such as pavement damage and pavement shadow on a SBI-3 image set. (**a**) TuSimple; (**b**) DataSet-1; (**c**) DataSet-3.

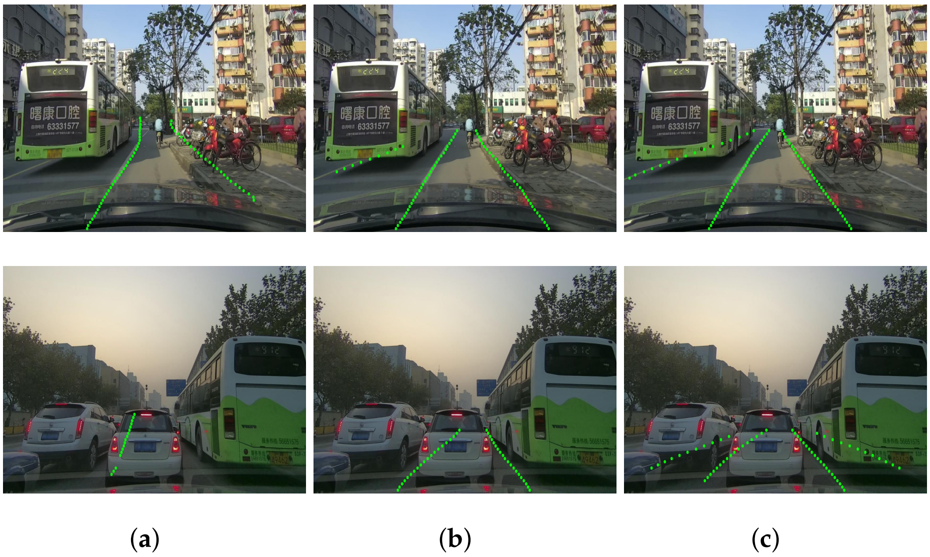

(**a**) (**b**) (**c**)

Figure 9. Experimental results of light occlusion and severe occlusion. (**a**) TuSimple; (**b**) DataSet-1; (**c**) DataSet-3.

In summary, whether in simple or complex scenarios, the training datasets constructed by DataSet-3 can well-detect lane lines under the same network model and parameter conditions, with the strongest universality, especially in complex environments. Therefore, the model training method proposed in this paper can deal with more scenarios and achieve good results.

5.2.2. Self-Built Video Sequence

To further illustrate the effectiveness and practicability of the detection method in this paper, we also drove a car equipped with a camera to collect road videos outdoors. Similarly, urban, suburban, rural areas and highways under different environmental conditions, different road alignments and different traffic scenes were selected, including the lanes under the straight, curved, night-time lighting, night reflection, night no lighting, pavement shadow, pavement damage, vehicle occlusion, and so forth, and multiple video sequences were built by ourselves. DataSet-3 was combined with the ResNet-18 backbone network, which adds a mixed-attention mechanism to train the model, and the testing effect of the road video sequence was analyzed with the trained model.

The detection results of part of the video frames of the straight and curved road scenes during the day are shown in Figure 10.

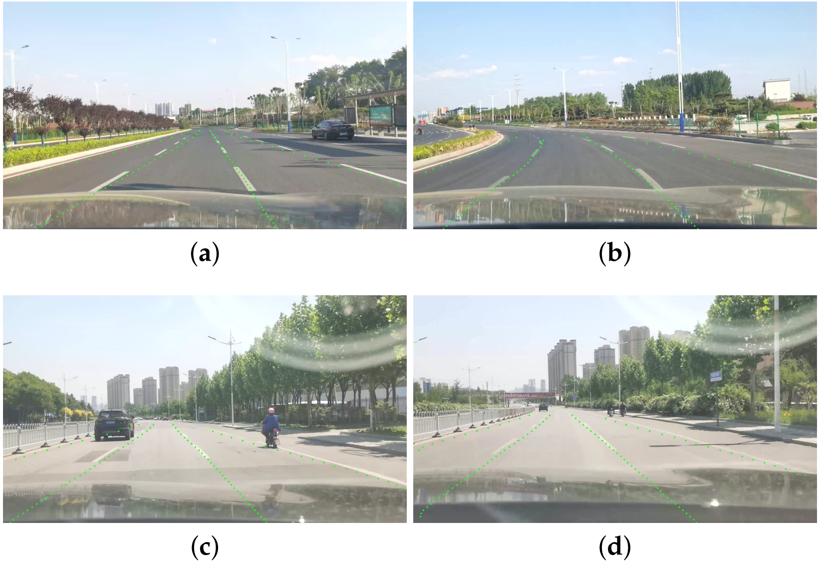

(**a**) (**b**) (**c**) (**d**)

Figure 10. Straight and curved video experiment results. (**a**) Straight; (**b**) Curved; (**c**,**d**) Straight that reflect light.

It can be seen from Figure 10 that our method can accurately detect lane lines when facing the video scene of straight and curved lanes on sunny days. The interference of vehicles and pedestrians in Figure 10c has also been successfully detected, and has strong robustness to the reflection of the front windshield of vehicles. Lane lines can also be detected successfully in no-vision areas under the hood of a vehicle.

The detection results of some frames of the video scenes with lighting, reflection and no lighting at night are shown in Figure 11.

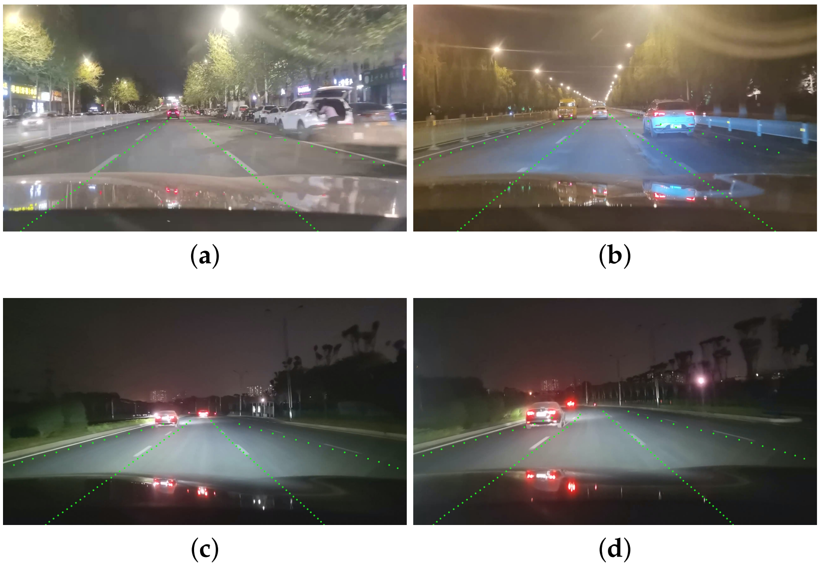
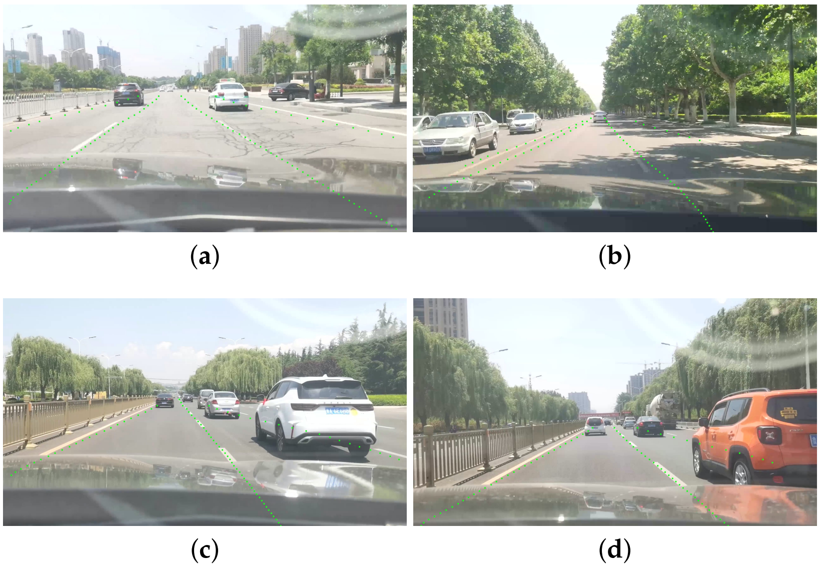

(**a**) (**b**) (**c**) (**d**)

Figure 11. Experiment results with light and dark video at night. (**a**) Lighting at night; (**b**) Reflction at night; (**c**,**d**) No lighting at night.

It can be seen from Figure 11 that there is no problem in detecting videos with lighting, reflection and no lighting at night. Similarly, lane lines can be successfully detected in the area without vision, such as facing the vehicle hood occlusion, and even in the case of no lighting in Figure 11c,d, lane lines can be successfully detected in the curved scene.

The partial frame detection results of the video such as pavement breakage, pavement shadow and vehicle occlusion are shown in Figure 12.

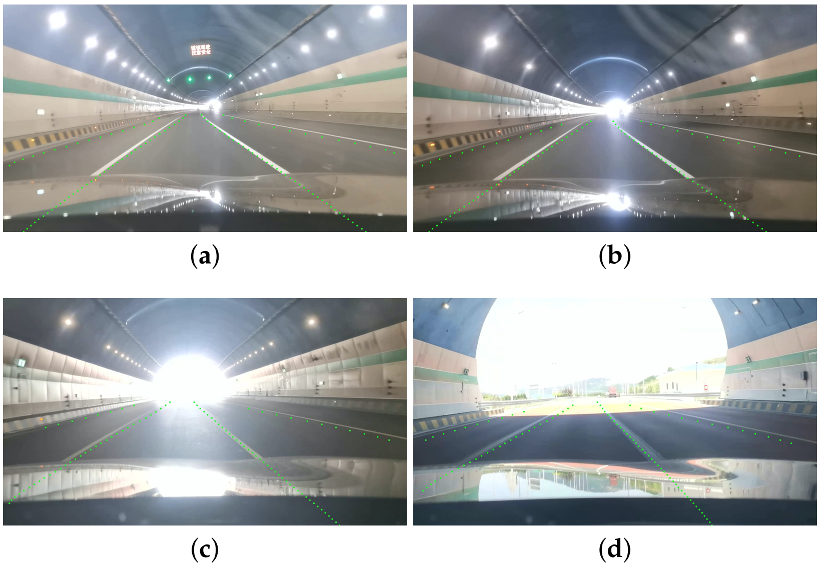

(**a**) (**b**)

(**c**) (**d**)

Figure 12. Experimental results of pavement breakage, shadow, occlusion video.

It can be seen from Figure 12 that Figure 12a is pavement breakage interference that vehicles are more likely to encounter during driving, Figure 12b is pavement shadow interference, and Figure 12c,d is vehicle occlusion interference. Our method has successfully realized the lane line detection of the video sequences with the above interference cases.

Finally, partial detection results of the tunnel scene video are shown in Figure 13.


(**a**) (**b**)

(**c**) (**d**)

Figure 13. The experimental results of the tunnel. (**a**) Straight in tunnel; (**b**) Curved in tunnel; (**c**) White hole effect in tunnel; (**d**) Tunnel exit.

As seen from Figure 13, the proposed method overcomes various disturbances and achieves accurate detection of lane lines, no matter whether it is in the tunnel or at the tunnel mouth, or whether it is affected by light reflection in the tunnel or illumination at the tunnel mouth.

Through the above self-build video test, it is proved that the model and method in this paper can adapt to most of the lane line geographic information conditions encountered by drivers during driving, and have good robustness and universality for lane line detection of video sequences.

In summary, it can be concluded from the above results that the proposed method has the advantages of fast speed, high accuracy, strong robustness, and a good balance between accuracy and efficiency. However, the number of lane lines detected by the proposed method has an upper limit, which needs to be set manually.

6. Conclusions

With the development of deep learning and computer vision, we have used the lane line detection algorithm based on deep learning to provide drivers with accurate geographic auxiliary information, prevent drivers from lane yaw due to inattention, bring a better driving experience to drivers, and reduce the accident rate.

We proposed a lane line detection algorithm based on a mixed-attention mechanism ResNet and row anchor classification. ResNet with identity mapping was used as the backbone network to solve the problem of gradient disappearance in the deep network to improve the model inference effect. It was proposed to add the mixed-attention mechanism after the convolution, normalization and activation layers of the backbone network, respectively, to make the model more focused on important features and improve the pertinence and efficiency of feature extraction. In addition, in order to achieve faster detection speed and solve the problem of no vision, a lane line location selection and classification method based on the row direction was used to detect whether there are lane lines in each candidate point according to the row anchor, reducing the high computational complexity caused by pixel-by-pixel segmentation of traditional semantic segmentation. On the basis of the TuSimple and CurveLane datasets, multi-scene, multi-environment and multi-linear road image datasets and video sequences were integrated and self-built, and several experiments were designed and tested to verify the effectiveness of the proposed method. The proposed method satisfies the safety perception effect of lane geographic information in natural traffic environments, and achieves an effective balance between accuracy and efficiency in actual road application scenarios.

Author Contributions: Conceptualization, Yongchao Song and Xuan Wang; methodology, Tao Huang; software, Jindong Zhao and Tao Huang; investigation, Xuan Wang, Jindong Xu and Xin Fu; formal analysis, Yongchao Song; writing—original draft preparation, Yongchao Song and Tao Huang; writing—review and editing, Yahong Jiang and Xuan Wang; supervision, Weiqing Yan, Jindong Zhao, Yahong Jiang and Xin Fu; funding acquisition, Yongchao Song, Xuan Wang and Jindong Xu. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Natural Science Foundation of Shandong Province (ZR2022QF037, ZR2020QF108, ZR2020MF148, ZR2020QF046), and supported by the Fundamental Research Funds for the Central Universities, CHD (300102342510) and the National Natural Science Foundation of China (62072391, 62066013, 62103350), and China University Industry-University-Research Innovation Fund (New generation information technology innovation projects) under Grant 2021ITA01020.

Data Availability Statement: The datasets are available on Github at https://github.com/TuSimple/tusimple-benchmark (accessed on 20 October 2018); https://github.com/XingangPan/SCNN (accessed on 17 November 2017).

Acknowledgments: We would like to thank the providers of the datasets and the reference authors, as well as the manuscript reviewers.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chen, W.; Wang, W.; Wang, K.; Li, Z.; Liu, S. Lane departure warning systems and lane line detection methods based on image processing and semantic segmentation: A review. *J. Traffic Transp. Eng.* **2020**, *7*, 748–774. [CrossRef]
2. Yu, T.; Huang, H.; Jiang, N.; Acharya, T.D. Study on Relative Accuracy and Verification Method of High-Definition Maps for Autonomous Driving. *ISPRS Int. J. Geo-Inf.* **2021**, *10*, 761. [CrossRef]
3. Yoo, J.H.; Lee, S.W.; Park, S.K.; Kim, D.H. A Robust Lane Detection Method Based on Vanishing Point Estimation Using the Relevance of Line Segments. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 3254–3266. [CrossRef]
4. Du, X.; Tan, K.K. Vision-based approach towards lane line detection and vehicle localization. *Mach. Vis. Appl.* **2016**, *27*, 175–191. [CrossRef]
5. Yi, S.C.; Chen, Y.C.; Chang, C.H. A lane detection approach based on intelligent vision. *Comput. Electr. Eng.* **2015**, *42*, 23–29. [CrossRef]
6. Wang, Q.; Han, T.; Qin, Z.; Gao, J.; Li, X. Multitask Attention Network for Lane Detection and Fitting. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *33*, 1066–1078. [CrossRef]
7. Wang, Y.; Jing, Z.; Ji, Z.; Wang, L.; Zhou, G.; Gao, Q.; Zhao, W.; Dai, S. Lane Detection Based on Two-Stage Noise Features Filtering and Clustering. *IEEE Sens. J.* **2022**, *22*, 15526–15536. [CrossRef]
8. Zhang, J.; Deng, T.; Yan, F.; Liu, W. Lane Detection Model Based on Spatio-Temporal Network with Double Convolutional Gated Recurrent Units. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 6666–6678. [CrossRef]
9. Obradovic, D.; Konjovic, Z.; Pap, E.; Rudas, I.J. Linear fuzzy space based road lane model and detection. *Knowl.-Based Syst.* **2013**, *38*, 37–47. [CrossRef]
10. Choi, H.C.; Park, J.M.; Choi, W.S.; Oh, S.Y. Vision-based fusion of robust lane tracking and forward vehicle detection in a real driving environment. *Int. J. Automot. Technol.* **2012**, *13*, 653–669. [CrossRef]
11. Hou, Y.; Ma, Z.; Liu, C.; Loy, C.C. Learning Lightweight Lane Detection CNNs by Self Attention Distillation. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 7 October–2 November 2019; pp. 1013–1021.
12. Tabelini, L.; Berriel, R.; Paixao, T.M.; Badue, C.; De Souza, A.F.; Oliveira-Santos, T. Keep your eyes on the lane: Real-time attention-guided lane detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 294–302.
13. Xu, H.; Wang, S.; Cai, X.; Zhang, W.; Liang, X.; Li, Z. CurveLane-NAS: Unifying lane-sensitive architecture search and adaptive point blending. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Springer: Cham, Switzerland, 2020; pp. 689–704.
14. Xu, C.; Wu, H.; Zhang, Y.; Dai, S.; Liu, H.; Tian, J. A Real-Time Complex Road AI Perception Based on 5G-V2X for Smart City Security. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 4405242. [CrossRef]
15. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*. [CrossRef] [PubMed]
16. Neven, D.; De Brabandere, B.; Georgoulis, S.; Proesmans, M.; Van Gool, L. Towards end-to-end lane detection: An instance segmentation approach. In Proceedings of the 2018 IEEE intelligent vehicles symposium, Changshu, China, 26–30 June 2018; pp. 286–291.
17. Zhang, Y.; Lu, Z.; Ma, D.; Xue, J.H.; Liao, Q. Ripple-GAN: Lane Line Detection with Ripple Lane Line Detection Network and Wasserstein GAN. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 1532–1542. [CrossRef]
18. Liu, R.; Yuan, Z.; Liu, T.; Xiong, Z. End-to-end lane shape prediction with transformers. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 3–8 January 2021; pp. 3694–3702.
19. Tabelini, L.; Berriel, R.; Paixao, T.M.; Badue, C.; De Souza, A.F.; Oliveira-Santos, T. Polylanenet: Lane estimation via deep polynomial regression. In Proceedings of the 2020 25th International Conference on Pattern Recognition, Milan, Italy, 10–15 January 2021; pp. 6150–6156.
20. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
21. Li, Q.; Zhou, J.; Li, B.; Guo, Y.; Xiao, J. Robust Lane-Detection Method for Low-Speed Environments. *Sensors* **2018**, *18*, 4274. [CrossRef] [PubMed]
22. Fernando, S.; Udawatta, L.; Horan, B.; Pathirana, P. Real-time Lane Detection on Suburban Streets Using Visual Cue Integration Regular Paper. *Int. J. Adv. Robot. Syst.* **2014**, *11*, 61. [CrossRef]
23. Lee, C.; Moon, J.H. Robust Lane Detection and Tracking for Real-Time Applications. *IEEE Trans. Intell. Transp. Syst.* **2018**, *19*, 4043–4048. [CrossRef]
24. Yoo, H.; Yang, U.; Sohn, K. Gradient-enhancing conversion for illumination-robust lane detection. *IEEE Trans. Intell. Transp. Syst.* **2013**, *14*, 1083–1094. [CrossRef]
25. Wei, X.; Zhang, Z.; Chai, Z.; Feng, W. Research on lane detection and tracking algorithm based on improved hough transform. In Proceedings of the 2018 IEEE International Conference of Intelligent Robotic and Control Engineering (IRCE), Lanzhou, China, 24–27 August 2018; pp. 275–279.
26. Mammeri, A.; Boukerche, A.; Tang, Z. A real-time lane marking localization, tracking and communication system. *Comput. Commun.* **2016**, *73*, 132–143. [CrossRef]

27. Zhao, K.; Meuter, M.; Nunn, C.; Müller, D.; Müller-Schneiders, S.; Pauli, J. A novel multi-lane detection and tracking system. In Proceedings of the 2012 IEEE Intelligent Vehicles Symposium, Madrid, Spain, 3–7 June 2012; pp. 1084–1089.
28. He, P.; Gao, F.; Wei, H. Study on curved Lane Detection Using Catmull-Rom Spline. *Chin. J. Automot. Eng.* **2015**, *5*, 276–281.
29. Lee, M.; Jang, C.; Sunwoo, M. Probabilistic lane detection and lane tracking for autonomous vehicles using a cascade particle filter. *Proc. Inst. Mech. Eng. Part D J. Automob. Eng.* **2015**, *229*, 1656–1671. [CrossRef]
30. Tang, J.; Li, S.; Liu, P. A review of lane detection methods based on deep learning. *Pattern Recognit.* **2021**, *111*, 107623. [CrossRef]
31. Zou, Q.; Jiang, H.; Dai, Q.; Yue, Y.; Chen, L.; Wang, Q. Robust lane detection from continuous driving scenes using deep neural networks. *IEEE Trans. Veh. Technol.* **2019**, *69*, 41–54. [CrossRef]
32. Zhao, Z.; Wang, Q.; Li, X. Deep reinforcement learning based lane detection and localization. *Neurocomputing* **2020**, *413*, 328–338. [CrossRef]
33. Qin, Z.; Wang, H.; Li, X. Ultra fast structure-aware deep lane detection. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Springer: Cham, Switzerland, 2020; pp. 276–291.
34. TuSimple: Tusimple Benchmark. Available online: https://github.com/TuSimple/tusimple-benchmark (accessed on 20 October 2018).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.