Hindawi
Journal of Advanced Transportation
Volume 2022, Article ID 4489770, 11 pages
https://doi.org/10.1155/2022/4489770

Research Article

The Line Pressure Detection for Autonomous Vehicles Based on Deep Learning

Xuexi Zhang, Ying Li, Ruidian Zhan, Jiayang Chen, and Junxian Li

Guangdong University of Technology, Guangzhou, Guangdong 510006, China

Correspondence should be addressed to Ruidian Zhan; rd.zhan@gdut.edu.cn

Received 18 June 2022; Revised 31 July 2022; Accepted 4 August 2022; Published 10 September 2022

Academic Editor: Wen Liu

Nowadays, vehicle line pressure detection is an important function of an intelligent transportation system. At present, the line pressure detection algorithms mainly include algorithms based on traditional features and models and algorithms based on deep learning. However, these algorithms also have shortcomings such as low detection accuracy or relying on specific scenarios. In this regard, this paper proposes a fast and accurate vehicle line detection algorithm based on deep learning for vehicle images. The algorithm builds a GooleNet-based FCN semantic segmentation network and adds a BN layer, 1×1 convolution, and FPN structure to improve the segmentation effect of the GooleNet-FCN network and reduce network parameters. The MobileNet-SSD (no pretrained model) network structure is used for vehicle detection. According to the relationship between the receptive field and the anchor, and then combined with specific data, the prediction branch of the network and the Default Box on the branch are modified and the FPN structure is added for feature fusion to form the final improved MobileNet-SSD network. The experimental results show that the algorithm takes an average time of 67.8 ms per frame, the detection rate of line pressing for a vehicle is 96.6%, and the deep learning models are 25.5 M and 19.2 M, respectively. The experimental results verify the effectiveness and practicality of the detection algorithm proposed in this paper.

1. Introduction

With the development of science and technology, automobiles have become more and more common in our life, and the number of traffic accidents caused by automobiles has also increased rapidly. The main reasons for accidents are the bad habits and illegal operations of drivers, and so on. Among them, (the line pressing of a vehicle) LPV is the most common illegal driving behavior, which accounts for the highest proportion of traffic accident mortality. The existing traffic management system [1] uses surveillance cameras to monitor LPV, which can record the violations of drivers and optimize the distribution of traffic flow. In addition, it has a deterrent effect on drivers.

At present, the algorithms for lane and vehicle detection based on vision mainly include fixed cameras and vehicle-mounted cameras. Noteworthy, the fixed camera method, as reviewed in the literature [2], is based on the wavelet transform method to segment the yellow line area in the image without vehicles. Then the vision processor separates the same area of the subsequent captured real-time data and compares it with the template to determine whether there is the phenomenon of LPV. Literature [3] is based on the statistical method of gray frame difference, which mainly detects VPL by calculating the difference between the average value of the pixels for two adjacent frames and the interest region of image. If it is greater than the set threshold, it is judged as a broken lane line. Otherwise, it is judged as an LPV. Literature [4] analyzes that if the yellow line is pressed by the vehicle, the yellow line will have a gap. By analyzing the contour characteristics of the object that produces the gap, it is judged whether it meets the vehicle contour, and the centroid position of the vehicle is found from the contour, and then based on the distance between the vehicle centroid and the lane line to judge whether there is an LPV. On the other hand, in the mobile camera method, literature [5] collects a bird's-eye view of the road by installing the camera directly above the four wheels. If there is an intersection

between the lane line and the wheel tangent line, it is judged as an LPV. Literature [6] uses the camera to select a frame of picture data every 20 minutes and extract the color histograms around the yellow line of the picture as a template, and extract pictures every 5 frames based on the subsequent real-time data, the information of the same position of the template is matched with the information of the template, and the similarity between the two templates is compared to determine whether there is a LPV. Literature [7] expands the detected vehicle as a target in proportion, estimates the place where the wheel touches the ground, finds the tangent line between the vehicle and the ground, and judges whether there is an intersection with the lane line. If there is an intersection, the LPV is determined. In the literature [8], a data set of pressure line detection is first constructed, and then the vehicle and lane line detection is completed by combining the image semantic segmentation method, and then the front and rear wheel positions of the vehicle are obtained by the method of front and rear wheel estimation, and finally the vehicle pressure line judgment is realized by comparing the position of the wheel and the lane line. Literature [9] proposes to use the angle formed by the two lane lines and the vehicle in the process of driving, and it is simply defined that the maximum angle formed by the vehicle and the lane lines on both sides is greater than the set threshold, and it is determined as a line pressure. Literature [10] uses object detection and semantic segmentation techniques to obtain vehicle and lane information and uses a lightweight spatial convolutional network module to achieve high-precision segmentation of lane lines. The vehicle and lane line information is fused to detect the indentation behavior, and the solid dotted line is judged using pixel-based statistics in multiple areas of interest. In the literature [11], an automatic verification method for vehicle pressure line violations based on CNN and geometric projection is proposed. That is, in the three-dimensional coordinate system, the ground is the x-y plane, and the projection frame from the vehicle to the ground is fitted, and finally, according to the lane line, judging whether to press the line with the pose relationship of the projection frame.

In general, the current algorithms for line pressing detection based on deep learning have the following issues: (1) based on the high cost of a fixed camera, it cannot cover a large area; (2) most of the current algorithms are based on template matching to make judgments. For vehicles using this method, it is easy to cause false detection and missed detection; (3) the algorithms based on traditional features and models perform better in speed and accuracy in specific environments but have poor robustness on complex roads; (4) the algorithm based on deep learning has good adaptability to the environment, but the speed is slow and cannot be detected in real time; (5) algorithms based on CNN and geometric projection are difficult to automatically verify the line pressing behavior in the face of large vehicles.

Motivated by the aforementioned discussions, this article uses mobile devices to collect in-car video, designs a deep learning algorithm with a small number of model parameters and high accuracy to extract information, and finally determines whether the lane line and vehicle position are line-pressed. The main contributions of this paper are summarized as follows: (1) Considering that the deep learning detection algorithm needs a large amount of data, this paper builds the data set required for the experiment according to the needs and preannotating the constructed data, which save a lot of manual time, and then use the annotation tool to modify it. (2) GoogleNet-FCN semantic segmentation network is conducted to extract lane line and wheel-line line information, perform morphological filtering on the segmentation information and connected area threshold method to improve the segmentation effect, and the model evaluates the MIoU value on the test set to 66.2%. This paper improves the MobileNet-SSD network to modify the prediction branch and anchor value according to the specific data set and performs feature fusion. Therefore, the accuracy of the network performance on the data set is 95.7%, and the recall rate is 94.6%. (3) This paper establishes a pressure line algorithm model and uses the information extracted by the deep learning model to judge the line pressure of the vehicle. According to the position of the intersection of the lane line and the wheel-line line, a threshold is set to determine whether the line is pressed, and the best threshold is found according to the evaluation results of the different thresholds on the line pressure data. Finally, the performance of the algorithm is analyzed, which shows that the average time of the algorithm is 67.8 ms and the accuracy rate is 96.6%. It is proved that the target vehicle pressure line detection algorithm in this experiment has good accuracy and robustness, and it can also meet the real-time requirements.

The remaining part of this paper is organized as follows. In Section 2, the extraction scheme of vehicle image information based on deep learning is proposed. The performance analysis of vehicle line pressing and the simulation results showing the validation of our designed method are presented in Section 3. Finally, the conclusion is given in Section 4.

2. Extraction of Vehicle Image Information Based on Deep Learning

Deep learning is a new direction in many fields of machine learning. Due to its strong learning ability, strong adaptability, and excellent portability, it performs well in many fields, especially computer vision-related tasks [12]. In this paper, a lightweight model is used to extract useful information from experimental data. A GoogleNet-based full convolutional network semantic segmentation model is built for segmentation data to extract lane lines and other information. The vehicle detection data set uses an improved MobileNet-SSD target detection model.

2.1. Lane Line Detection Algorithm Based on GoogleNet-FCN

2.1.1. GoogleNet-FCN. GoogleNet-FCN is a (fully convolutional network) FCN based on the GoogleNet classification network, which is finally used in the detection of lane lines and wheel-line lines in the semantic segmentation data

set in this experiment [13]. The inception structure in GoogleNet has a small number of parameters and a wide width. It is a combination of speed and effect. This experiment continues to improve it. The improved model Inception structure is visualized in Netscope as shown in Figure 1.

From Figure 1, it can be observed that the improved Inception structure of this experiment replaces the 5×5 convolutional layer with two 3×3 convolutional layers, which can greatly reduce network parameters and enhance network nonlinearity.

From Figure 2, it shows the convergence effect of the BN layer by comparing the loss curve of the split lane line network with the BN layer and the loss curve without the BN layer. The loss in the segmentation network is softmax loss, the batch is set to 64, and the training is 15w times in total. It can be seen from the figure that without the BN layer, the loss fluctuation will be larger and the convergence time will be longer. Therefore, the network adds a BN (Batch Normalization) layer after each convolution, which can be used to solve the problem of nonuniform change scales between feature variables, so that the data becomes stable to reduce gradient dispersion and accelerate network convergence.

The entire model is a FCN model. In the decoder part of the model, the last convolutional layer of the Encoder is upsampled, as shown in the black box in Figure 3. The five black boxes are all upsampling processes. The entire network structure is FCN32s, which can improve the segmentation effect on small targets.

The blue part in Figure 3 belongs to the last convolutional layer of the decoder. In the black box, a 1×1 convolutional layer is introduced before the addition of the upsampling result channels to reduce the number of channels and greatly reduce the amount of calculation. The green part of Figure 3 shows the addition of feature channels. On the whole, the above-mentioned green boxes integrate network feature maps of different depths. This is also the Feature Pyramid Networks (FPN) feature fusion structure commonly used in deep learning now, which can better retains the target position and edge information and makes the segmentation effect more refined.

After continuous training and testing to modify the network, the brief structure parameters of GoogleNet-FCN are shown in Table 1.

Table 1 shows the final parameters of the GoogleNet-FCN network model, in which 1×1 convolution, BN layer, and FPN structure are not described in detail, and only the input and output of the key nodes of the network are summarized. From the perspective of deep learning, conv3 indicates that the size of the convolution kernel is 3×3, and Inception indicates input to the output of the structure. The final output size is $480 \times 224 \times 4$, consistent with the input data of the original image. 4 represents the 3 categories of this experiment and one background category, which is used for loss calculation or output during training.

The overall parameters of the network are small in number, and the BN layer is added to reduce overfitting and speed up the convergence, using the FPN structure to fuse features to improve the detection effect. At the same time, the network is FCN32s to improve the segmentation effect of small targets.

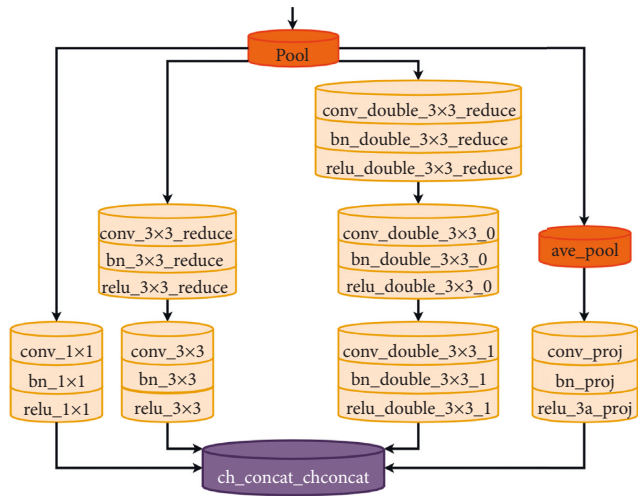


Figure 1: Improved inception structure.

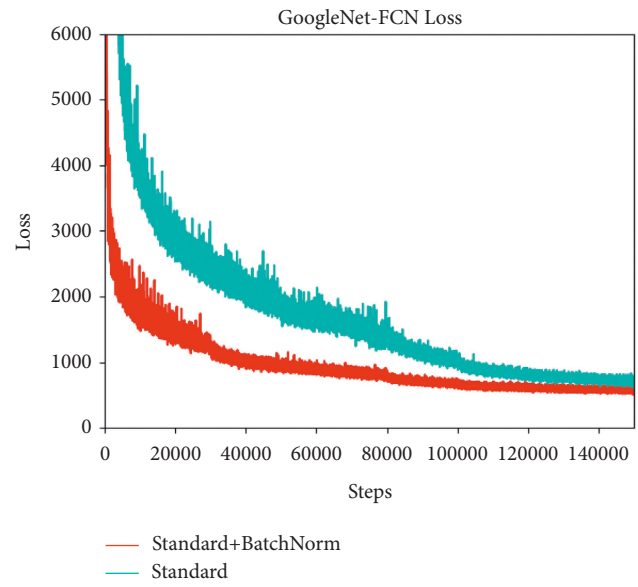


Figure 2: Comparison before and after adding the BN layer.

2.1.2. Lane Line Segmentation Effect and Postprocessing. The experiment was carried out under the Linux operating system, the corresponding caffe environment of GoogleNet-FCN was built, the segmented data set was trained, and the converged model was used to test the data. The original picture and the forecasted effect picture are as follows.

It can be seen from Figure 4 that the overall effect of segmentation of lane lines and wheel-line targets is better, and edge contour segmentation can be segmented. Analyzing the segmentation effects of all test sets, it is found that there are subtle edges with unevenness, background pixels appear in the target connected area, and a small number of target pixels appear in the background.

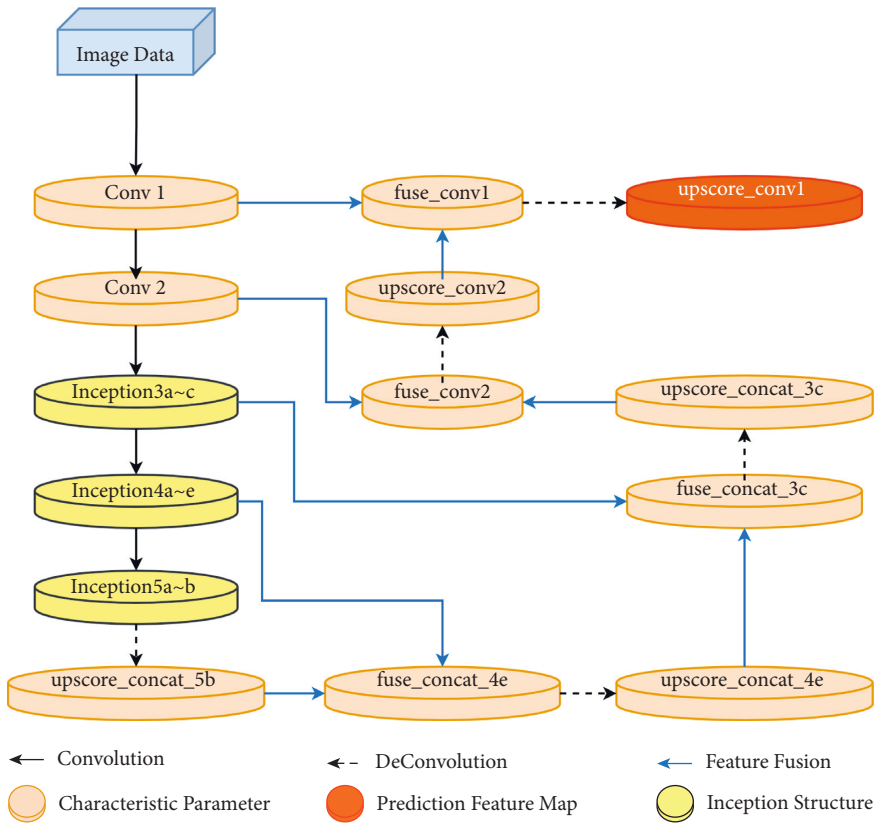
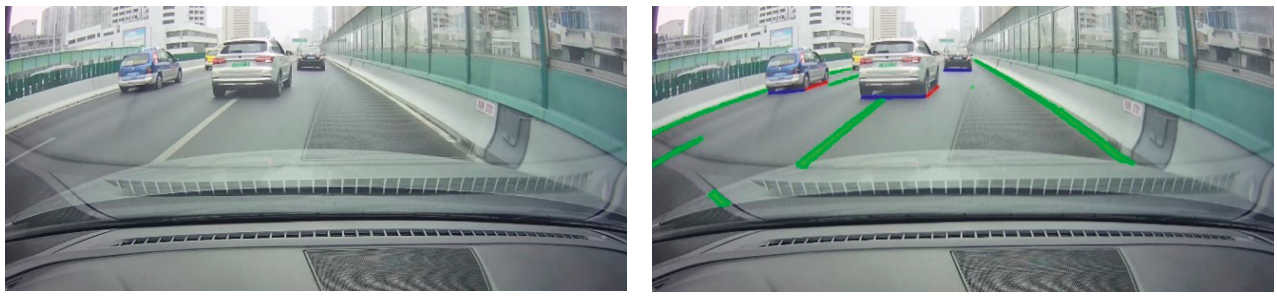


Figure 3: Improved GoogleNet-FCN network structure.

Table 1: Brief structure parameters of the improved GoogleNet-FCN.

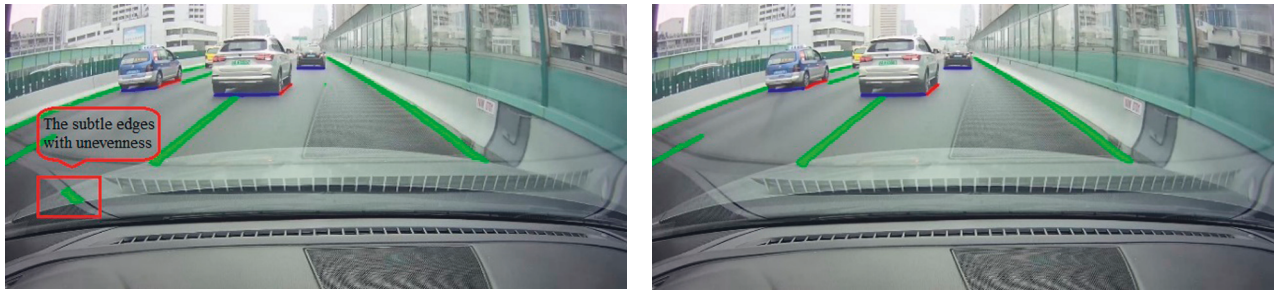
Input	Layer	Frequency	Output
$480 \times 224 \times 1$	Conv3	3	$240 \times 112 \times 96$
$240 \times 112 \times 96$	Pool	1	$120 \times 56 \times 96$
$120 \times 56 \times 96$	Conv3	1	$120 \times 5 \times 288$
$120 \times 56 \times 288$	Pool	1	$60 \times 28 \times 288$
$60 \times 28 \times 288$	Inception	3	$30 \times 14 \times 864$
$30 \times 14 \times 864$	Inception	7	$15 \times 7 \times 1024$
$15 \times 7 \times 256$	Upsampling	5	$480 \times 224 \times 4$



(a) (b)

Figure 4: (a) Original picture. (b) Segmentation effect diagram.

Because the lane line and the wheel-line shape edge are biased to a straight line, in order to solve the above problem, the processing scheme is carried out from two aspects: Firstly, the closed operation is used to smooth the segmentation of the target edge, and the second is the connected region threshold method to remove the pixel area whose area is less than the threshold. Figure 5 shows the model prediction effect and the postprocessing effect diagram.



(a) (b)

Figure 5: (a) Initial segmentation effect. (b) Postprocessing segmentation effect.

It can be seen that the target edge segmentation after postprocessing is more detailed and more in line with the target shape, and the small area target pixels that are incorrectly detected in the initial segmented background area are also improved.

2.1.3. Model Evaluation Results. Semantic segmentation is based on pixel-level segmentation. It commonly uses evaluation indicators which include Pixel Accuracy (PA), Mean Pixel Accuracy (MPA), and Mean Intersection over Union (MIoU). In this experiment, the evaluation code is written as needed and evaluated on the experimental split test set. The final results are shown in Table 2.

Suppose that there are $n+1$ classes in the segmented dataset. 0 denotes the background set and n is the number of target categories. p_{ii} means that the real class is i and the forecast class is also i; p_{ij} means that the real class is i, but the forecast class is j.

Pixel Accuracy (PA) represents the proportion of all pixels in the image correctly classified. The calculation formula is expressed as follows:

$$\mathrm{PA} = \frac{\sum_{i=0}^{n} p_{ii}}{\sum_{i=0}^{n}\sum_{j=0}^{n} p_{ij}}. \tag{1}$$

Mean Pixel Accuracy (MPA) represents the proportion of pixels correctly classified and the pixels predicted for each category in the image, and then the average is obtained as follows:

$$\mathrm{MPA} = \frac{1}{n+1}\sum_{i=0}^{n}\frac{p_{ii}}{\sum_{j=0}^{n} p_{ij}}. \tag{2}$$

From Table 2, we can obtain the pixel accuracy results of the network under the experimental data. The background has the largest proportion in the image and the segmentation accuracy is also the highest. Although the PA_wheel_line_v accuracy rate is the lowest at 0.668, it can also meet the application requirements.

Mean Intersection over Union (MIoU) represents the average of the intersection union ratio of each type of prediction result and the real label. The calculation formula is defined as follows:

Table 2: Pixel accuracy evaluation index.

Evaluation index	Result
PA	0.963
PA_bg	0.991
PA_lane	0.832
PA_wheel_line_v	0.668
PA_wheel_line_h	0.724
MPA	0.799

$$\mathrm{MIoU} = \frac{1}{n+1}\sum_{i=0}^{n}\frac{p_{ii}}{\sum_{j=0}^{n} p_{ij} + \sum_{j=0}^{n} p_{ji} - p_{ii}}. \tag{3}$$

The IoU evaluation indicators in Table 3 are more widely used in segmentation models. The experimental recognition rate of MIoU is 0.662. The background of IoU is the largest, which is easier to distinguish from other categories. The lowest IoU is 0.505, indicating that the ratio of nearly 0.7 prediction results for this category is the correct pixel, which satisfies experiment requirements in this paper.

2.2. Vehicle Detection Algorithm Based on Improved MobileNet-SSD Network

2.2.1. Improvement of MobileNet-SSD Network. MobileNet-SSD is a faster and less parameterized network designed based on the SSD network [14]. Its detection part is the same as that of SSD. It uses the convolution characteristics of different stages for multiscale prediction. MobileNet-SSD and the difference in SSD are that MobileNet-SSD replaces the basic network with the MobileNet [15] structure.

The entire convolution of the MobileNet-SSD network convolution structure adopts the depth separation convolution, and the BN layer and the ReLU layer are used to optimize the model after the convolution. Aspect rations are used to change the size of the anchor to make it a rectangle of the corresponding proportion, which is helpful for the model to find positive and negative samples for training, and at the same time, the matching degree of the target frame and the label is higher during the target regression. A well-designed Default Box can be beneficial to model detection

Table 3: IoU evaluation index.

Evaluation index	Result
redIoU_bg	0.923
redIoU_lane	0.679
redIoU_wheel_line_v	0.505
redIoU_wheel_line_h	0.544
MIoU	0.662

capabilities. How to set the anchor size for different data and how to choose the corresponding feature layer for anchor placement will be the key points for model improvement.

The important parameter Anchor in target detection transforms the detection problem into whether there is a recognized target in this fixed frame and how far the target frame deviates from the fixed frame. A well-designed Anchor is conducive to the model to find the difference between positive and negative sample characteristics, and at the same time, the target box and the label match better when the target returns. The setting of the Anchor size in the network is related to the Receptive Field (RF) and actual data of each convolutional layer.

The receptive field represents the size of the pixel points on the feature map mapped to the input image in the network. In the range mapped by the receptive field, there are differences in the importance of pixels at different positions, and the closer the pixel is to the center, the greater the impact on RF. According to the characteristics of convolution, the closer to the center, the more convolution times, the whole importance of division is similar to the Gaussian distribution, and the characteristics of the receptive field output are basically concentrated in the central area, which is the Effective (ERF) Receptive Field [16]. Each pixel on the feature map of the convolutional layer corresponds to a receptive field. The pixel extracts the features of the corresponding area of the theoretical receptive field, but the final response range of each layer to the output of the feature map is actually ERF.

The effective receptive field area is related to the actual size of the target. The size of the target in the image can be used to set the anchor size. According to the setting of the anchor size of the classic network, calculate the ratio of anchor to RF, explore the ratio, and set anchors in all layers of the network, setting up nine sets of comparative experiments according to the ratio of anchor to RF of 0.1 m to 0.9 m, designing the training network and tested on multiple sets of data, and finally founding that the ratio of anchor to RF is the best in the range of 0.2 m-0.3 m. The most suitable anchor value for this layer can be calculated according to the size of RF of the convolutional layer.

The specific Anchor value is related to the detection target in the data set. In this experiment, the vehicle is detected, and the anchor is the size of the vehicle target frame. This experiment will use a clustering algorithm to cluster all target sizes and get representative sets of data as anchor values. Furthermore, YOLOv3 was selected in the multiclustering method, and the clustering was used the IoU value of the box and the actual target box as the criterion so that the clustering target and the size of the target box would not have too much relationship and the clustering the data will be more balanced. This clustering method is used to cluster the vehicle detection data set, and 9 sets of coordinates are obtained by clustering. The cluster coordinates are converted into the Default Box and aspect ratios required by the MobileNet-SSD network.

According to the basic structure of MobileNet-SSD, the theoretical receptive field size of each layer of convolution is calculated, and according to the ratio of the anchor to the receptive field, the suitable range of the anchor for each layer can be found. Select the appropriate convolutional layer to predict the Default Box value calculated after the target box clustering. The improved MobileNet-SSD network predicts the convolution branch and its related parameters are shown in Table 4.

There are still 6 prediction branches in the network, but it is more in line with the detection of vehicles in the vehicle data set. In the end, the total number of predictable targets for each branch of the network is 6219 prediction boxes. The convolution depth of the model where the changed prediction branch is located is shallow, especially the prediction branch corresponding to the conv6 and conv8 convolutional layers. It may appear that the network is too shallow and has not yet extracted features that can be used for detection. Using the characteristics of the FPN structure, the deep feature layer and the shallow feature layer can be merged, and the shallow network contains more features such as locations, details, and deep semantic features, which can be used for network prediction. The finally improved MobileNet-SSD network structure is shown in Figure 6.

As can be seen from the figure, upsampling is performed at conv13, the number of relevant convolution channels is changed, and finally, feature fusion is performed with the features of the conv10 layer. After conv10 to conv8 and conv8 to conv6 adopt 1×1 convolution and deep convolution operations, the FPN structure is formed after fusion, and the fused features are trained and predicted.

2.2.2. Model Evaluation Results. In the experiment, the original MobileNet-SSD network and the improved network were trained separately, and the trained model was tested on the test set of vehicle detection data. The test comparison results are shown in Figure 7:

The evaluation indexes of target detection are generally precision (notated as P) and recall (notated as R). It is used to represent the error relationship between the number of detected targets and the real tag. Terms related to precision and recall are TPFPFNTN. The calculation formula of precision and recall are expressed as follows:

$$P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}},$$
$$R = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}. \tag{4}$$

It can be seen that the prediction effect of the improved network model is significantly better than that before the improvement. The improved network is more suitable for

Table 4: Improved MobileNet-SSD prediction branch and parameters.

Layer	Receptive field	Default box	Aspect red rations
Conv6	59	17	1.3
Conv8	123	21, 33	2.8, 2.3
Conv10	187	37, 46	1.1, 2.3
Conv13	315	56, 71	1.6, 2.1
Conv14	379	83, 112	1.3, 1.6
Conv15	517	112, 150	1.6, 2.0

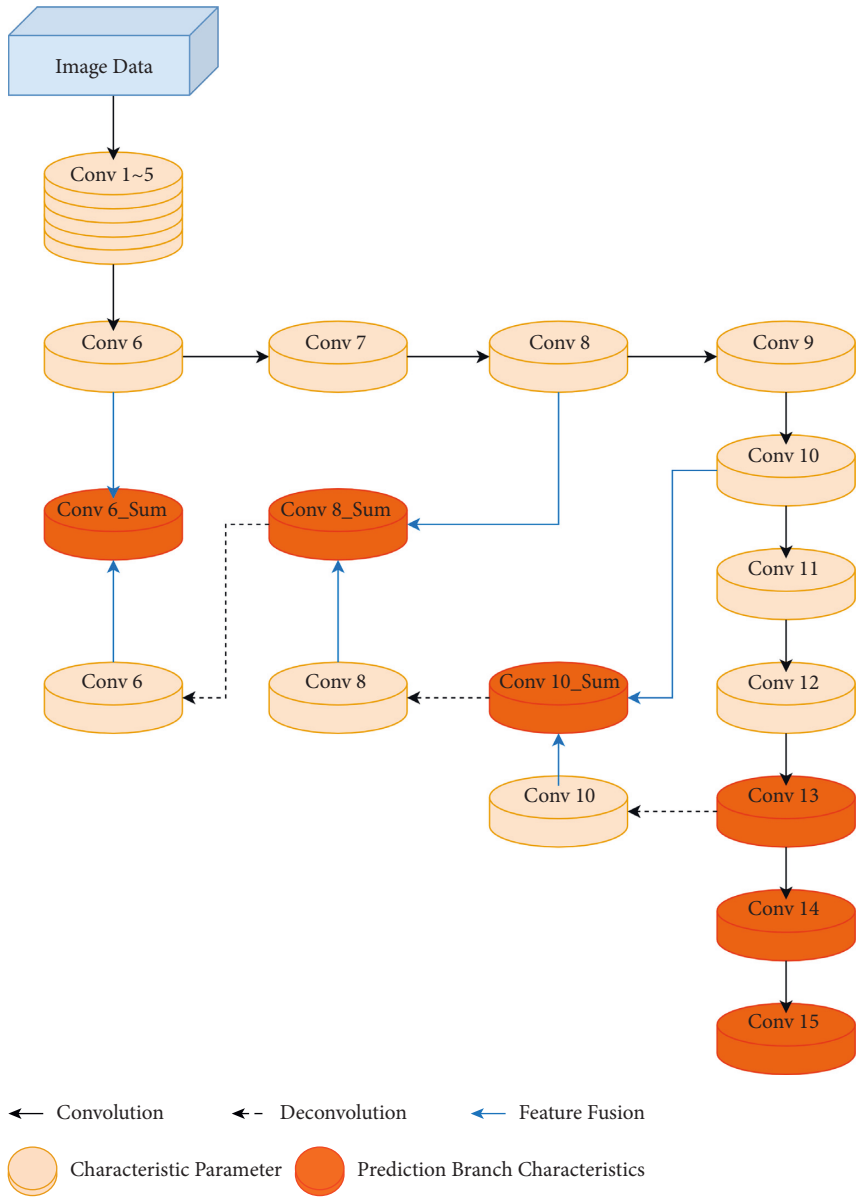


Figure 6: The improved MobileNet-SSD network prediction branch structure.

this data set, whether it is the fit between the prediction frame and the target or the prediction precision is more accurate.

The evaluation indicators of target detection are generally Precision and Recall. The experiment draws a comparison chart of the P-R curve of the network before and after the improvement, as shown in Figure 8.

The red curve in the figure is the improved PR curve, and the blue is the PR curve detected by the original network test. It can be clearly seen that the improved red curve is higher than the blue curve. The larger the area enclosed by the PR curve and the coordinate axis is, the better the network detection ability is. It proves that the improved network performs better on the experimental data set.

(a) (b)

Figure 7: The detection results of the vehicle data set before and after the network improvement. (a) Original network target detection result. (b) Improved network target detection result.

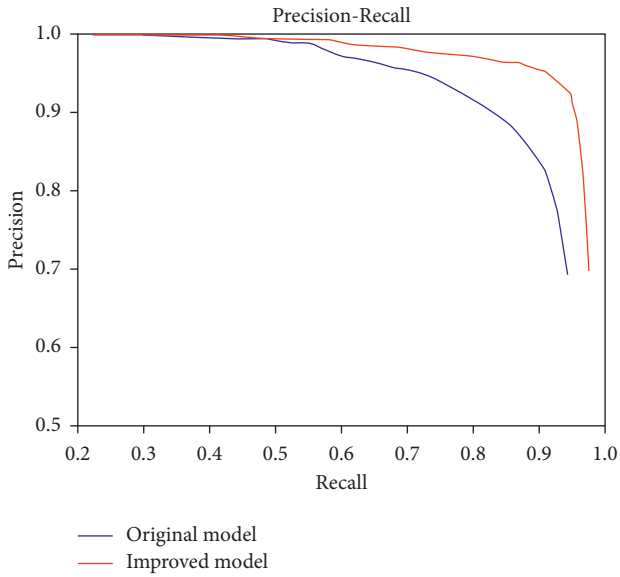


Figure 8: The P-R curve of the network on the vehicle detection data before and after the improvement.

The experiment set the IoU of the predicted target frame and the label to be greater than 0.5, and the predicted category is consistent with the predicted true positive sample. The conf value is the predicted probability of belonging to the category. When the value is set to 0.5, the vehicle target detection data set is evaluated, and the total number of vehicles in the test set is 18916. At this time, the TP is 17801, the FP is 800, and the FN is 996. The corresponding accuracy rate is 0.957, and the recall rate is 0.946, which meets the test availability standard.

3. Algorithm and Performance Analysis of Vehicle Line Pressing

3.1. Establishment of the Vehicle Line Model. The judgment of the vehicle pressure line is based on whether the entire car and lane line intersect in three-dimensional space, but the images collected by the vehicle camera belong to two-dimensional space. Therefore, how to use the two-dimensional space map to judge the vehicle pressure line will be the key to the experiment. The experiment uses a semantic segmentation network to segment the lane line and the wheel-line of the vehicle ahead captured by the vehicle camera (the blue line and the red line in Figure 4(b)), and finally the lane line and the wheel-line line. The information in the image can be used to judge the line pressure.

The detection data is output by the model built in the early stage of the experiment. That is, the lane line and the wheel-line line are divided by the segmentation model, the rectangular frame surrounding each car is detected by the vehicle detection model, and then the wheel-line line is judged whether it belongs to the rectangular frame of the detected car. If it belongs, use the position information of the wheel line and the lane line to judge whether the line is pressed. The specific process is shown in Figure 9.

As can be seen from the figure, the whole process is mainly divided into three steps. The first step is the straight line fitting of the lane line and the wheel-line line, the second step is the judgment of wheel-line ownership, and the third step is the judgment of the vehicle pressure line.

3.1.1. Lane Line Fitting. The lane lines are fitted using the postprocessed connected regions of the semantic segmentation model. At the same time, considering that when the detected vehicle is under line pressure, the distance between the recorder and the detected vehicle is relatively close, so the lane line can be approximated as a straight line, and the real situation of the wheel-line line is a straight line. In this paper, fitting is performed on the convex hull after postprocessing. This method greatly reduces the number of fitting detection points. Fitting adopts the improved probability Hough transformation [17, 18] to fit the edge of the straight line. The two straight lines of the same lane line are drawn through their center points to represent the middle line of the lane line.

3.1.2. Judgment of Wheel-Line Ownership. The wheel-line attribution judgment is to judge whether the wheel-line line belongs to the vehicle target frame detected in the current image, which can prevent other objects with similar

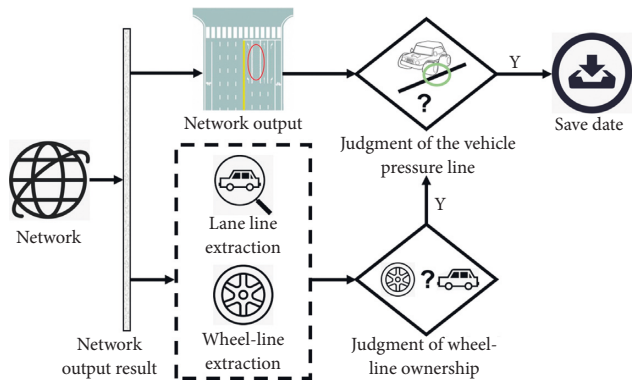


Figure 9: Flow chart of vehicle pressure line judgment.

characteristics from interfering with the experimental results. After the image is detected by the vehicle target, the target frame of the vehicle ahead will be predicted, so it is necessary to match the best wheel-line-h line (red line in Figure 4(b), defined as the line connecting the tangent points between the two front or rear wheels of the target vehicle and the ground) and wheel-line-v line (blue line in Figure 4(b), defined as the line connecting the tangent points between the two side wheels of the target vehicle and the ground). Extract the midpoint coordinates of the fitted wheel-line-h and determine whether the center coordinates are within the current vehicle target frame. If the center points of all wheel-line-h are not within the current vehicle target frame, the vehicle does not exist in wheel-line-h; if there is only one center point of wheel-line-h in the target frame of the current vehicle, then the wheel-line-h where the center point is located is the best matching item for the vehicle; if there are multiple center points in the target frame, the matching is performed according to the point closest to the center of the target frame. Use the same method to match the best wheel-line-v line to the current vehicle. Repeat the above operations until all detected vehicles have the best wheel line.

3.1.3. Judgment of Vehicle Pressure Line. After the target vehicle is matched to the corresponding wheel-line line, the positional relationship between the wheel-line line and the lane line is analyzed to determine whether the current vehicle has a line violation. The effect diagram after fitting is converted to a simple geometric model for analysis, as shown in Figure 10.

In Figure 10, $L_0 - L_2$ represents the lane line from left to right, the line segment AB represents the wheel-line-h line of the current vehicle, and the line segment AC represents the wheel-line-v line of the current vehicle. Point T is the intersection of line segment AB and L1; point G is the midpoint of line segment AB; line segment TG represents the distance from the intersection point T to midpoint G. Through the position of the intersection T in the line segment AB to determine whether the target vehicle is pressing the line, the relational expression is as follows:

$$\frac{\mathrm{TG}}{\mathrm{AB}} = \frac{T_x G_x}{A_x B_x} \leq \theta. \tag{5}$$

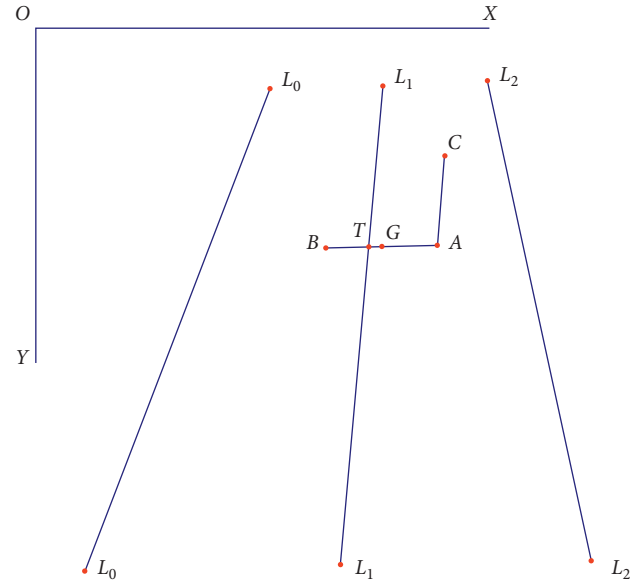


Figure 10: The geometric model of the fitting effect diagram.

The previous formula $T_x G_x$ represents the distance between the line segment TG and the X axis; the line segment $A_x B_x$ represents the distance between the line segment AB and the X axis; the angle θ is a set threshold, which represents the wheel-line of the target vehicle and the two ends of the lane line ratio. Only when the ratio of the distance TG from the intersection to the center point to the line segment AB is less than θ, the vehicle is judged to be on the line; otherwise, the line is not pressed. In the same way, the wheel-line-v can be judged by pressing the line.

3.2. Evaluation Results of the Line Pressing Algorithm. The judgment of LPV is a classification problem, and the evaluation indicators are set according to specific needs: accuracy, false detection rate, and missed detection rate. In this experiment, the accuracy rate is expressed as the ratio of the number of correctly classified images to the total number of predicted images; the false detection rate is the ratio of the number of all unlined images that are falsely detected as crimped images to the total number of predicted images; the rate of missed detection indicates the ratio of the number of unlined images that are predicted to be unlined images to the total number of predicted images. The total number of frames of the pressure line detection data set is 3415, and the number of pressure line frames is 996. The experiment is evaluated according to different values.

From Table 5, it can be found that when the lane line has an intersection with the wheel, a large number of missed detection will occur when the distance between the intersection points and the center of the wheel line is too close, and a large number of false detection will occur when the distance is too far. The accuracy of the algorithm is better when the distance is moderate. When the set threshold is equal to 0.25 m, the performance of the whole algorithm is the best. At this time, the accuracy rate reaches 96.6%, the missed detection rate is 1.67%, and the false detection rate is

Table 5: Evaluation results of the vehicle pressure line.

θ	Number of missed detections	Missed detection rate (%)	Number of false detections	False detection rate (%)	Predicting the pressure line	Number of correct detections	Precision (%)
0.1	452	13.23	4	0.1	548	2959	86.64
0.15	278	8.14	17	0.5	735	3120	91.36
0.2	154	4.51	36	1.05	878	3225	94.43
0.25	57	1.67	59	1.72	998	3299	96.60
0.3	14	0.4	105	3.07	1087	3296	96.51
0.35	5	0.1	219	6.41	1210	3098	90.72
0.4	2	0.06	648	18.97	1642	2765	80.96

Table 6: Algorithm time-consuming performance analysis.

Segmentation network preprocessing	Network segmentation	Split network postprocessing	Total time to split the network
5.4 ms	21.5 ms	6.2 ms	33.1 ms
Vehicle inspection	Press line judgment		Algorithm time-consuming
16.6 ms	18.1 ms		67.8 ms

Table 7: Analysis of network model parameters.

GoogleNet-FCN	Improved MobileNet-SSD (M)
25.5 M	19.2

1.72%, which shows that the vehicle line pressing algorithm in this paper is accurate and reliable.

3.3. System Performance Analysis. The algorithm is tested on the environment where the CPU is Core i7 and the graphics card is RTX2070. The performance of the algorithm is related to the experimental environment, but the algorithm itself is more important. This experiment is mainly divided into three modules: segmentation network module, target detection module, and pressure line judgment module.

The data in Table 6 are the average time for testing all images in the respective test set, the unit is ms, the total pressure line detection algorithm is 67.8 ms, and it can be detected as 14.7 frames per second in the above experimental environment. For a vehicle recorder, 30 frames per second, one or two frames can be selected for detection to meet real-time requirements.

This experiment uses two deep learning models to extract features, and the model parameters are given in Table 7.

Table 7 is the size of the model parameters obtained after training of the network built under the caffe framework. The two network parameters are small in number and can be transplanted to the mobile terminal for detection.

4. Conclusion

This article uses the image information obtained by the vehicle-mounted camera. The lightweight deep learning network is used to extract useful information from the vehicle image, and the vehicle pressure line detection algorithm model is constructed to realize a fast and accurate target vehicle pressure line detection method. Experiments show that this method has good accuracy and robustness for the target vehicle pressure line detection, and it can also meet real-time requirements and the algorithm can be transplanted to the mobile terminal to run. The whole system has certain practical application value.

Data Availability

The data supporting the research results of this paper are divided into three parts: the original vehicle video image, the segmentation effect image, and the vehicle detection effect image. The image data used to support the results of this study have been deposited at https://github.com/chenjiayang-fm/data-statement.git.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Guangdong University of Technology Higher Education Research Fund under GXLX20210213 and the National Natural Science Foundation of China under Grant 61803104.

References

[1] H. Lu, "Progress of intelligent transportation system key technologies," *Science and Technology Review*, vol. 37, no. 06, pp. 27–35, 2019.

[2] W. Zhao, Y. Zhan, and Y. Yan, "Design and implementation of vehicle pressing yellow line detection based on wavelet," *Computer Engineering and Design*, vol. 31, no. 10, pp. 2412–2415, 2010.

[3] J. Xiong and L. Hong, "Improved gray frame difference statistics of vehicle violation of traffic line," *Industrial Control Computer*, vol. 26, no. 05, pp. 112-113+116, 2013.

[4] H. Cheng, *Autonomous Intelligent Vehicles: Theory, Algorithms, and Implementation*, Springer Science & Business Media, Berlin/Heidelberg, Germany, 2011.

[5] F. Wang, C. Hu, and R. Zhou, *Lane pressure line detection method and system*, China, 2018.

[6] S. Memon, S. Bhatti, L. A. Thebo, M. M. B. Talpur, and M. A. Memon, "A video based vehicle detection, counting and

classification system," *International Journal of Image, Graphics and Signal Processing*, vol. 10, no. 9, pp. 34–41, 2018.

[7] H. Maeda, Y. Sekimoto, T. Seto, T. Kashiyama, and H. Omata, "Road damage detection and classification using deep neural networks with smartphone images," *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 12, pp. 1127–1141, 2018.

[8] K. Qiu and Z. Wang, "Lane-Crossing detection method of vehicles with in-vehicle image," *Computer Systems & Applications*, vol. 28, no. 11, pp. 188–194, 2019.

[9] K. Qiu and Z. Wang, "Lane-Crossing Detection and lane-departure waring on vehicle video," *Information Technology and Network Security*, vol. 38, no. 06, pp. 41–45, 2019.

[10] R. Yu, X. Jiang, K. Zhu, G. Jiang, W. Zhou, and Y. Wu, "Design of mobile terminal violation lane-crossing detection system based on information fusion," *Transducer and Microsystem Technologies*, vol. 40, no. 12, pp. 115–118, 2021.

[11] F. Gao, M. Zhou, L. Weng, and S. Lu, "An automatic verification method for vehicle line-pressing violation based on CNN and geometric projection," *Journal of Ambient Intelligence and Humanized Computing*, 2021.

[12] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, Las Vegas, NV, USA, October 2016.

[13] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3431–3440, Boston, MA, USA, June 2015.

[14] W. Liu, D. Anguelov, D. Erhan et al., "Ssd: single shot multibox detector," *European Conference on Computer Vision*, pp. 21–37, Springer, Cham, 2016.

[15] A. G. Howard, M. Zhu, B. Chen et al., "Mobilenets: efficient convolutional neural networks for mobile vision applications," 2017, https://arxiv.org/abs/1704.04861.

[16] W. Luo, Y. Li, R. Urtasun, and R. Zemel, "Understanding the effective receptive field in deep convolutional neural networks," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pp. 4905–4913, Barcelona Spain, December 2016.

[17] R. O. Duda and P. E. Hart, "Use of the Hough transformation to detect lines and curves in pictures," *Communications of the ACM*, vol. 15, no. 1, pp. 11–15, 1972.

[18] C. Galamhos, J. Matas, and J. Kittler, "Progressive probabilistic Hough transform for line detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision & Pattern Recognition*, IEEE, Fort Collins, CO, USA, June 1999.