ITC 1/54
Information Technology and Control
Vol. 54 / No. 1/ 2025
pp. 32-43
DOI 10.5755/j01.itc.54.1.38802

Occluded Lane Line Detection with Deep Polynomial Regression in Global View

Received 2024/09/13 | Accepted after revision 2024/12/16

HOW TO CITE: Ren, S. (2025). Occluded Lane Line Detection with Deep Polynomial Regression in Global View. *Information Technology and Control*, 54(1), 32-43. https://doi.org/10.5755/j01.itc.54.1.38802

Occluded Lane Line Detection with Deep Polynomial Regression in Global View

Shuman Ren
School of Art, Design and Media, East China University of Science and Technology, Shanghai 200030, China; e-mail: 2478746826@qq.com

Corresponding author: 2478746826@qq.com

Occluded Lane line detection method based on depth polynomial regression in global field of view is proposed for the problem of lane lines being obscured on driving road. In order to obtain better lane line feature representation capability, a dual attention mechanism module that connects spatial attention and channel attention in series is introduced to improve the network's ability to obtain lane line features, and then its feature information is used to adopt the lane line detection method of line-direction position classification by adding a line-by-line detection branch after the VGG backbone network to search lane line pixel points through line-direction scanning; in order to distinguish the lane line In order to distinguish which lane line the pixel points belong to, a loss function is designed according to the idea of metric learning, and a vector block is introduced on the semantic segmentation network to record the vector distance of the lane line pixels; finally, the pixels on the current lane line are extracted by the OPTICS clustering model, and a depth polynomial approach is used to complete the fitting of the lane line. Experiments are conducted on the Tusimple dataset, and the results show that compared with the LaneNet network, the method in this paper improves 4.79% and 6.34% in accuracy and precision, respectively, and has a better detection effect on the obscured lane lines.

KEYWORDS: Lane Obstruction Detection, Instance Segmentation, Clustering, Deep Polynomial Fitting, Vehicle Departure Warning.

1. Introduction

With the rapid development of the economy and society, road traffic safety is increasingly being emphasized. Every year, more than one million people die in traffic accidents worldwide, and among these, the proportion of deaths directly or indirectly caused by vehicle loss of control or instability deviating from the intended route is as high as 50%. Statistics from the Chinese Ministry of Transport show that traffic

accidents caused by vehicles deviating from normal lanes account for about 50% of the total, and lane departure is also considered to be a major contributing factor to rollover accidents involving motor vehicles. Therefore, people have been exploring various methods to improve the accuracy of Lane Departure Warning (LDW) systems, assist drivers in safe driving, and reduce the occurrence of traffic accidents [1, 2, 4, 11]. Lane detection, as a fundamental task in lane departure warning systems, directly affects the accuracy of warnings. In this paper, the image information on the road is input into a lane detection network, which goes through segmentation, fitting, and other processes to detect lane lines, aiming to assist drivers in safe driving.

Early lane line detection primarily used traditional image processing detection algorithms. Anbalagan et al. [1] divided the input image into different sub-regions along the vertical direction. Within these sub-regions, a local line extraction method was employed to extract candidate lane lines, and finally, a dynamic programming method was used to find the lane line that best met the requirements. Later, with the rise of machine learning, many began to utilize classifiers to distinguish between lane lines and the background for lane line detection. Ghanem et al. [4] proposed a machine learning method based on Real-Adaboost, training a linear classifier on the appearance and edge cues of training samples, and ultimately making a final decision on the lane line area by analyzing the consistency of the slope direction in the lane line region. The final correct detection rate can reach around 92%. In recent years, lane line detection has mainly been based on deep learning methods. Shen [8] proposed a new method called LaneNet that transforms the detection problem into a segmentation problem. This network consists of an encoder for feature extraction and a decoder for reconstructing the lane lines. At the same time, Neven designed a lightweight network called HNet that autonomously learns the inverse perspective transformation matrix, allowing images to be converted to a top-down view, resulting in better fitting outcomes on the top-down view. Li et al. [5] proposed a SCNN lane line detection method, which slices the feature maps output by the CNN network in both the row and column directions, and then performs convolution operations separately. SCNN indirectly achieves the goals of increasing the receptive field and enhancing global information, making it particularly suitable for detecting and segmenting long and continuous shapes. Although deep learning-based lane line detection methods have achieved significant success, they need to operate in real-time during lane line detection. However, existing deep learning approaches cannot meet this requirement due to their large computational load. Additionally, current methods do not address the issue of other vehicles obstructing lane lines, leading to suboptimal detection results in scenarios where lane lines are obscured by vehicles in heavy traffic.

In response to the above issues, this article focuses on detecting obscured lane lines and proposes a method for detecting obscured lane lines based on deep polynomial regression with a global view. This method uses a VGG feature extraction network with an attention mechanism and a fully connected layer as the main framework. It traverses the image in the row direction based on the global features of the lane lines to select the lane positions. Based on the pre-selected lane position information, it integrates the results of lane line instance segmentation to fit the lane line pixels and achieve the detection of obscured lane lines.

2. Overall Research Methods

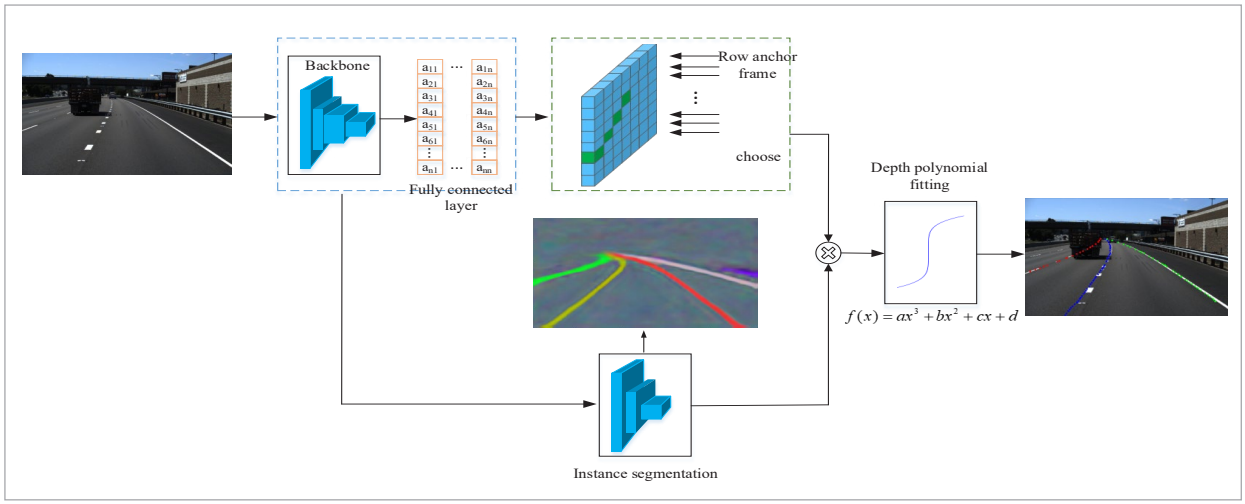
The research method of occluded lane line detection based on deep polynomial regression with a global perspective is shown in Figure 1, mainly including lane line feature extraction network, lane line instance segmentation network, and lane line fitting.

The lane line image extracts lane line features through a deep learning framework, while drawing lane lines using an instance segmentation network. By applying deep polynomial fitting to the lane lines, the obtained lane line features are traversed on the original image pixel by pixel. Finally, the results of the instance segmentation are integrated to complete the detection of occluded lane lines.

2.1. VGG Feature Extraction Network with Attention Mechanism

When a vehicle is in motion, the lane lines are continuous in space and time, meaning that the possibility of the same object changing its motion state, shape, and area between adjacent images is relatively small. Ordinary VGG networks pay less attention to features

Figure 1
Overall framework



in this scenario. To address this issue, this paper introduces a VGG feature extraction network with an attention mechanism to extract lane line features. By incorporating a Convolutional Block Attention Module (CBAM) that combines channel attention and spatial attention to enhance feature representation, the feature extraction capability is improved. Adding the CBAM module to the VGG classification network significantly enhances classification performance [7]. Therefore, this paper introduces the CBAM module into the lane feature extraction network to enhance the network's ability to extract lane line features.

To determine whether a pixel belongs to a certain lane, it is necessary to consider the feature map information of different channels. Since each channel has different importance depending on the task, channel attention enhances the ability to determine important feature channels by assigning different weights to each channel for each task. Spatial attention is responsible for capturing global contextual information and can infer the information of hidden pixels [7] by combining the position information of all pixels on the lane line and the pixels on the feature map. The working principles of channel and spatial attention modules are as follows [2, 4, 11].

As shown in Figure 2, input features, using max pooling and average pooling operations to transform the features into max-pooled features and average-pooled features, and sending these two results to a multi-layer perceptron (MLP). After ReLU activation, two activated results are obtained, then these two output results are element-wise summed to give the final output result. The formula for the channel attention mechanism is:

$$\begin{aligned} &M_c(F) \\ &= \sigma(MLP(AvgPool(F)) + MLP(MaxPool(F))) \\ &= \sigma(\omega_1(\omega_0(F^c_{avg})) + \omega_1(\omega_0(F^c_{max}, \end{aligned} \tag{1}$$

where, σ represents the sigmoid function, ω_0 and ω_1 represents the weights of the MLP.

As shown in Figure 3, the feature map F after passing through the channel attention mechanism module is used as input, and then sent to the pooling layer to generate average pooling feature map F^S_{avg} and max pooling feature map F^S_{max}, The purpose of the two pooling operations is to highlight the occluded lane line features [9]; then overlay F^S_{avg} and F^S_{max}, generate a 2-channel feature map F^S_{concat}; finally F^S_{concat} sequentially pass through 7×7 standard convolution and sigmoid activation function to generate spatial attention feature map $M_S(F)$. The calculation formula for the generated spatial attention feature map after processing is:

$$\begin{aligned} &M_S(F) \\ &= \sigma(f^{7\times7}([AvgPool(F); MaxPool(F)])) \\ &= \sigma(f^{7\times7}([F^S_{avg}; F^S_{\max}])) \\ &= \sigma(f^{7\times7}(F^S_{concat})) \end{aligned} \tag{2}$$

Figure 2
Channel attention

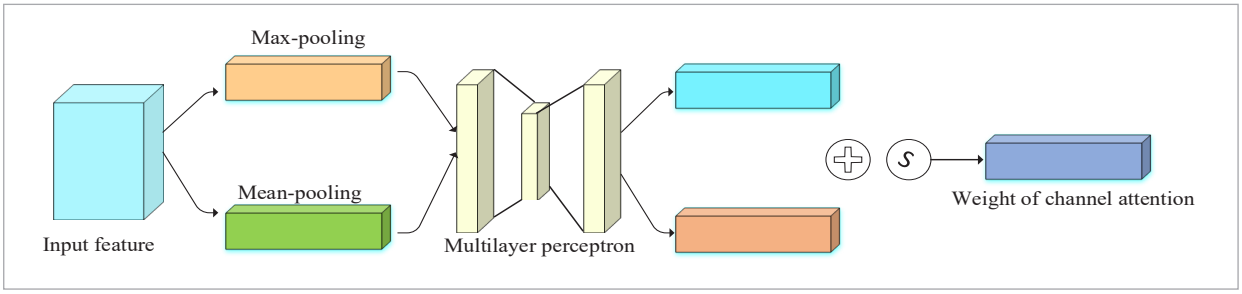
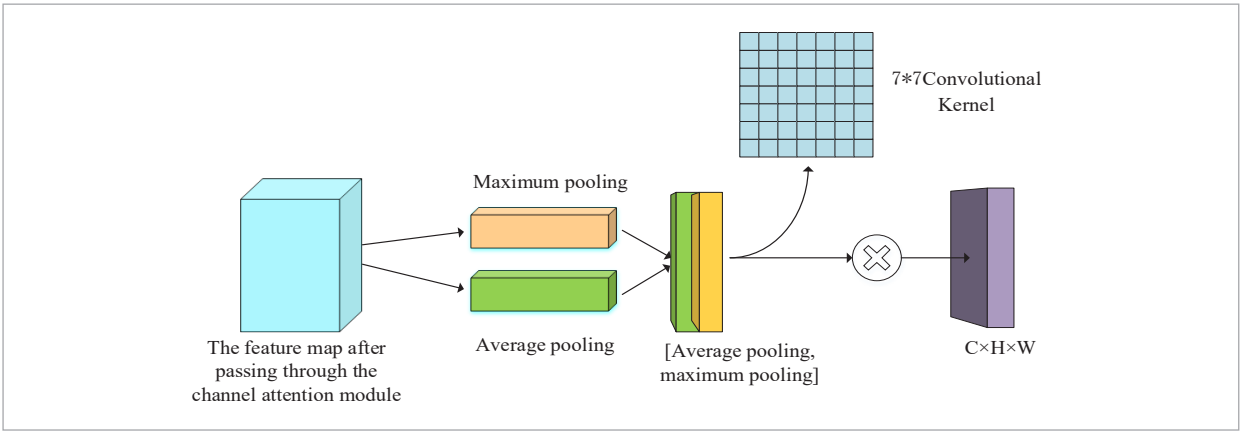


Figure 3
Spatial attention

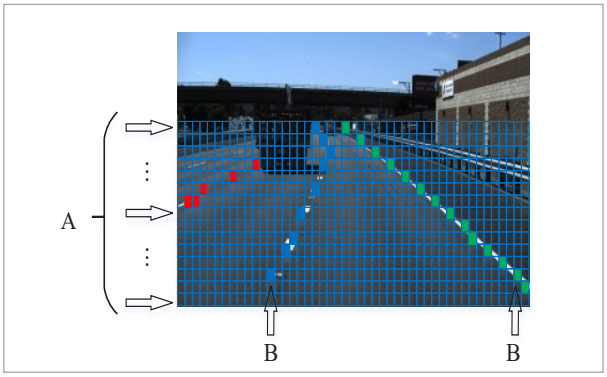


Among them, $M_S(F)$ is spatial attention feature map, F is feature map after the channel attention mechanism module, average pooling feature map F^S_{avg}, and maximum pooling feature map F^S_{max}.

2.2. Classification Based on Row Direction Position

For image segmentation of images containing lane line information, when selecting units containing lane markings, it is necessary to scan each pixel in the image and judge based on the extracted lane line features. Lane line pixels are retained if they belong to the lane line, otherwise they are discarded. However, since lane line information occupies a very small proportion of the entire image, scanning pixel by pixel would increase the processing of irrelevant pixel data significantly. Therefore, using image segmentation for processing results in a large computational load and slow inference speed. This paper considers lane line detection as a relatively small computational load line classification problem. As shown in Figure 4, the lane line image is first divided into grids, and then the lane line features extracted by the VGG feature extraction network with an attention mechanism are used to select one row of grids at a time, choosing the

Figure 4
Row direction position classification



unit most likely to contain lane markings. This approach not only improves the speed of lane line detection but also allows for better selection of lane markings from a global perspective.

The horizontal arrow at "A" represents the row anchor direction, while "B" refers to the unit module containing the pixel points of lane lines in the image. Assuming the height of the input image is H the width is W, and there are C lane lines in the image. If classifying the image pixel by pixel, the calculation complexity is $P_1 = H \times W \times C$. If dividing the lane line portion of the image into $m \times n$ grid units, the calculation complexity for classifying lane line pixel points in a single image is $P_2 = m \times n \times C$. Due to $P_1 \ll P_2$, the calculation complexity is lower when classifying based on the row direction, which can improve lane line detection speed [6].

2.3. Instance Segmentation

In the lane departure warning system, it is necessary to accurately obtain the position of the vehicle relative to the current lane. In order to obtain this relative position, accurate detection of the lane lines of the current lane is particularly important. Due to the characteristics of lane lines being thin and curved, using deep learning object detection algorithms based on detection boxes may select many irrelevant information. Therefore, this method is not very adaptable for lane line detection [13]. The introduction of instance segmentation algorithms has solved this problem. Compared to object detection algorithms based on detection boxes, instance segmentation algorithms can not only obtain the outer contour information of each lane line, but also use different IDs to distinguish lane lines belonging to different lanes, achieving true pixel-level lane line detection [13]. In order to identify which lane each pixel point on the lane line belongs to, this paper uses an instance segmentation algorithm based on metric learning for lane line detection. The method can be described as follows: introducing a vector module in the semantic segmentation network to record the vector distance from the current lane line pixel point to the surrounding lanes. Before starting the detection, it is necessary to initialize a vector element foreach lane line pixel point to be detected. Then, based on the metric concept, a loss function is designed to calculate the vector distance. If the vector distance from the detected lane line pixel point to the current lane is the smallest, it is considered to belong to the current lane, and vice versa.

This part of the loss function is mainly composed of three parts: variance $Loss(L_{var}$, distance $Loss(L_{dist})$, and the average of the mean of different lane lines $Loss(L_{reg})$.

$$L \frac{1}{C}\sum_{c=1}^{C}\frac{1}{N_c}[\|\mu_c - x_i\| - \delta_v]^2_{var} \quad (3)$$

$$L_{dist} = \frac{1}{C(C-1)}\sum_{cA=1}^{C}\sum_{cB=1,cA\neq cB}^{C}[\delta_d - \|\mu_{cA} - \mu_{cB}\|]^2 \quad (4)$$

$$L_{reg} = \frac{1}{C}\sum_{c=1}^{C}\|\mu_c\| \quad (5)$$

$$L = Ldist_{reg_{var}}. \quad (6)$$

Among them, C is the number of lane lines, N_c is the number of pixels in the same lane line, mean μ represents the mean vector of the lane line, x_i is the vector of each pixel, δ is the hyper parameter [13], and L is the final loss function.

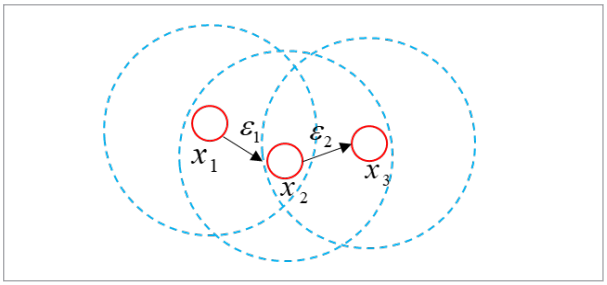
Through the above design, not only can the VGG feature extraction network with attention mechanism be used to accurately extract the feature information of lane lines, obtain lane line pixel points through row-wise search, but also the instance segmentation method with vector module can be introduced to group pixels on the same lane line together, completing the distinction of which lane the lane line pixel points specifically belong to.

2.4. Lane Line Fitting

Through the processing of the above model, visible pixels on each lane line can be obtained. However, in order to achieve accurate detection of occluded lane lines and predict occluded areas, it is necessary to fit the lane lines and predict the occluded parts. This paper uses OPTICS (Ordering Points to Identify the Clustering Structure, OPTICS) clustering to cluster the segmented pixels, grouping pixels of the same category together to form a lane line. Compared to DBSCAN clustering, OPTICS clustering is less sensitive to input parameters and has better adaptability. However, both methods require setting two parameters, denoted as $\varepsilon_i(0 \le \varepsilon_i \le \varepsilon)$ and $Minpts$. Parameter $\varepsilon_i(0 \le \varepsilon_i \le \varepsilon)$ is the minimum neighborhood radius for the current lane line pixels to become core points.

The blue dashed circle represents the neighborhood formed by the current lane line pixels as core points with radius $\varepsilon_i(0 \le \varepsilon_i \le \varepsilon)$.To address the issue of clustering effectiveness being affected by different pixel densities, OPTICS clustering dynamically sets the neighborhood radius $\varepsilon_i(0 \le \varepsilon_i \le \varepsilon)$ as a parameter within a range to adapt to clustering of pixels with different densities. Once the neighborhood is determined, lane line pixels falling within the neighborhood are considered core objects, The red solid circle x_1 shown in Figure 5 is the core object, with their quantity denoted as $Minpts$ [5]. When clustering lane line pixels, this paper only considers core objects within the neighborhood, treating pixels outside the neighborhood as noise points that do not participate in the lane line fitting process.

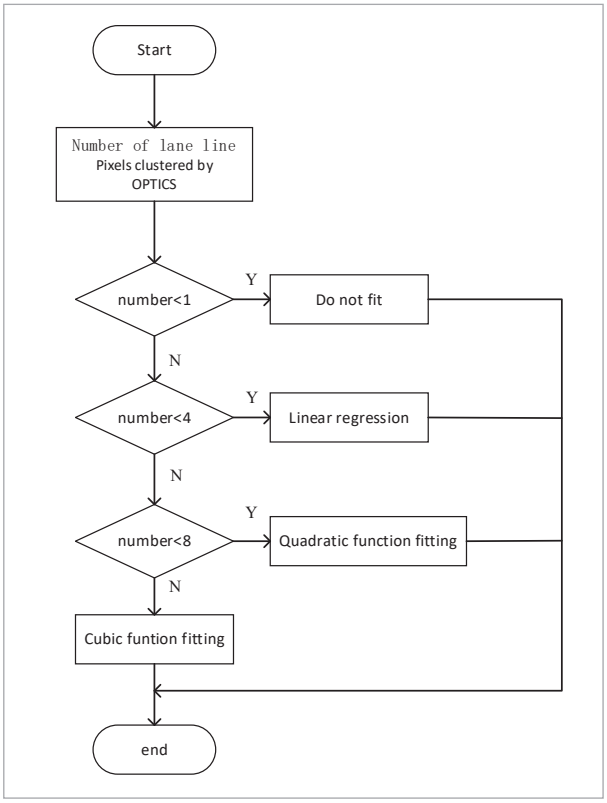
Figure 5
OPTICS clustering process



After clustering is completed, fitting of the pixels will then begin. Due to the diversity of lane line types, fitting straight lane lines is efficient, but fitting curved lane lines (such as curves, corners, etc.) will result in very inaccurate fitting. Quadratic curves can fit curved lane lines with small curvature, but for lane lines with larger curvature, the fitting process will introduce significant errors. On the other hand, cubic curve fitting has high accuracy for fitting multiple pixels, but the fitting process is relatively time-consuming. Therefore, this paper combines these several types of deep polynomials to fit the lane lines, as shown in the fitting process in Figure 6.

This article uses the OPTICS clustering method to cluster lane line pixels. If there are no other pixels in the neighborhood of the current lane pixel except itself, no fitting is performed. If the number of lane line pixels in the neighborhood is in the range of [1,4), a first-degree polynomial is used to fit the lane

Figure 6
Lane Line Fitting



line pixels. If the number of lane line pixels in the neighborhood is in the range of [4,8), a second-degree polynomial is used for fitting. If the number of lane line pixels in the neighborhood after clustering is greater than 8, a third-degree polynomial is used for fitting.

3. Experiment and Result Analysis

3.1. Experimental Dataset and Evaluation Metrics

To verify the detection performance of the method in this article, the Tusimple dataset was used for testing. The Tusimple dataset is an open-source dataset about lane lines released by Tusimple, which includes lane line data under various occlusion conditions. The lane lines annotated in the Tusimple dataset are in JSON format, meaning that when initially annotat-

ing the lane line dataset, a series of continuous points were used to describe the lane lines. In this study, these continuous annotated points were connected to express the lane lines for training purposes.

In terms of lane detection, this article evaluates the detection of occluded lane lines based on accuracy, false positive rate (FPR), and false negative rate (FNR) provided by the Tusimpe dataset, as well as precision and recall.

3.2. Road Image Preprocessing

During the process of vehicle driving, various factors such as actual road conditions and sudden changes in lighting can cause different degrees of interference in the collection and transmission of lane image information, affecting the observable information of the target objects in the collected images. This article preprocesses the collected road image information through image smoothing and enhancement techniques to reduce the complexity of lane line detection algorithms and improve the efficiency of the algorithms.

The road images studied in this article are mainly affected by salt-and-pepper noise originating from the CMOS image sensor, which is prone to interference in its circuit structure during signal acquisition or transmission. Salt-and-pepper noise, also known as impulse noise, is a common type of noise in images, characterized by the appearance of black and white pixels interspersed in the image, where some pixels are randomly changed to the minimum or maximum values. Therefore, using median filtering can effectively remove salt-and-pepper noise, smooth the acquired images, suppress interference, and enhance the system's detection and recognition capabilities.

If we consider salt-and-pepper noise as a variable in the image and represent it with variable z, then the probability density function of z satisfies Equation 7:

$$p(z)=\begin{cases}P_a & z=a\\ P_b & z=b\\ 0 & \text{other}\end{cases} \tag{7}$$

Among them, $0 \le P_a \le 10 \le P_b \le 1$.The noise points on the image are respectively $z=a$ and $z=b$.

Median filtering is a common nonlinear filtering method. Its filtering principle is to select apixel in the captured image, determine its neighborhood range, arrange all pixel values in the neighborhood in order, then select the middle value from the arranged values, and use this middle value to replace the originally selected pixel value. For example, if the selected pixel value is x_1 and the pixel values in its neighborhood are $x_2, x_3, \ldots, x_n$, then the specific steps of median filtering to eliminate salt-and-pepper noise are:

1 Select the size of the filter template;

2 Arrange the data in the template in descending order, that is, arrange $x_1, x_2, x_3, \ldots, x_n$ in $y_1, y_2, y_3, \ldots, y_n$.

3 Calculate the median Mid using formula 8.

$$Mid=\begin{cases}y_{\frac{n+1}{2}} & n \text{ is an odd number}\\ \frac{1}{2}\left(y_{\frac{n}{2}}+y_{\frac{n+1}{2}}\right) & n \text{ is an even number}\end{cases} \tag{8}$$

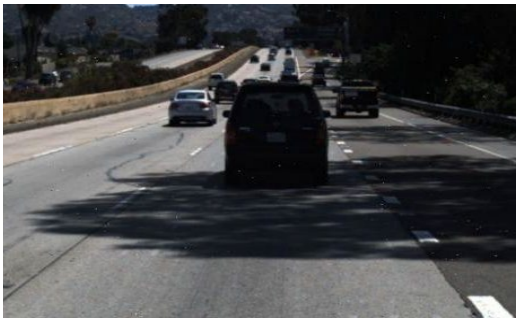
4 Replace the grayscale value at the center position of the template with the median value Mid, that is, the value x_1.

In order to better study the performance of median filtering method in handling salt-and-pepper noise, a common salt-and-pepper noise density is 0.05. As shown in Figure 7(a), the result after processing with a square window using median filtering template is shown in Figure 7(b). From the images, it can be seen that after smoothing with median filtering, noise in the image can be effectively filtered out, while the edge features of the image can be well preserved.

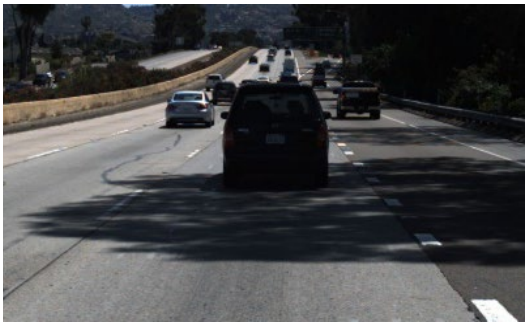
After filtering out noise in the image, when driving in actual conditions, situations such as entering and exiting tunnels, encountering fog patches ahead, and significant changes in light intensity within the driver's line of sight can lead to the foreground and background of the captured image being either too dark or too bright. Therefore, image enhancement is performed on the captured image information to improve the clarity of the image for detecting lane markings that may be obstructed.

Currently, common methods for image enhancement include histogram equalization, adaptive histogram equalization, and contrast limited adaptive histogram equalization. Histogram equalization transforms the grayscale histogram of the original image into a uniform distribution across the entire grayscale range to enhance the image, but it may not handle areas that are too bright or too dark well. Adaptive histogram equalization mainly equalizes local histograms, performing histogram equalization within selected

Figure 7
Image smoothing results

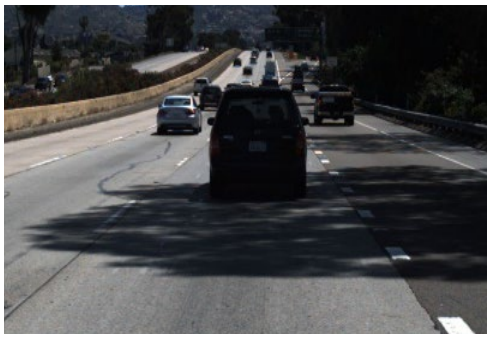


(a) Original image with pretzel noise

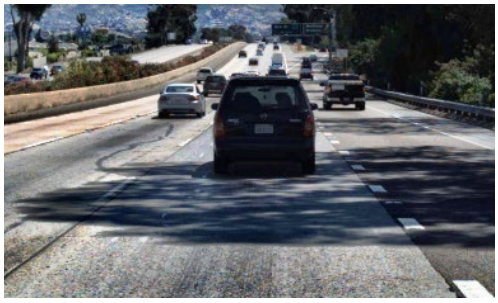


(b) Median filter smoothing

Figure 8
CLAHE algorithm processing results



(a) Original image



(b) CLAHE processing

rectangular regions. However, when the pixel values in the selected rectangular region are highly similar, there may be a defect of excessive enhancement in the region. Therefore, to address significant changes in brightness within the driving line of sight range when entering or exiting tunnels or encountering fog ahead, this paper improves the Contrast Limited Adaptive Histogram Equalization (CLAHE) algorithm. This method enhances lane detection in the global field of view by limiting the contrast amplitude based on adaptive histogram equalization, effectively overcoming the defect of excessive enhancement in regions.

Figure 8 shows the result after processing with the CLAHE algorithm. From 8(b), it can be seen that the Contrast Limited Adaptive Histogram Equalization (CLAHE) algorithm can significantly enhance the darker areas of the image without overly enhancing the brighter areas. The overall information of the image of obstructed lane lines is also well preserved.

3.3. Experiment Results and Analysis of Lane Detection

After completing the design of the lane detection algorithm, experiments on lane detection were conducted in the experimental environment as shown in Table 1.

To address the issue of lane markings being obscured on the road, in order to improve the network's ability to

Table 1
Experimental environment

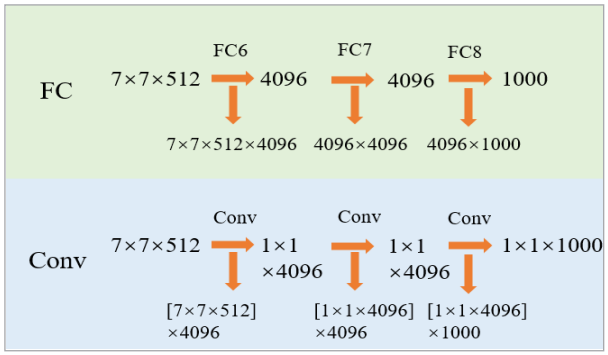
Configuration item	Configuration parameters
Memory	64G
CPU	Intel(R) Xeon(R) E5-2680 v4
GPU	NVIDIA GeForce RTX2080ti
Video memory	11G
Deep learning framework	PyTorch
Programming language	Python3.7

extract lane marking features, a dual attention mechanism module is introduced that concatenates spatial attention and channel attention. By incorporating attention mechanisms into the VGG feature extraction network to obtain lane marking feature information, the Convolutional Block Attention Module (CBAM) that combines channel attention and spatial attention is used to enhance feature representation capabilities. Utilizing the VGG network model for lane marking feature extraction ensures a larger receptive field, making it easier to capture changes in the image and bring about greater local information diversity. This approach also enables better description of features such as road lane edges and texture details.

However, due to the large number of computational parameters in the 3 fully connected layers of the VGG network structure, it will affect computational efficiency. Therefore, by replacing the fully connected layers in the VGG19 network with convolutional layers, it is possible to improve processing speed by reducing the number of parameters, while also preserving the spatial information of the images. The specific process is shown in Figure 9.

Figure 9

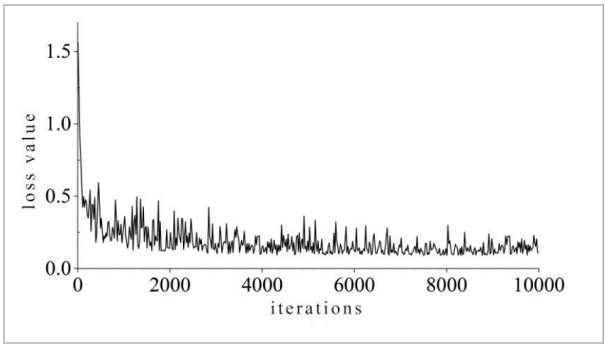
Convolutional replacement of the fully connected process



In order to distinguish which lane a pixel point on the lane line belongs to, a loss function was designed based on the concept of metric learning. In the process of improving the network training, this paper set the batch size to 2, used stochastic gradient descent optimizer, set the momentum factor to 0.95, set the weight decay factor to 0.0001, and the iteration number for network training was set to 10000 times. Figure 10 shows the relationship between the loss function curve and the number of iterations during the learning process. It can be seen that as the number of iterations increases, the loss value gradually decreases, eventually leveling off and stabilizing around 0.1.

Figure 10

Plot of loss value change during training



The current lane pixels are extracted using the OPTICS clustering model, and the lane lines are fitted using a deep polynomial method. In order to verify the lane detection performance of this method, this paper designs ablation experiments to verify the effectiveness of various improvements, with results shown in Table 2.

Table 2

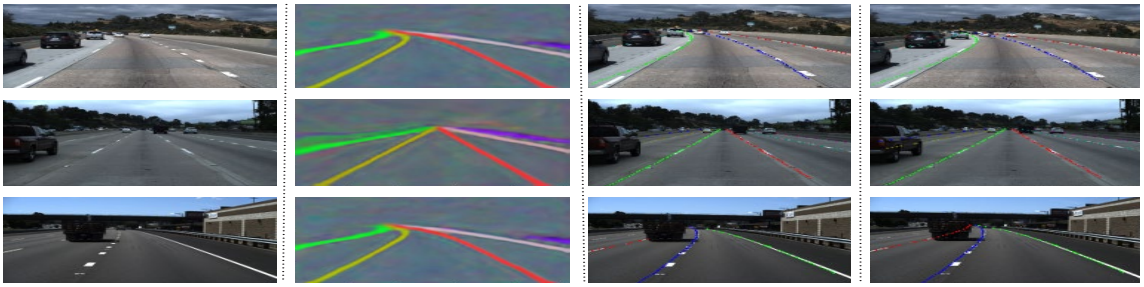
Results of ablation experiments

Serial number	CBAM	OPTICS	mAP%
1	×	×	89.7
2	√	×	92.3
3	×	√	91.9
4	√	√	94.3

Among them, Convolutional Block Attention Module (CBAM) is a lightweight convolutional attention module; Ordering points to identify the clustering structure (OPTICS) is a density based polymer; The mAP represents mean average precision; "×" indicates not using this improvement method; "√" indicates using this improvement method.

As shown in Table 2, introducing the CBAM module in the feature extraction part and using the OPTICS clustering model have both improved the detection performance. By incorporating the CBAM module, which integrates attention weights in both channel and spatial dimensions, a larger receptive field is achieved, enabling better exploration of occluded lane

Figure 11
Obstruction of lane line detection results

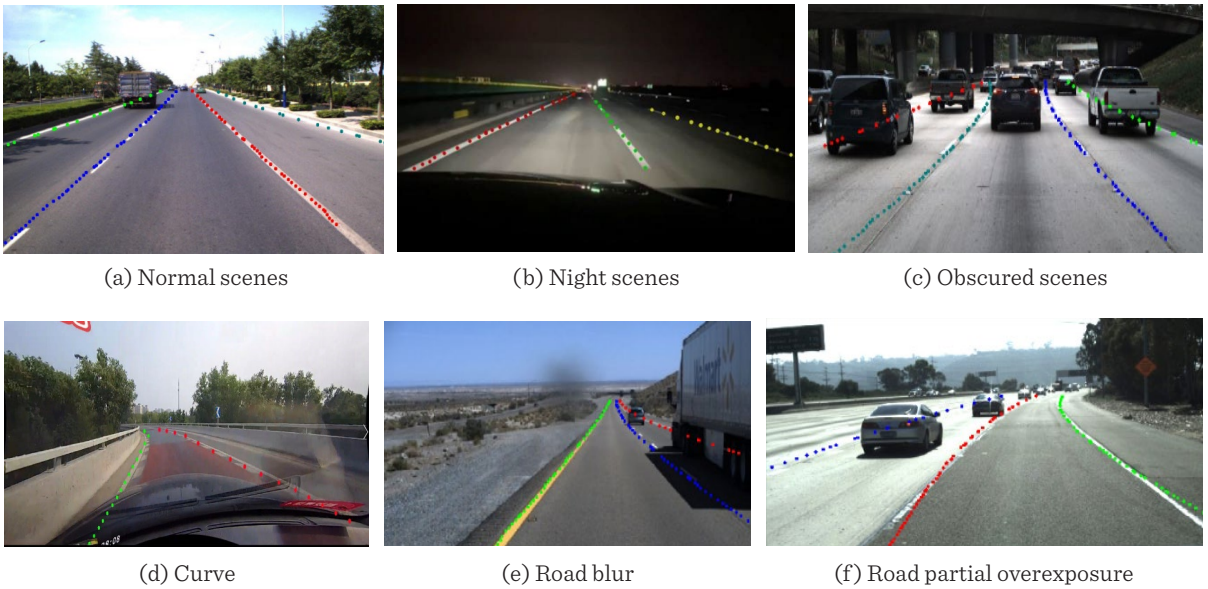


(a) (b) (c) (d)

line features. The mAP of the network model has increased by 2.6%. In the lane line fitting part, using the OPTICS clustering method adjusts based on the density of pixel points, resulting in better tolerance for fitting lane line pixel points, with a 2.2% increase in mAP. Combining these two improved network models, the mAP is 94.30%, a 4.6% improvement over the unimproved method. The proves that the improved method has a good effect on detecting occluded lane lines. The detection results are shown in Figure 11. In Figure 11(a), the first column shows images with a resolution of 1280x720. The subsequent columns in Figure 11(b) show the results of lane line instance segmentation. Figures 11(c)-(d) respectively show the LaneNet network and the results detected through the method described in this paper. It can be observed that when vehicles obstruct lane lines on the road, the features of the obstructed lane lines disappear. However, this method can cluster lane line pixel points and fit them through deep polynomials by combining global lane line features and instance segmentation results, enabling the detection of obstructed lane lines.

The colored dots in Figure 12 represent the fitting results of the current road lane lines, with different col-

Figure 12
Lane line detection results



(a) Normal scenes (b) Night scenes (c) Obscured scenes

(d) Curve (e) Road blur (f) Road partial overexposure

ors indicating different lanes on the road. Figure 12(a) shows the fitting results of lane lines in normal scenes with good lighting and no obstructions. Figure 12(b) shows the fitting results of lane lines in nighttime scenes. Figure 12(c) shows the fitting results when vehicle obstructs the lane lines. Figure 12(d) shows the fitting results for curved lanes. Figure 16(e) shows the fitting results when the road image is blurry. Figure 12(f) shows the fitting results in scenes with local overexposure on the road. The results indicate that the lane detection algorithm designed in this paper has good lane fitting capabilities in the above scenarios, thus enabling accurate positioning of lane lines in various complex scenes.

To further explain the detection results of the network, this article uses calculated evaluation indicators to illustrate the network's performance in detecting obscured lane lines, with the results shown in Table 3.

In Table 3, a comparison was made between the proposed method, LaneNet method, and SCNN method. Ultimately, in terms of accuracy, precision, and recall, the performance indicators of False Positive rate (FPR), False Negative rate (FNR), and Frames per second (FPS) were compared. Finally, As can be seen from Table 3, the method in this paper has improved the accuracy, precision, and recall rate of lane detection by 4.79, 6.34, and 2.10 percentage points, respectively. By introducing the CBAM module into the main network, the network pays more attention to the features of lane pixels, and flexibly uses polynomial fitting during lane fitting. Therefore, this paper has achieved a good improvement in the fault tolerance of occluded lane detection, with FPR and FNR decreasing by 6.89 and 2.70 percentage points, respectively.

4. Conclusion

This article proposes a method for detecting occluded lane lines based on deep polynomial regression under a global view. The method treats the lane line detection problem as a continuous and elongated instance segmentation problem. By introducing CBAM into the backbone network, the method improves the lane line segmentation network's ability to capture feature information. It utilizes lane line feature information to search for lane line pixels in the image under a global view, thus improving detection speed. Additionally, a vector block is added to the instance segmentation network to record the vector distance of lane line pixels, addressing the issue of mutual influence between pixels on different lane lines. Finally, the method completes the fitting of lane line pixels using deep polynomial regression. Finally, experiments were conducted on the Tusimple dataset. After comparing with the LaneNet and SCNN methods, it was found that this occluded lane line detection method based on deep polynomial regression with a global view improved both the accuracy and fault tolerance of lane detection. Compared to the LaneNet method, the false positive rate (FPR) and false negative rate (FNR) decreased by 6.89 and 2.70 percentage points, respectively. Compared to the SCNN method, the FPR and FNR decreased by 1.17 and 3.20 percentage points, respectively. The results indicate that this occluded lane line detection method based on deep polynomial regression with a global view can enhance the detection of occluded lane lines, which is significant for autonomous driving technology.

Table 3
Performance Index Comparison

Model	Accuracy	FPR	FNR	Precision	Recall	FPS
This article's method	94.30%	7.61%	4.16%	93.97%	95.84%	41
LaneNet	89.51%	14.5%	6.86%	87.63%	93.14%	35
The method in this article differs from LaneNet	+4.79%	-6.89%	-2.70%	+6.34%	+2.10%	+6
SCNN	91.63%	8.78%	7.36%	92.54%	94.36%	25
The difference between the methods in this article and SCNN	+2.67%	-1.17%	-3.20%	+1.43%	+1.48%	+16

However, although this method shows improvements over the LaneNet and SCNN methods, its accuracy metric can only reach 94.30%, which is still not sufficient for direct application in practical engineering. In the context of autonomous driving, it only serves as an auxiliary driving function. There is still a considerable gap in meeting the requirements for Level 3 autonomous driving. In future research, modifications to the deep polynomial regression and improvements to the VGG model can be made to enhance the accuracy of feature extraction, ultimately aiming to further improve the accuracy metric for occluded lane line detection.

References

1. Anbalagan, S., Srividya, P., Thilaksurya, B., Sai, G. S., Suganeshwari, G., Raja, G. Vision-Based Ingenious Lane Departure Warning System for Autonomous Vehicles. Sustainability, 2023, 15(4), 3535. https://doi.org/10.3390/su15043535
2. Chen, L., Xu, X., Pan, L., Cao, J., Li, X. Real-Time Lane Detection Model Based on Non-Bottleneck Skip Residual Connections and Attention Pyramids. PLoS One, 2021, 16(10). https://doi.org/10.1371/journal.pone.0252755
3. Ding, W., Xu, Y., Zhang, Z., Sun, K. Fast Lane Detection Based on Bird's Eye View and Improved Random Sample Consensus Algorithm. Multimedia Tools and Applications, 2017, 76(21), 22979-22998. https://doi.org/10.1007/s11042-016-4184-6
4. Ghanem, S., Kanungo, P., Panda, G., Parwekar, P. An Improved and Low Complexity Neural Network Model for Curved Lane Detection of Autonomous Driving System. Soft Computing, 2021, 1-12. https://doi.org/10.1007/s00500-021-05815-0
5. Li, Q., Yu, X., Chen, J., He, B., Wang, W., Rawat, D., Lyu, Z. PGA-Net: Polynomial Global Attention Network with Mean Curvature Loss for Lane Detection. IEEE Transactions on Intelligent Transportation Systems, 2024, 25(1), 417-429. https://doi.org/10.1109/TITS.2023.3309948
6. Liu, Y., Zhang, L., Xin, S., Zhang, Y. Video Action Classification Based on Deep Learning Network with Spatio-Temporal Attention Mechanism. China Science Paper, 2022, 17(3), 281-287.
7. Lu, P., Cui, C., Xu, S., Peng, H., Wang, F. SUPER: A Novel Lane Detection System. IEEE Transactions on Intelligent Vehicles, 2021, 6(3), 583-593. https://doi.org/10.1109/TIV.2021.3071593
8. Shen, H. Complex Lane Line Detection Under Autonomous Driving. 5th International Conference on Mechanical, Control and Computer Engineering, 2020, 627-631. https://doi.org/10.1109/ICMCCE51767.2020.00139
9. Wang, X., Deng, W., Liu, S. X., Huang, Z. T. Anomaly Detection Method of Electromagnetic Time Series Based on Attention Mechanism. Journal of Terahertz Science and Electronic Information, 2021, 19(4), 581-588.
10. Xu, H. H., Li, L., Fang, M., Hu, L. A Method of Real-Time and Fast Lane Line Detection. Eighth International Conference on Instrumentation and Measurement, Computer, Communication, and Control, 2018, 1665-1668. https://doi.org/10.1109/IMCCC.2018.00344
11. Zhan, J., Yang, Y., Jiang, W., Jiang, K., Shi, Z., Zhuo, C. Fast Multi-Lane Detection Based on CNN Differentiation for ADAS/AD. IEEE Transactions on Vehicular Technology, 2023, 72(12), 15290-15300. https://doi.org/10.1109/TVT.2023.3292401
12. Zhenqi, H. Research on UAV Flight Control and Communication Method Based on Fuzzy Adaptive. Wireless Networks, 2023. https://doi.org/10.1007/s11276-023-03408-3
13. Zhou, D. M., Qiu, S., Song, Y. A Driver-Assistance Algorithm Based on Multi-Feature Fusion. Infrared Physics Technology, 2021, 116. https://doi.org/10.1016/j.infrared.2021.103747