

Research Article

Automatic Lane Line Detection System Based on Artificial Intelligence

Gaoqing Ji  and Yunchang Zheng 

College of Electrical Engineering, Hebei University of Architecture, Zhangjiakou 075000, Hebei, China

Correspondence should be addressed to Yunchang Zheng; zyc2023@hebiace.edu.cn

Received 14 February 2022; Revised 27 April 2022; Accepted 23 May 2022; Published 9 June 2022

Academic Editor: Ahmed A. Zaki Diab

Copyright © 2022 Gaoqing Ji and Yunchang Zheng. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

As the need for an intelligent transport system is growing rapidly, lane line detection has gained a lot of attention recently. Aiming at the problem that the YOLOv3 algorithm has low accuracy and high probability of missed detection when detecting lane lines in complex environments, a lane line detection method for improving YOLOv3 network structure is proposed. The improvement is focused on detection speed and accuracy. Firstly, according to the characteristics of inconsistent vertical and horizontal distribution density of lane line pictures, the lane line pictures are divided into $s * 2S$ grids. Secondly, the detection scale is adjusted to four detection scales, which is more suitable for small target detection such as lane line. Thirdly, the YOLOv3's backbone is changed by adopting Darknet-49 architecture. Finally, parameters of anchor and loss function are optimized so that they focus on detecting lane line. The experimental results show that on the KITTI (Karlsruhe Institute of Technology and Toyoko Technological Institute) dataset, the mean average precision value is 92.03% and the processing speed is 48 fps. Compared with other algorithms, it is significantly improved in detection accuracy and real-time performance. It is promising to employ the proposed approach in lane line detection system.

1. Introduction

Lane line detection is an important part of intelligent transport systems, such as in traffic monitoring and autonomous cars [1]. Therefore, the need for the lane line detection system is increasing in the industry [2].

There have been various approaches to lane line detection. These approaches are generally divided into two categories: traditional methods and deep-learning-based methods. Traditional methods use statistical approach in extracting image features, such as color, gray, and edge [3]. On the other hand, deep learning-based approaches use convolutional neural networks as their feature extractor [4].

Although the accuracy of traditional approach is still acceptable, the traditional approach must go through a complex process and requires high human involvement in the development and deployment process [5]. Furthermore, traditional approaches also cannot propagate its training feedback to its feature extractor section [6]. Therefore, this

approach is considered not good for industry's production. The use of deep learning is one solution to answer the problem faced by the traditional approach.

One of the main issues in developing a deep learning model is the tradeoff between the level of accuracy with the speed of detection of the model. Models with high accuracy generally require quite complex feature extraction, which results in low detection speed. Therefore, building a deep learning model architecture needs to pay attention to both aspects.

One deep learning architecture that is highly considered to be used for object detection model is YOLO (You Only Look Once) [7]. The architecture uses a regression-based single step model. YOLO is considered to be one of the best architectures because it has a high detection speed without much sacrificing its detection accuracy [8].

For this paper, a lane line detection model will be built using the improved YOLOv3 architecture. The main contributions are as follows: firstly, according to the

characteristics of inconsistent vertical and horizontal distribution density of lane line pictures, the lane line pictures are divided into $s * 2S$ grids; secondly, the detection scale is adjusted to four detection scales, which is more suitable for small target detection such as lane line; thirdly, the backbone is changed by adopting Darknet-49 architecture; and finally, parameters of anchor and loss function are optimized so that they focus on detecting lane line.

2. Related Work

At present, the main methods of lane line detection include the method of extracting road features by machine vision, the method of establishing road model for detection, and the method of multi-sensor fusion detection [9].

The method of extracting road features by machine vision mainly uses machine vision technology to classify the gray value features and color features of lane lines. After learning, the lane lines on the road can be detected. Because the gray value, color and other features in the image are often affected by external conditions such as light intensity and shadow; the lane line detection using this method is easy to be disturbed by environmental changes [10].

The method of establishing the road model for detection is to establish a two-dimensional or three-dimensional image model of the road image and then compare it with the lane line in the photo which is to be detected. The application scope of this detection method is relatively small, and it is only suitable for roads with the characteristics of known templates. In addition, the algorithm has a large amount of computation and poor real-time performance.

The method of multisensor fusion detection is to detect the lane line by means of high-definition camera, UAV aerial photography, GPS, radar, and other fusion methods [11]. The cost of this method is high, and it is difficult to be applied in the large-scale practical lane line detection system [12, 13].

In this work, lane line detection process is done by using improved YOLOv3 model architecture. The model architecture is customized in such a way so that it can perform faster without much sacrificing its detection accuracy. The detected lane lines are then passed to the post-processing steps to validate each of them [14].

2.1. YOLOv3 Algorithm. YOLO is a real-time target detection algorithm proposed by Redmon et al. in 2016. It considers detection as a regression problem. Firstly, YOLO adjusts the size of the input image to a $448 * 448$ pixels, then convolutes the image, and finally detects the target by the full connection layer [15].

The algorithm divides the input image into $s * s$ grids. If the center point of the detected target falls into a grid, the grid is responsible for detecting the target. The numbers of parameters to be predicted for each grid are as shown in the following formula:

$$\text{Num} = s * s(5 * B + C). \quad (1)$$

In formula (1), B refers to the number of boundary boxes and C refers to the number of class probabilities [16]. Each

bounding box includes five predicted values: X, Y, W, H , and Conf which refers to the confidence of each bounding box. (X, Y) , respectively, represent the abscissa and ordinate values of the central point coordinates of the bounding box, and W and H are the width and height values of the bounding box.

The confidence of each bounding box refers to whether the goal is included in the bounding box and the accuracy of its prediction. The definition of confidence is shown in the following formula:

$$\text{Conf} = \text{Pr}(\text{Object}) \times \text{IoU}(\text{Pred}, \text{Truth}). \quad (2)$$

In formula (2), $\text{Pr}(\text{Object})$ refers to the probability that the bounding box contains the target. When the target is included, $\text{Pr}(\text{Object}) = 1$; otherwise, $\text{Pr}(\text{Object}) = 0$. In formula (2), $\text{IoU}(\text{Pred}, \text{Truth})$ refers to the matching degree between the prediction box and the real boundary box [17]. Its definition is shown in formula (3), where $\text{area}(\text{Boxt} \cap \text{Boxp})$ represents the area where the prediction box and the real bounding box intersect, and $\text{area}(\text{Boxt} \cup \text{Boxp})$ represents the area where the prediction box and the real bounding box are combined. When the prediction box completely coincides with the real bounding box, $\text{IoU}(\text{Pred}, \text{Truth}) = 1$. When the two do not coincide et al. l, $\text{IoU}(\text{Pred}, \text{Truth}) = 0$.

$$\text{IoU}(\text{Pred}, \text{Truth}) = \frac{\text{area}(\text{Boxt} \cap \text{Boxp})}{\text{area}(\text{Boxt} \cup \text{Boxp})}. \quad (3)$$

In formula (4), $\text{Pr}(\text{Class } i | \text{Object})$ refers to the probability of conditional category which is based on the condition that the grid contains detection targets and each grid only predicts the probability of a specific category of targets [18]. $\text{Pr}(\text{Class } i)$ is the probability that the detection target is a specific class target [19].

$$\text{Pr}(\text{Class } i | \text{Object}) = \frac{\text{Pr}(\text{Class } i)}{\text{Pr}(\text{Object})}. \quad (4)$$

YOLOv3 is a single-stage target detection algorithm proposed in 2018. It not only maintains the operation speed of the algorithm but also improves the prediction accuracy. It is a popular real-time target detection algorithm [20]. The network structure model of YOLOv3 is shown in Figure 1.

Compared with the previous versions of YOLO, the main differences of the YOLOv3 algorithm are as follows: using darknet-53 (106 layers in total, including 53 convolution layers) basic network model; using independent logical classifier instead of the softmax function; and using a method similar to FPN for multiscale feature prediction. The previous version of YOLOv2 can only use $1 * 1$ convolution kernel to detect on a single feature map. The most prominent feature of YOLOv3 is that it can detect targets in three sizes. When the feature size is $13 * 13$, it is used to detect larger targets; when the feature size is $26 * 26$, it is used to detect medium-sized targets; and when the feature size is $52 * 52$, it is used to detect smaller targets. Therefore, it solves the problem of adjusting the detection network according to the size of the target [21].

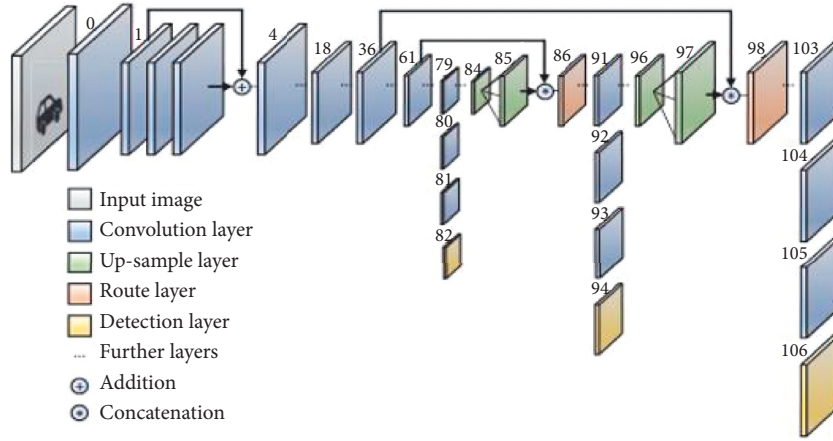


FIGURE 1: Network structure model of YOLOv3.

The detection steps of the three feature maps with different sizes in YOLOv3 are 32, 16, and 8. For the pictures with $416 * 416$ pixels, the first detection layer is located in the layer 82 of darknet-53 network. Because its detection step is 32, the feature map of this layer is a picture with a resolution of $13 * 13$. The second detection layer is located in the layer 94 of darknet-53 network. Because its detection step is 16, the feature map of this layer is a picture with a resolution of $26 * 26$. The third detection layer is located at layer 106 of darknet-53 network. Because its detection step is 8, the feature map of this layer is a picture with a resolution of $52 * 52$ [22].

2.2. Parameters of YOLOv3 Algorithm. Loss function is used to indicate the inconsistency between the predicted value and the real value in the model. The training model in the algorithm aims to reduce the value of loss function. The smaller its value is, the higher the robustness the model has.

Loss function which is frequently used includes mean square difference loss function and cross entropy loss function [23].

The definition of loss function in the YOLOv3 algorithm is shown in formula (5), which is composed of three parts: L_{coord} refers to prediction error of bounding box coordinate, L_{conf} refers to confidence error of bounding box, and L_{class} refers to prediction error of target classification.

$$L = L_{\text{coord}} + L_{\text{conf}} + L_{\text{class}}. \quad (5)$$

The definition of L_{coord} is shown in formula (6). In the formula, λ_{coord} refers to the weight coefficient of coordinate error, S^2 indicates that there are S^2 numbers of grids in the picture, and B refers to the number of prediction bounding boxes of each grid. I_{ij}^{obj} refers to the possibility that the j -th prediction box of the i -th grid contains targets. If targets are included, $=1$; otherwise $=0$. $(x_i, y_i, w_i, \text{ and } h_i)$, respectively, represents the abscissa, ordinate, width, and height values of the center point of the real bounding box in the i -th grid [24].

$$L_{\text{coord}} = \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B I_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + w_i - \hat{w}_i + (h_i - \hat{h}_i)^2 \right]. \quad (6)$$

The definition of L_{conf} is shown in formula (7). λ_{noobj} refers to the weight coefficient when the box does not contain targets target, and, respectively represent the real and predicted confidence. I_{ij}^{noobj} refers to the possibility that the j -

th prediction box of the i -th grid does not contain targets. Other parameters in the formula have the same meaning as formula (6).

$$L_{\text{Conf}} = \sum_{i=0}^{S^2} \sum_{j=0}^B I_{ij}^{\text{obj}} \left[(C_i - \hat{C}_i)^2 \right] + \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^B I_{ij}^{\text{noobj}} \left[(C_i - \hat{C}_i)^2 \right]. \quad (7)$$

The definition of L_{class} is shown in formula (8). There are C types of targets to be detected ($c = 1, 2, 3, \dots, C$), and, respectively represent the real and prediction probabilities

of targets of c -th class in the i -th grid. The other parameters in the formula have the same meaning as formula (6).

$$L_{\text{class}} = \sum_{i=0}^{S^2} I_{ij}^{\text{obj}} \sum_{c \in \text{class}} \left[(P_i(C) - \hat{P}_i(C)) \right]^2. \quad (8)$$

3. Proposed Method

Although the YOLOv3 algorithm is widely used in the field of real-time target detection, it is easy to be affected by external factors such as illumination, ground humidity, and so on. In order to improve the detection accuracy and speed, this paper improves the YOLOv3 algorithm, and the improved algorithm is more suitable for the lane line detection system.

3.1. Improvement of the Backbone. Firstly, in the conventional YOLOv3 algorithm, the backbone is Darknet-53, and the input image is divided into $s * s$ grids. When using the YOLOv3 algorithm for lane line detection, the shape feature of lane line image is that the transverse length is short and the longitudinal length is long. Therefore, in order to increase the grid detection density of image in the longitudinal direction, the algorithm is improved to divide the image into $s * 2S$ grids. This network structure is more suitable for the lane line detection system.

Secondly, under different road conditions, the size of lane line targets is different, which will result the poor effect of the detection algorithm. In order to solve this problem, the YOLOv3 algorithm is adjusted to four detection scales: $13 * 13$, $26 * 26$, $52 * 52$, and $104 * 104$. After the improvement, each detection scale not only obtains the detailed information of the bottom layer of the network but also can be predicted separately. The improved algorithm is more suitable for lane detection than the original lane detection system.

Finally, the YOLOv3 algorithm is implemented using Darknet-53 as the backbone for feature extraction. The down sampling operation is realized by convolution kernel, with the step size of 2 and the size of $3 * 3$. The receptive field of the network is large, but the spatial resolution of the network is insufficient. After extracting features through deep convolution, the information of small targets may be lost. As the detection of small targets such as lane line is more dependent on the shallow layer, Darknet-49 is used as the backbone of the algorithm.

There are 49 convolution layers and 5 residuals in Darknet-49. The activation function of ReLU may cause losses of low-dimensional feature information. So, linear activation function is used in the first convolution layer of Darknet-49 to reduce the loss of low dimensional information. Darknet-49 network performs 4 down sampling operations on the input image to obtain the characteristic images of 4 scales. The improved detection network expands the original three detection scales to 4 detection scales. Multiscale feature fusion can improve the detection performance. The improved detection network structure is shown in Figure 2.

3.2. Parameters of Anchor. In the process of target detection, the size of prediction boxes directly affects the accuracy and speed of the algorithm, which is a very important parameter in the algorithm. In the original YOLOv3 algorithm, the K-means clustering algorithm is used to obtain anchor parameters. In this method, different feature attributes in the distance center distance formula are treated with equal weight, without considering the differences of different feature attributes. Furthermore, the algorithm needs to estimate the K value which is the number of clusters by man-made. It has a certain subjectivity. In addition, if there are some noise points or isolated points in the sample, these points will have a great impact on the calculation results and cause serious deviation.

In the lane line detection system, in order to solve the problem of clustering center distance parameters, a method is proposed to replace the K-means clustering algorithm in the original algorithm with the intersection and union ratio of sample boxes and prediction boxes. The formula for calculating the parameters of anchor by this method is shown in formula (9). In the formula, parameter box is referred to the prediction box of the sample, parameter cen is referred to the clustering center, and IoU is referred to the area where the prediction box and the real box are combined.

$$D = \min \sum_{\text{box}=0}^n \sum_{\text{cen}=0}^k [1 - \text{IoU}]. \quad (9)$$

The detection results comparison between the improved method and the original YOLOv3 algorithm is shown in Table 1. Its average accuracy and speed are higher than those of the original YOLOv3 algorithm. It shows that the improved method of obtaining anchor parameters is more suitable for the detection of small targets such as lane lines.

3.3. Loss Function. The loss function is used to estimate the inconsistency between the predicted value and the real value in the model which is the basis for the evaluation of the false detection rate by the neural network and reflects the convergence performance of the network model. The loss function in the YOLOv3 algorithm consists of three parts, as shown in formula (5).

When calculating the loss function in the YOLOv3 algorithm, the boundary box coordinate prediction error, boundary box confidence error, and target classification prediction error are activated by logistic function. The function expression is shown in formula (10), and the derivative formula of the function is shown in formula (11).

$$f(x) = \frac{1}{1 + e^{-x}}, \quad (10)$$

$$f'(x) = \frac{e^{-x}}{2e^{-x} + e^{2x} + e^{2x}(1/e^{-x})^2}. \quad (11)$$

The characteristic of logistic function is that it is single and continuous, and the function value is limited. So, the

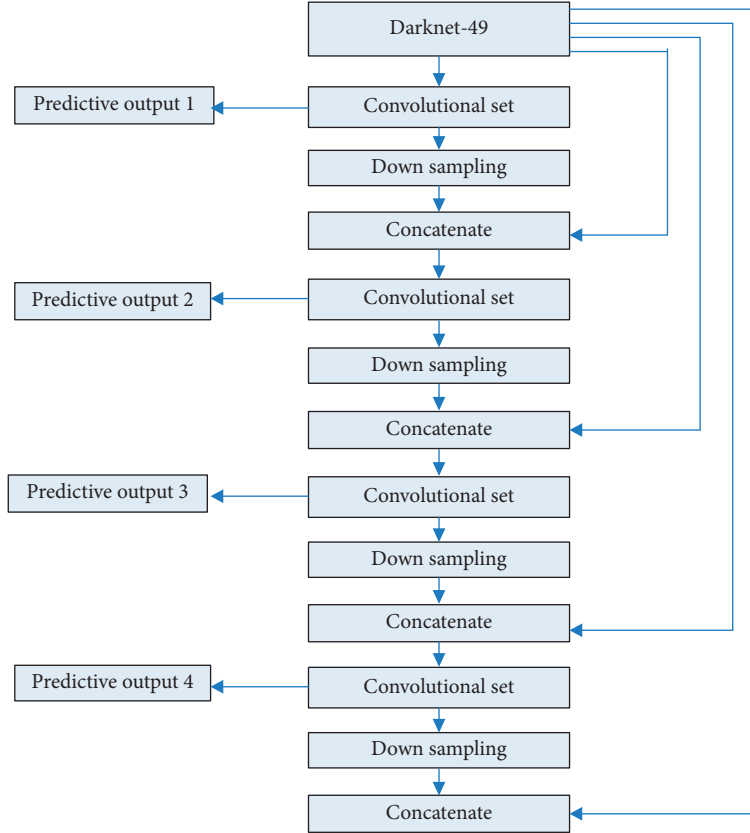


FIGURE 2: Improved network structure.

TABLE 1: Comparison of obtaining anchor parameters by different methods.

Methods	Anchor	Average accuracy (%)	Average speed (frame/s)
YOLOv3	(2, 8), (4, 10), (5, 14), (10, 24), (14, 33), (40, 89), (100, 283), (162, 320)	989.32	420.78
Improved algorithm	(3, 9), (6, 12), (8, 15), (12, 28), (20, 46), (49, 102), (113, 290), (185, 348)	991.56	443.21

data will not diverge during network transmission. The characteristic of the derivative of this function is that its value is less than 1. When the input value of the network is large, the derivative value will be very small, so the error of the calculated loss function will become large, and the convergence performance of the network model will become worse.

In order to solve the above problems, the focal loss function is proposed to replace the logistic function in the original algorithm. The expression of the focal loss function is shown in formula (12). In formula (12), p is the output value of the logistic function, $(1-p)^\gamma$ is the adjustment factor of the network system, α is the weight coefficient of the target

category ($0 \leq \alpha \leq 1$), γ is the focusing coefficient ($\gamma \geq 0$), and y is the predicted probability value of the tag.

$$FL(p, y) = -\alpha y (1-p)^\gamma \lg p - (1-\alpha)(1-y)p^\gamma \lg(1-p), \quad (12)$$

$$L_{\text{coord}} = \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B I_{ij}^{\text{obj}} [\text{FL}(x_i, \hat{x}_j) + \text{FL}(y_i, \hat{y}_j)] \\ + \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B I_{ij}^{\text{obj}} [\text{FL}(w_i, \hat{w}_j) + \text{FL}(h_i, \hat{h}_j)], \quad (13)$$

$$L_{\text{conf}} = \sum_{i=0}^{S^2} \sum_{j=0}^B I_{ij}^{\text{obj}} [\text{FL}(C_i, \hat{C}_j)] + \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^B I_{ij}^{\text{noobj}} [\text{FL}(C_i, \hat{C}_j)], \quad (14)$$

$$L_{\text{class}} = \sum_{i=0}^{S^2} I_{ij}^{\text{obj}} \sum_{c=\text{class}} [\text{FL}(P_i(C), \hat{P}_i(C))]. \quad (15)$$

The boundary boxes coordinate prediction error, boundary boxes confidence error, and target classification prediction error of the improved loss function are shown in formulas (13)–(15), respectively. After adopting the improved adjustment factor, the reception range of small targets such as lane line is effectively increased and the detection accuracy is improved.

3.4. Automatic Lane Line Detection System. Based on the improved YOLOv3 algorithm network structure, the lane line detection system is mainly composed of data preparation module, learning and training module, and lane line detection module. The implementation process is shown in Figure 3.

The main work of data preparation module includes the collection, marking, screening, and preprocessing of lane line samples. The sample collection method is to install a high-definition camera on the car and take lane line photos on the way. Data marking refers to marking lane lines in the picture, including white solid line, white dotted line, yellow solid line, yellow dotted line, and other lane lines. The main work in the screening and preprocessing stage is to select high-quality pictures from the marked pictures and preprocess them into the format required by YOLOv3. The nearest neighbor down sampling method is used as the main method of picture preprocessing.

The flow chart of the nearest neighbor down sampling algorithm is shown in Figure 4. Firstly, the coordinates of each pixel in the final picture should be traversed. Then, we use the nearest neighbor interpolation method to convert it into the corresponding coordinates in the final image. It is converted according to formula (16). In formula (16), $t(x', y')$ is the RGB value at the coordinate (x, y) in the final image, $s(x, y)$ is the RGB value at the coordinate (x, y) in the original image, and int refers to the rounding operation.

$$t(x', y') = s(\text{int}(x + 0.5), \text{int}(y + 0.5)). \quad (16)$$

The processed images are sent to the improved YOLOv3 algorithm network for lane line feature extraction. The four features of lane line are extracted at layers 67, 75, 89, and 97 of the network, and the features are sent to the YOLO layer for training. When the specified number of training times is reached, the iterative process is stopped. Then, the final detection model is generated, and the learning and training are completed. Finally, the lane line images are sent into the trained detection model, and the system can output the lane line detection results.

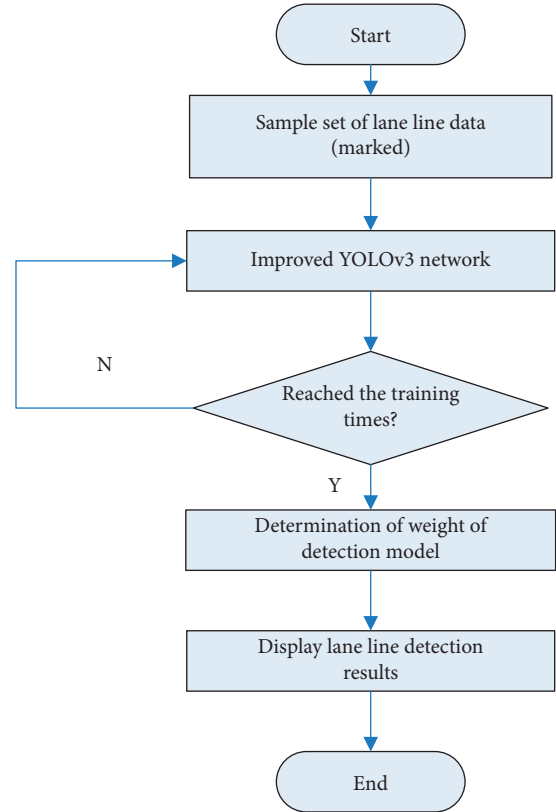


FIGURE 3: Process of the lane line detection system.

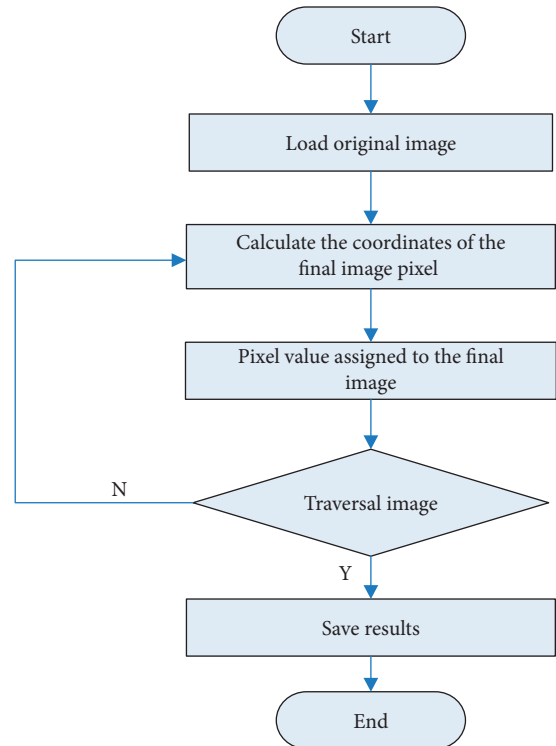


FIGURE 4: Flow chart of the nearest neighbor down sampling algorithm.

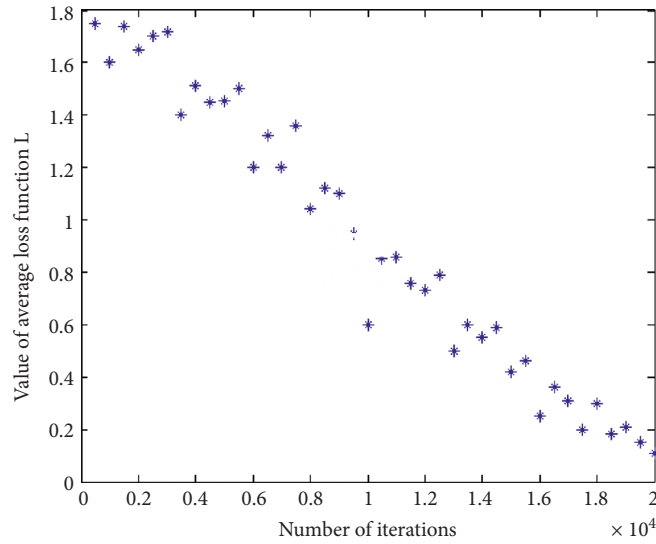


FIGURE 5: Variation trend of average loss function value.

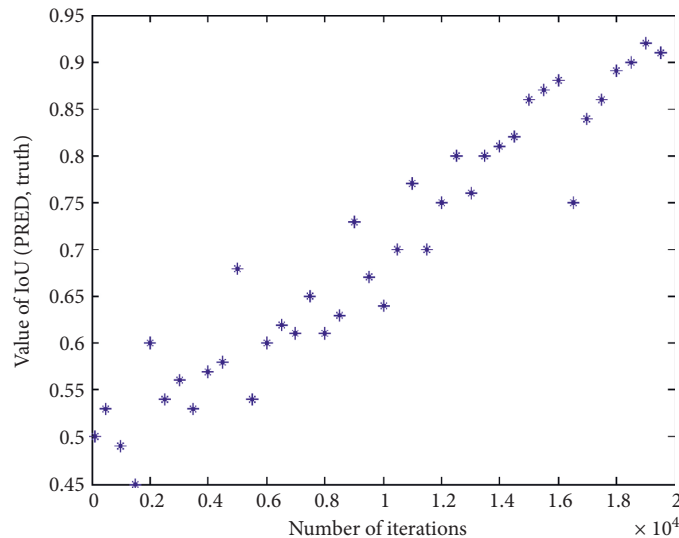


FIGURE 6: Change trend of matching degree value.

4. Results and Discussion

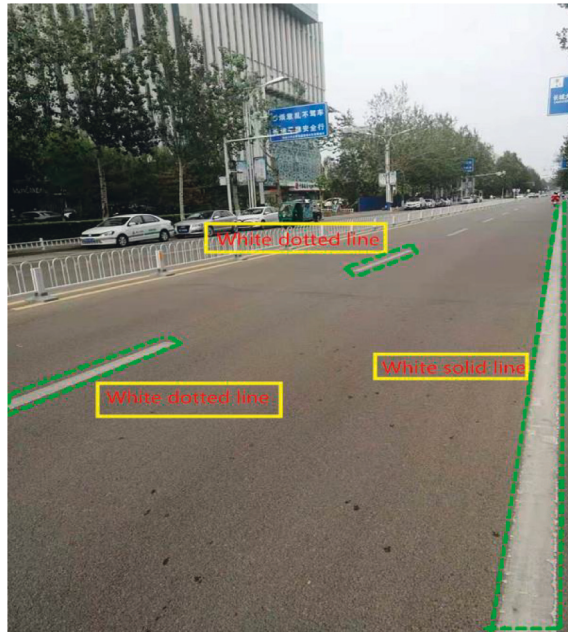
4.1. Experimental Environment. In the experiment, the hardware environment used is Intel® Core™-i7 4720HQ 2.6 GHz CPU and Nvidia TITAN X 12 GB VRAM GPU. The corresponding hardware environment is running under Ubuntu OS.

4.2. Experimental Dataset. Karlsruhe Institute of Technology and Toyoko Technological Institute (KITTI) is a commonly used traffic dataset and includes scenes, such as sunny, cloudy, highway, and urban roads. The scenes are increased under complicated working conditions, such as rain, tunnel, and night, to ensure coverage. In this experiment, KITTI is used as the dataset. A total of 500 photos of urban routes were selected, including 400 learning samples and 100 test samples.

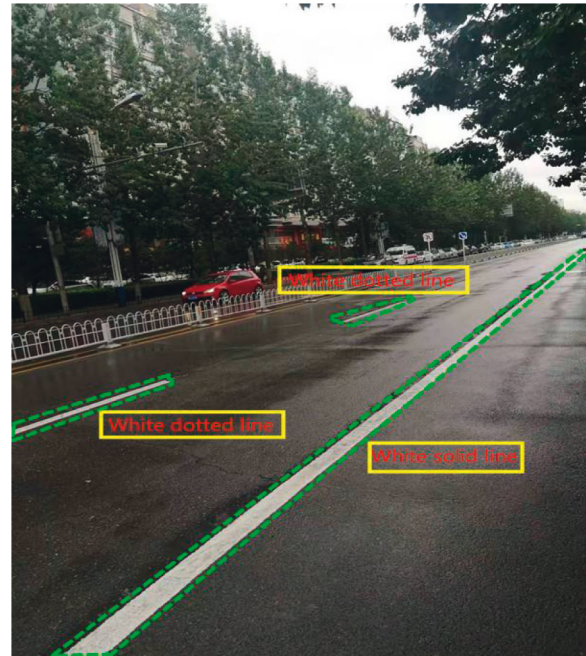
4.3. Experimental Results and Analysis. The marked lane line pictures are sent to the improved YOLOv3 algorithm network for training. In the training stage, the size of the pictures used is $416 * 416$ pixels. During the training process, several important index parameters in the algorithm are dynamically recorded.

The change process of the average loss function L value is shown in Figure 5. It can be seen from Figure 5 that at the beginning of training, the loss function value is about 1.8. When the number of training increases, the convergence of the loss function value decreases. When the number of iterations is about 20000, the loss value is about 0.1. The loss function L has achieved the expected effect.

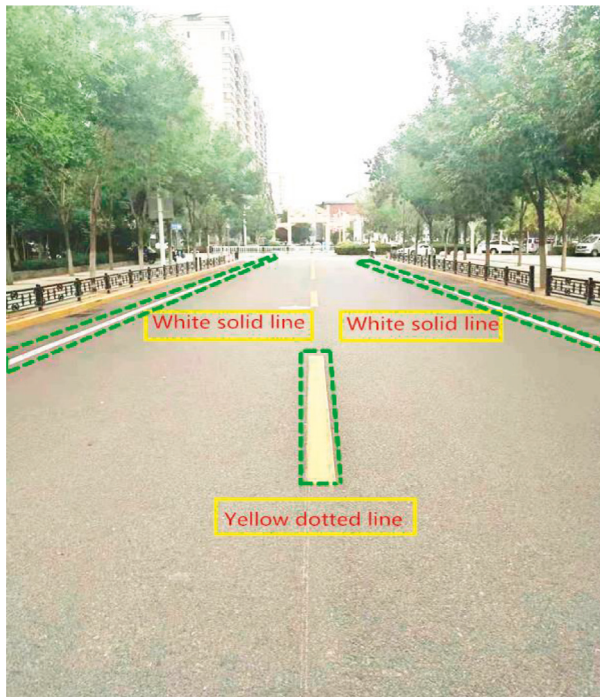
The change process of IoU (Pred, Truth) is shown in Figure 6. It can be seen from Figure 6 that at the beginning of training, value of IoU (Pred, Truth) is 0.5. When the number of training increases, it also increases gradually. When the



(a)



(b)



(c)



(d)

FIGURE 7: Results in different scenes: (a) normal illumination; (b) rainy days; (c) bright light; (d) dark light at night.

number of iterations reaches 20000, value of IoU (Pred, Truth) is about 0.93. The detection accuracy has achieved the expected effect.

According to the value of average loss function L , value of IoU (Pred, Truth), and other parameters, the training times of the model are set to 20000. After the sample learning and training, 100 lane line photos are tested, and the system can automatically identify the lane lines in the pictures. The

test results of the system in different scenes are shown in Figure 7.

4.4. Comparison of Different Algorithms. The automatic lane line detection system based on improved YOLOv3 has good performance in detection accuracy, detection time, missed detection rate, and so on. The performance parameters of

TABLE 2: Comparison of test performance of different algorithms.

Algorithm	FPS	mAP (%)	Times of iterations when $L = 0.1$	Average missed detection rate (%)
YOLO v3	8	84.87	20922	2.4
R-CNN	6	87.35	22178	4.2
Fast R-CNN	10	86.27	21896	8.1
Faster R-CNN	11	80.73	19989	9.3
YOLOP	11	81.04	20012	4.9
Improved YOLOv3	12	91.03	19985	1.1

different algorithms are shown in Table 2. It can be seen that the performance of the lane line detection system of the improved YOLOv3 model is better than that of the other algorithms in all aspects.

5. Conclusion

Aiming at the problem that the traditional lane detection algorithms cannot balance detection accuracy and detection speed, a detection system based on the improved YOLOv3 algorithm is proposed in this paper.

The main improvements include

- (1) According to the characteristics of inconsistent vertical and horizontal distribution density of lane line images, it is proposed to divide the images into $s * 2S$ grids to improve the vertical detection density
- (2) The detection scale is adjusted to four detection scales: $13 * 13$, $26 * 26$, $52 * 52$, and $104 * 104$, which is more suitable for the detection of small targets such as lane lines
- (3) The YOLOv3's backbone is changed by adopting Darknet-49 architecture, which simplifies the network and improves the system performance
- (4) The parameters such as cluster center distance and loss function are improved, which are more suitable for lane line detection system

The experimental results show that the improved algorithm has good detection performance when detecting flat roads, but when the roads have large slopes, the detection is easy to be affected. Therefore, solving the problem of lane line detection in large slope scenes will be the focus of the further study.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

This work was supported by Basic Scientific Research Project in Hebei Province (Nos. 2021QNJS13 and 2021QNJS06) and by Project of Zhangjiakou science and Technology Bureau (No. 1911002B).

References

- [1] H. G. Zhu, "An efficient lane line detection method based on computer vision," *Journal of Physics: Conference Series*, vol. 1802, Article ID 032006, 2020.
- [2] R. Xia, Y. Chen, and B. Ren, "Improved anti-occlusion object tracking algorithm using unscented rauch-tung-striebel smoother and kernel correlation filter," *Journal of King Saud University - Computer and Information Sciences*, vol. 18, no. 3, pp. 22–30, 2022.
- [3] C. Y. Fu, W. Liu, and A. Ranga, "Deconvolutional single shot detector," 2021, <https://arxiv.org/abs/1701.06659>.
- [4] T. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 936–944, Honolulu, HI, USA, July 2021.
- [5] S. Lee, J. Kim, and I. S. Kweon, "VPGNet: vanishing point guide network for lane and road marking detection and recognition," in *Proceedings of the International Conference on Computer Vision*, pp. 2380–7504, Venice, Italy, March 2021.
- [6] A. Gupta and A. Choudhary, "A framework for camera-based real-time lane and road surface marking detection and recognition," *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 4, pp. 476–485, 2018.
- [7] T. Lin, M. Maire, S. J. Belongie et al., *Microsoft Coco: Common Objects In Context (Eccv)*, Springer, Berlin, Germany, 2019.
- [8] R. B. Girshick, "Fast RCNN," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1440–1448, Washington, DC, USA, December 2020.
- [9] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection, computer vision and pattern recognition," vol. 1, pp. 886–893, in *Proceedings of the IEEE Computer Society Conference on 2021 CVPR*, vol. 1, pp. 886–893, IEEE, Seattle, WA, USA, June 2021.
- [10] X. Zhang, W. Yang, X. L. Tang, and J. Liu, "A fast learning method for accurate and robust lane detection using two-stage feature extraction with YOLO v3," *Sensors*, vol. 18, no. 12, p. 4308, 2018.
- [11] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pp. 7263–7271, IEEE, Honolulu, HI, USA, July 2017.
- [12] J. Redmon and A. Farhadi, "YOLOv3: an incremental improvement," 2018, <https://arxiv.org/abs/1804.02767>.
- [13] F. C. Soon, H. Y. Khaw, J. H. Chuah, and J. Kanesan, "Vehicle logo recognition using whitening transformation and deep learning," *Signal, Image and Video Processing*, vol. 13, no. 1, pp. 111–119, 2019.
- [14] M. Tan, R. Pang, and Q. V. Le, "Efficientdet: scalable and efficient object detection," in *Proceedings of the IEEE/CVF Conference On Computer Vision And Pattern*

- Recognition(CVPR)*, pp. 202010781–202010790, Seattle, WA, USA, June 2020.
- [15] C. F. Olson, “Constrained hough transforms for curve detection,” *Computer Vision and Image Understanding*, vol. 73, no. 3, pp. 329–345, 2021.
 - [16] Q. Zou, H. Jiang, Q. Dai, Y. Yue, L. Chen, and Q. Wang, “Robust lane detection from continuous driving scenes using deep neural networks,” *IEEE Transactions on Vehicular Technology*, vol. 69, no. 1, pp. 41–54, 2020.
 - [17] W. Y. Ming, Q. J. Cheng, and H. Yang, “Aerial image location of unmanned aerial vehicle based on Yolo v2,” *Laser and Optoelectronics Progress*, vol. 54, no. 11, pp. 101–110, 2019.
 - [18] W. Ming, J. Quan, and Y. Q. Hou, “Research on unmanned aerial image location based on Yolo v2,” *Progress in laser and optoelectronics*, vol. 54, no. 11, 2017.
 - [19] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: towards real-time object detection with region proposal networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.
 - [20] J. Redmon and A. Farhadi, “YOLOv3: an incremental improvement,” in *Proceedings of the Conference on Computer Vision and Pattern Recognition(PCCV)*, pp. 1–4, Salt Lake City, UT, USA, July 2018.
 - [21] A. Karpathy and J. Johnson, “Neural networks part 3: learning and evaluation,” *Cs231n Convolutional Neural Network For Visual Recognition*, vol. 25, no. 4, pp. 986–998, 2019, <http://cs231n.github.io/neuralnetworks-3>.
 - [22] J. Walsh, “Deep learning vs. traditional computer vision,” in *Proceedings of the Computer Vision Conference (CVC)*, Las Vegas, NA, USA, April 2019.
 - [23] C. Li, D. Song, R. Tong, and R. Min, “Illumination-aware faster RCNN for robust multispectral pedestrian detection,” *Pattern Recognition*, vol. 85, pp. 161–171, 2019.
 - [24] X. Wang, M. Seyf, and M. Chen, “Face detection using small-scale convolutional neural network (CNN) modules for embedded systems,” U.S. Patent 10,268-947, 2019.