Hindawi
Journal of Sensors
Volume 2020, Article ID 4372847, 14 pages
https://doi.org/10.1155/2020/4372847

Research Article

Vehicle Detection and Ranging Using Two Different Focal Length Cameras

Jun Liu and Rui Zhang

School of Automotive and Traffic Engineering, Jiangsu University, 212013, China

Correspondence should be addressed to Jun Liu; liujun@ujs.edu.cn

Received 14 October 2019; Revised 4 February 2020; Accepted 21 February 2020; Published 20 March 2020

Academic Editor: Giovanni Diraco

Vehicle detection is a crucial task for autonomous driving and demands high accuracy and real-time speed. Considering that the current deep learning object detection model size is too large to be deployed on the vehicle, this paper introduces the lightweight network to modify the feature extraction layer of YOLOv3 and improve the remaining convolution structure, and the improved Lightweight YOLO network reduces the number of network parameters to a quarter. Then, the license plate is detected to calculate the actual vehicle width and the distance between the vehicles is estimated by the width. This paper proposes a detection and ranging fusion method based on two different focal length cameras to solve the problem of difficult detection and low accuracy caused by a small license plate when the distance is far away. The experimental results show that the average precision and recall of the Lightweight YOLO trained on the self-built dataset is 4.43% and 3.54% lower than YOLOv3, respectively, but the computing speed of the network decreases 49 ms per frame. The road experiments in different scenes also show that the long and short focal length camera fusion ranging method dramatically improves the accuracy and stability of ranging. The mean error of ranging results is less than 4%, and the range of stable ranging can reach 100 m. The proposed method can realize real-time vehicle detection and ranging on the on-board embedded platform Jetson Xavier, which satisfies the requirements of automatic driving environment perception.

1. Introduction

Autonomous driving technology not only facilitates the driver but also improves the safety of the traffic environment. It senses the surrounding environment through various sensors, which is equivalent to the eyes of the vehicle, and then processes the sensory data to provide a dangerous warning for driving and even control the vehicle [1]. It can be seen that high precision and fast traffic target sensing technology is crucial for autonomous driving.

At present, the sensing sensors used in autonomous vehicles mainly include Lidar, millimeter-wave radar, ultrasonic radar, and camera [2–5]. Lidar can scan and measure by transmitting laser pulses to generate a precise map of road scene topography, which can be used for short-distance and long-distance obstacle detection. However, the high price of Lidar is not conducive to mass promotion. Millimeter-wave radar has a low price and strong ability to cope with environmental changes, but the perceived information is not comprehensive enough because it cannot identify the target. Ultrasonic radar measures short distances and is mostly used in scenes such as automatic parking. Compared with the above kinds of radar sensors, vision sensors are inexpensive and informative. The camera can accurately detect traffic targets such as vehicles, pedestrians, lane lines, and traffic signs through image algorithms and estimate the relative distance through the ranging model to complete the comprehensive perception of road information. Besides, some researches integrate multiple sensors such as radar and vision to improve perception efficiency [6, 7]. Considering the cost, a camera-based solution is a strategy of choice. For example, MobilEye uses a single camera to implement adaptive cruise control (ACC) and forward collision warning (FCW) [8, 9].

Traditional vision-based vehicle detection methods can be divided into two categories: the appearance-based method or the motion-based method. The appearance-

based methods use shadow feature [10], symmetry property [11], color [12], texture [13], and headlights/taillights [14, 15] to detect vehicle. Also, some methods introduce machine learning classifiers to train appearance features. Liu et al. used an AdaBoost classifier trained with the Haar-like features to detect vehicles [16]. Wei et al. proposed to train a support vector machine (SVM) by combining the features of Haar and histogram of oriented gradients (HOG) to extract vehicle positions [17]. The motion-based methods mainly include the optical flow method [18] and the dynamics background modeling method [19]. Fang and Dai proposed to combine the optical flow method and Kalman filter to realize vehicle detection and tracking [20].

In recent years, deep learning object detection algorithms based on the convolutional neural network have become more popular. One is a convolutional neural network based on region proposals, such as R-CNN [21], SPP-net [22], and faster R-CNN [23], but the calculation cost is relatively high. The other is a convolutional neural network based on the regression, such as YOLO [24], SSD [25], and YOLOv2 [26], which has continuously improved the calculation speed. Sang et al. proposed an improved YOLOv2 to improve detection accuracy and speed [27]. However, there are still tradeoffs between accuracy and speed when using convolutional neural networks for real-time vehicle detection.

Vision-based object ranging methods are mainly divided into two categories: stereo vision ranging and monocular vision ranging. Toulminet et al. proposed a stereo vision system to extract three-dimensional features of the preceding vehicle and calculate the distance [28]. However, stereo vision ranging is necessary to calibrate multiple cameras, and there are problems such as matching difficulties and massive calculation, which is not suitable for traffic target sensing in complex driving environments. Monocular vision ranging is relatively simple, which uses the object detection bounding box to estimate the distance. The distance estimation method based on monocular vision mainly uses the camera pinhole model or inverse perspective mapping (IMP). Adamshuk et al. proposed a distance estimation method based on IPM in the HSV color map, which used the linear relationship between the transformed image pixels and the actual distance [29]. Han et al. proposed calculating the distance based on vehicle width estimated by using the lane line and considered the situation without the lane line, but there is a big error in estimating the width of the lane line and target vehicle [30]. Mehdi et al. estimated the distance using the height and the pitch angle of the camera by assuming the road is flat, but this method does not consider lateral distance and the influence of camera attitude angles [31]. The above ranging methods have no length reference of a realistic target and rely solely on the imaging principle of the camera, which is challenging to achieve high robustness. Zhao et al. proposed to estimate the distance based on the license plate with a fixed width, but the license plate is difficult to accurately detect when the distance is long, and the plate is small, resulting in a limited scope of application [32].

Therefore, in order to solve the above problem that the deep learning target detection network is difficult to deploy on the embedded platform and the accuracy and robustness of vehicle ranging methods are unstable caused by lacks of the actual length reference, this paper proposes a vehicle and license plate detection model based on Lightweight YOLO and the long and short focal length cameras fusion ranging method. The main work of this paper is as follows:

(1) Propose the Lightweight YOLO network, combined with the multiscale prediction of YOLOv3 and the lightweight network Shufflenet, which reduces the number of parameters of the network model and improves the detection speed of the model under the premise of detection accuracy. Use the loss of generalized IoU to modify the loss function, and improve the regression accuracy of the detection bounding box. Use the Hungarian matching algorithm to match the position information of the vehicle in video frames, and use Kalman filter to achieve stable tracking of the same vehicle

(2) Use the license plate width to calculate the vehicle width and measure the relative distance between self-vehicle and the tracked vehicle. At the same time, the fusion ranging method based on matching vehicles captured by long and short focal length cameras is proposed to solve the problem that the vehicle and license plate are challenging to detect when far away

2. Vehicle Detection and Tracking

2.1. Lightweight YOLO Network. The backbone network Draknet53 of YOLOv3 [33] has too many convolution layers and network parameters, which takes up a large part of the time in the feature extraction process, resulting in slow network forward propagation.

Based on the network design structure of YOLOv3, this paper combines the lightweight network ShuffleNet [34] to build a Lightweight YOLO object detection model. ShuffleNet uses the depth-separable convolution (DWconv) [35] to reduce the computational complexity of convolution and introduces channel shuffle to increase the flow of information across channels. Compared to Darknet53, this network structure can map more channel features with lower computational complexity and memory loss.

The Shufflenet unit is composed of two convolution blocks. As shown in Figure 1, the convolution block 1 is a downsampling module. By copying input features, and performing deep convolution with a stride of 2 at the same time, the feature size is reduced by half and the number of channels is doubled. The convolution block 2 preserves the shallow feature semantic information by splitting and splicing the channel and ensures that the size of the output feature is unchanged. At the same time, the channel exchange is performed between every two convolutions blocks, and the feature channels are arranged in a cross, which solves the problems of information duplication and information loss caused by the channel split.

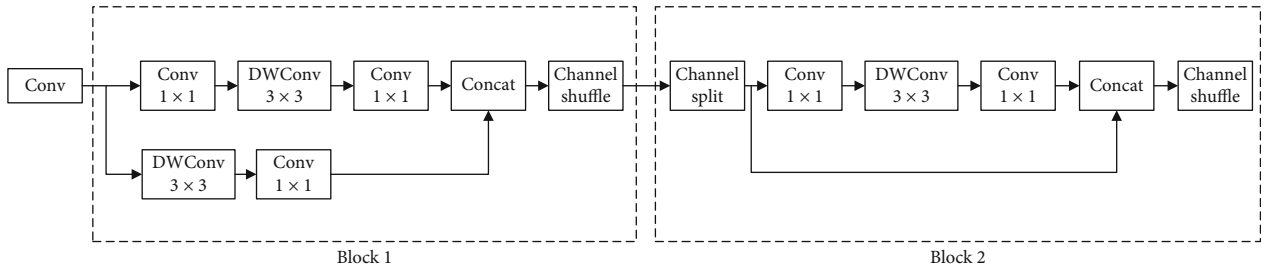

Figure 1: Shufflenet network convolution unit.

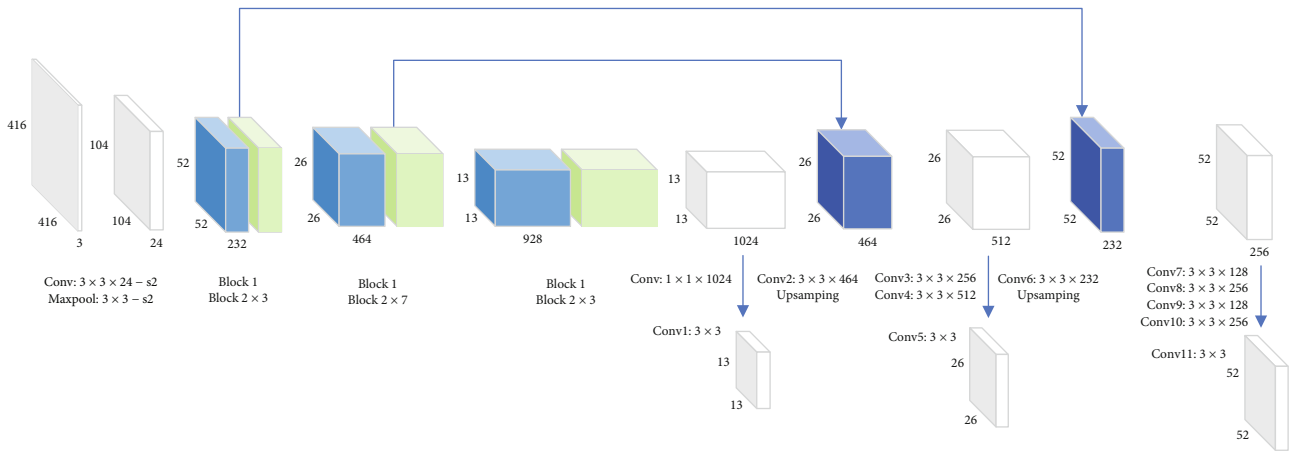

Figure 2: Lightweight YOLO network structure.

The auxiliary layer of the Lightweight YOLO network is modified based on the original network, retaining the multiscale prediction method and reducing the number of convolution layers and computational complexity of the entire network. Finally, the Lightweight YOLO network structure is shown in Figure 2. The backbone network uses the Shufflenet structure. The shallow and deep features are fused by upsampling the deep feature map. The network outputs prediction tensors at three different scales. The input image size is $416 \times 416 \times 3$, and the output sizes are three characteristic tensors of 13×13, 26×26, and 52×52, which detect objects of different sizes.

2.2. Loss Function. We found that the detection results of object detection models such as YOLO are very accurate and can successfully identify the object, but the boundary position of the detection box is relatively vague. The monocular vision ranging method mainly relies on the detection bounding box, and the detection box of the boundary blurring causes the ranging accuracy to be low or even invalid, so it is crucial to improve the accuracy of the detection bounding box.

During the training process, the convolutional neural network continuously updates the model parameters through the loss function and backpropagation, reducing the model loss and improving the detection accuracy. The loss function of YOLOv3 consists of three parts: the bounding box prediction L_{bbox}, the confidence prediction L_{conf}, and the category prediction L_{cls}. However, the bounding box predicts the loss using the mean square error, which only reflects the distance attribute between the detection box and the actual bounding box, while ignoring the Intersection over Union (IoU), as shown in Figure 3. When two rectangular boxes have the same L2 norm distance, their IoU may be different. It is necessary to introduce IoU prediction loss into the loss function.

The calculation formula for IoU is as follows:

$$\text{IoU} = \frac{|A \cap B|}{|A \cup B|}, \tag{1}$$

where A and B, respectively, indicate the detection box and the ground truth box.

However, when the two boxes are in different superimposed states, IoU may be the same. Using IoU as a loss function has a considerable drawback. Therefore, this paper uses generalized IoU [36] as the optimized IoU loss calculation method. The formula is as follows:

$$\text{GIoU} = \text{IoU} - \frac{|C \setminus (A \cup B)|}{|C|}, \tag{2}$$

where C is the smallest rectangular box containing both A and B. When two rectangles do not coincide, generalized

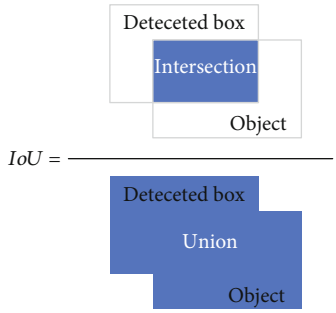

Figure 3: IoU calculation principle.

IoU can still describe the relative relationship. As shown in Figure 4, two rectangles still have the same IoU value in different overlapping cases, but the generalized IoU value of the right case is smaller than the left case. In other words, generalized IoU can highlight the misalignment between the two rectangles. It can be seen that generalized IoU solves the critical problem that IoU is not suitable as a loss function.

Finally, the loss function used by the Lightweight YOLO is as follows:

$$\begin{aligned} \text{Loss} &= L_{\text{conf}} + L_{\text{cls}} + L_{\text{bbox}}, \\ L_{\text{bbox}} &= 1 - \text{GIoU}. \end{aligned} \tag{3}$$

2.3. Vehicle Tracking. Vehicle tracking adopts the detection-based multiple object tracking method SORT proposed in [37]. The interframe displacements of the vehicle can be seen as a linear constant velocity model which is independent of other vehicles and camera motion, and the state of each vehicle can be defined as follows:

$$x = [u, v, s, r, \dot{u}, \dot{v}, \dot{s}], \tag{4}$$

where u and v represent the coordinates of the center point of the target vehicle and s and r represent the area and the aspect ratio of the detection box of the target vehicle, respectively.

This tracking method uses the Hungarian assignment algorithm [38] to correlate the detection box with the target vehicle based on the IoU between the predicted bounding box of the vehicle and the vehicle detection box in the current frame. When a detection box is associated with a target vehicle, the detected bounding box is used to update the target vehicle state where the velocity components are solved optimally via a Kalman filter [39]. If no detection box is associated with the target vehicle, the linear model is used to predict its state.

If the tracker matches the detection box for two consecutive frames, the algorithm judges it to be a valid tracking and outputs the detection bounding box and the corresponding tracker ID. If the tracker does not match any detection box for three consecutive frames, it is determined that the target disappears, and the predicted bounding box of the track is output during this period. The details of the tracking algorithm can be referred to in the literature [37].

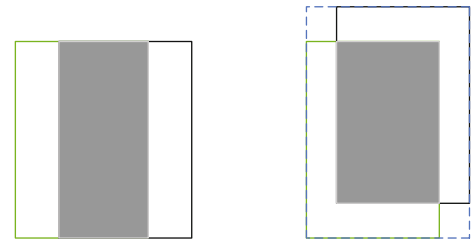

Figure 4: Two overlapping cases between two rectangles with the same IoU values.

3. Fusion Ranging

Most of the existing vehicle ranging methods rely on camera imaging principle to estimate the distance, mainly divided into two types: one based on the vehicle position [8, 29–31] and one based on the vehicle width [30, 32]. The location-based ranging model assumes that the road is flat and sensitive to noise. The distance measurement model based on the vehicle width is relatively robust because the number of the pixels of the vehicle width is less affected by the change of the pitch angle of the camera, but the vehicle width is not fixed and cannot be accurately measured. On this basis, this paper proposes a long and short focal length camera fusion ranging method, which firstly matches the target vehicle and license plate detected through the long focal length and short focal length cameras, then calculates the vehicle width through the license plate information, and finally calculates the distance between the two vehicles using a pinhole model. The specific steps are shown in Figure 5.

Step 1: Capture the image of the road ahead through the short focal length camera and detect vehicle and license plate and then track the vehicle.

Step 2.1: When the tracked vehicle is relatively close, the license plate position can be accurately detected, and the actual vehicle width of the tracked vehicle can be calculated based on the actual width of the license plate by Equation (5). Then go to Step 3.

Step 2.2: When the tracked vehicle is far away, the license plate cannot be accurately detected because the pixel width is small. Capture the current frame image by the long focal length camera and detect vehicle and license plate. Find a vehicle that matches the tracked vehicle by Algorithm 1, and calculate the actual vehicle width based on its license plate width by Equation (5).

Step 3: After obtaining the actual vehicle width of the vehicle, the actual distance of the tracked vehicle can be calculated by Equation (6).

The actual width of the vehicle is calculated as follows:

$$W = \frac{W_l}{w_l} w, \tag{5}$$

where W_l and W represent the actual width of the license plate and the actual width of the vehicle, respectively. The national standard stipulates that the width of the license plate

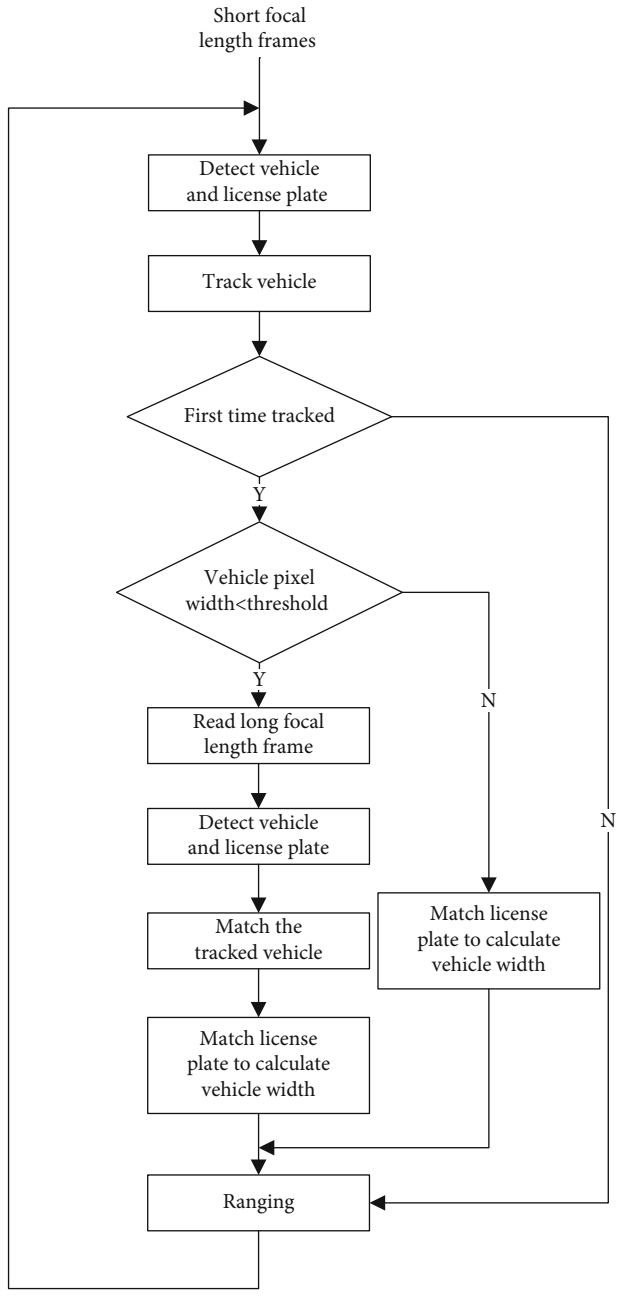

Figure 5: Long and short focal length fusion ranging flowchart.

of China is 440 mm. w_l and w indicate the pixel width of the license plate and the vehicle, respectively.

As the camera pin-hole model shown in Figure 6, the actual distance between the tracked vehicle and self-vehicle is calculated as follows:

$$D = \frac{f \cdot W}{w}, \tag{6}$$

where f represents the camera pixel focal length.

The two images taken at the same location by two cameras with different focal lengths have the same central point position, except that the image captured by the long focal length camera has a narrower field of view, and the relationship of the field of view between the long focal length and the short length is as follows:

$$\frac{W_1}{W_2} = \frac{f_2}{f_1}, \tag{7}$$

where W_1 and W_2 are the width of the field of view of the short focal length camera and the long focal length camera,

Input: Short focal length vehicle box $B^{w} = (x_1^{w}, y_1^{w}, x_2^{w}, y_2^{w})$,
Long focal length vehicle box $B^{l} = (x_1^{l}, y_1^{l}, x_2^{l}, y_2^{l})$.
Output: B^{l} matched with B^{w}.
Zoom B^{l} to $B\wedge^{l}$: $B\wedge^{l} = (\hat{x}_1^{l}, \hat{y}_1^{l}, \hat{x}_2^{l}, \hat{y}_2^{l})$
$\hat{x}_1^{l} = 1/2w(1 - f_1/f_2) + (f_1/f_2)x_1^{l}$, $\hat{x}_2^{l} = 1/2w(1 - f_1/f_2) + (f_1/f_2)x_2^{l}$,
$\hat{y}_1^{l} = 1/2h(1 - f_1/f_2) + (f_1/f_2)y_1^{l}$, $\hat{y}_2^{l} = 1/2h(1 - f_1/f_2) + (f_1/f_2)y_2^{l}$.
For each B^{w}:
Calculating area of B^{w}: $A^{w} = (x_2^{w} - x_1^{w}) \times (y_2^{w} - y_1^{w})$.
For each $B\wedge^{l}$:
Calculating area of $B\wedge^{l}$: $A^{l} = (\hat{x}_2^{l} - \hat{x}_1^{l}) \times (\hat{y}_2^{l} - \hat{y}_1^{l})$.
Calculating intersection v between B^{w} and $B\wedge^{l}$:

$$x_1^{v} = \max(x_1^{w}, \hat{x}_1^{l}),\ x_2^{v} = \min(x_2^{w}, \hat{x}_2^{l}),$$
$$y_1^{v} = \max(y_1^{w}, \hat{y}_1^{l}),\ y_2^{v} = \min(y_2^{w}, \hat{y}_2^{l})$$
$$v = \begin{cases} (x_2^{v} - x_1^{v}) \times (y_2^{v} - y_1^{v}), & \text{if } x_2^{v} > x_1^{v}, y_2^{v} > y_1^{v} \\ 0 & \text{otherwise.} \end{cases}$$

IoU = v/o, where $o = A^{l} + A^{w} - v$.
End for
Find the maximum IoU as the matched vehicle.
End for

Algorithm 1: The long-and-short focal length vehicle bounding box matching.

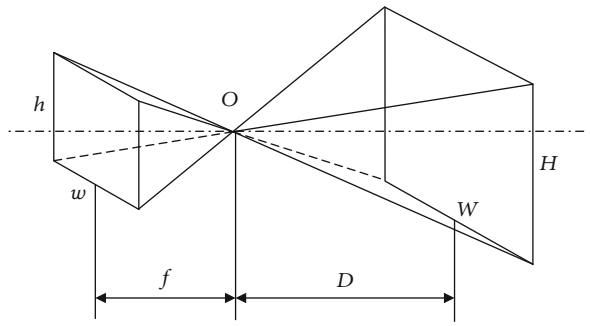

Figure 6: Camera pin-hole imaging model.

respectively. f_1 and f_2 are the focal lengths of the short focal length camera and the long focal length camera, respectively.

The image captured by a long focal length camera is scaled to match with the short focal length image according to the proportional coefficient by Equation (7). Based on the IoU of the vehicle detection bounding boxes in the two images, the box with the maximum IoU in the scaled long focal length picture is the object matched with the tracked vehicle.

The algorithm for finding a vehicle bounding box in the long focal length image that matches the tracked vehicle in the short focal length image is shown in Algorithm 1.

After determining the tracked vehicle, it is also necessary to match the license plate of the corresponding vehicle to obtain the actual vehicle width. We can match the license plate and the vehicle based on the size of the detection box. However, for a road environment with many vehicles and overlapping, there may be a wrong match. From the prior knowledge, we can know that each license plate only corresponds to one car, and the vehicle closest to the self-vehicle has the most features. Therefore, if a license plate bounding box falls within two vehicle detection boxes, the corresponding vehicle should be the closest one, and the image shows that the detection box of the vehicle is closer to the image bottom. Finally, the matching algorithm between the license plate and the corresponding vehicle is shown in Algorithm 2.

After the license plate is matched with the corresponding vehicle, the width of the vehicle can be calculated by the width of the license plate.

4. Experiment

In order to verify the performance of the vehicle detection and ranging method proposed in this paper, including the detection accuracy and speed of the Lightweight YOLO network, the stability of the vehicle tracking algorithm, and the accuracy of the long and short focal length cameras fusion ranging method, the road experiment was carried out. The experiment was implemented on the NVIDIA Jetson Xavier [40]. The camera was installed inside the windshield with a height of 1.3 m. The experimental equipment installation is shown in Figure 7.

The camera image sensor used in the experiment was OV10635 [41], with a pixel resolution of 1280×720 and a sensor image area of 5.5104mm × 3.4188mm. On the imaging area with width w', the pixel width corresponding to the imaging area width w_{m} is

$$w_{\mathrm{pixel}} = 1280 \frac{w_{\mathrm{m}}}{w'}. \tag{8}$$

Input: Vehicle detection bounding box $B^c = (x_1^c, y_1^c, x_2^c, y_2^c)$,
License plate bounding box $B^s = (x_1^s, y_1^s, x_2^s, y_2^s)$.
Output: the license plate B^s matched with the corresponding vehicle B^c.
For each B^c sorted by y_2^c from large to small:
If $B^s \subset B^c \Rightarrow \begin{pmatrix} x_1^s > x_1^c, y_1^s > y_1^c \\ x_2^s < x_2^c, y_2^s < y_2^c \end{pmatrix}$:
B^s matches B^c.
End if
End for

Algorithm 2: Vehicle and license plate matching.

Figure 7: Road experiment equipment installation.

Simultaneously, Equations (6) and (8) can obtain

$$w_{\text{pixel}} = 1280\frac{f_m W}{Dw'}, \tag{9}$$

where f_m represents the camera's actual focal length.

After testing, the network in this paper can detect the object with a width greater than 15 pixels, and we want to detect the object within 100 m. According to Equation (9), in order to detect the license plate with a width of 440 mm, the actual focal length should be longer than 14.67 mm. Therefore, the long and short focal lengths were selected as 6 and 16 mm, respectively, which can meet the needs of fusion ranging.

4.1. Vehicle and License Plate Detection and Vehicle Tracking Algorithm Experiment. Firstly, through the data screening and self-collection, an image data set containing different types of vehicles and license plates was established, a total of 30000, of which 24000 used as a training set and 6000 used as a test set. The training set contains a total of 45608 vehicle targets and 21888 license plate targets. The test set contains a total of 10801 vehicle targets and 5093 license plate targets. All images were labeled with yolo-mark [42].

The Lightweight YOLO model proposed in this paper was implemented through the PyTorch framework, and the Lightweight YOLO and YOLOv3 network models were trained on the NVIDIA GTX 1080Ti. The initial learning rate of the network is 0.001, the weight attenuation coefficient is 0.0005, and the training strategy uses a random gradient descent algorithm with a momentum term of 0.9.

In order to verify the performance of the Lightweight YOLO network, the detection results were compared to YOLOv3. The evaluation index of the experiment selects precision, recall, and calculation time per frame.

$$\begin{aligned} \text{precision} &= \frac{\text{TP}}{\text{TP} + \text{FP}}, \\ \text{recall} &= \frac{\text{TP}}{\text{TP} + \text{FN}}, \end{aligned} \tag{10}$$

where TP indicates that the number of targets is correctly detected, FN indicates that the number of targets is not detected, and FP indicates that the number of targets is erroneously detected. When the IoU of the bounding box in the network detection target box and the test set label data is greater than or equal to the set threshold, it is considered to be correct detection. Otherwise, it is regarded as error detection, and the experimental threshold is taken 0.5 [43].

The Lightweight YOLO and the YOLOv3 were tested in the test set, and the results are shown in Table 1. Compared with YOLO, the average precision and recall of the Lightweight YOLO network decreased by 4.43% and 3.54%, respectively, but the number of parameters is a quarter. The license plate target is relatively small, so its precision is slightly lower than that of the vehicle. However, the detection speed of each frame is increased by 49 ms, which achieved

Table 1: Comparison of network detection results.

Network	Classification	Precision (%)	Recall (%)	Calculation time per frame (ms)	Parameter number
YOLOv3	Vehicle	94.76	86.25	93.5	6.19e + 7
	License plate	91.21	82.54		
Lightweight YOLO	Vehicle	90.38	82.87	44.5	1.5e + 7
	License plate	86.81	78.91		

real-time vehicle detection on Jetson Xavier while still ensuring a higher detection accuracy.

In order to verify the detection effect after adding the tracking algorithm, the effect of the tracking algorithm was also tested. We chose five videos, each of which has 350 frames, and we labeled the ground truth with yolo-mark.

Considering that the tracking algorithm used in this paper is based on the detection results, we tested the precision and recall of the bounding box generated by the Lightweight YOLO and the tracking algorithm, as shown in Table 2. Compared with the Lightweight YOLO, the precision and recall of the tracking algorithm increased by 1.56% and 2.11%, respectively. Although the test sample size is small, it can still prove that the detection effect after adding the tracking algorithm is better, which is due to the tracking algorithm reduces false detections and missed detections.

There are 39 vehicles in the test sequence. In the experiment, IDswitch is 52, which means that the tracking algorithm has been interrupted 52 times. Most of the IDswitch are due to the tracked vehicle was wholly obscured, and fully visible vehicles can be continuously tracked. Given that the obscured vehicle is not a high-priority ranging object, the impact is small. However, this is also one of the directions for future improvement. After adding the tracking algorithm, the computation time of each frame is increased by 5.4 ms, and the whole vehicle detection and tracking algorithm can still run in real time.

Figure 8 shows the results of vehicle detection and tracking. It can be seen that the Lightweight YOLO can accurately detect the vehicle and license plate in the image. On the basis of the accurate detection of vehicles, the multitarget tracking method based on detection can also track vehicles stably.

4.2. Fusion Ranging Experiment. Firstly, the effect of the vehicle detection boxes matching algorithm in long and short focal images is verified, as shown in Figure 9. Figure 9(c) shows that the vehicle detection box in the long focal length image. Figure 9(b) is scaled by Algorithm 1 and almost overlaps with the corresponding vehicle detection box in the short focal length image Figure 10(a). The IoU of two vehicle detection boxes are 0.79 and 0.71, respectively, which verifies that the scaling relationship between long focal length to short focal length is correct. The experiment verified that the detection boxes in different focal length images can be successfully matched through Algorithm 1. The deviation of IoU in the matching process mainly comes from the jitter of the two cameras caused by vehicle bumps and the detection accuracy of different size vehicle images.

Table 2: Comparison of vehicle tracking results.

Algorithm	Precision (%)	Recall (%)	IDswitch	Calculation time per frame (ms)
Lightweight YOLO	89.16	81.65	—	45.2
After tracking	90.32	83.76	52	50.6

Then, in order to verify the dynamic performance of the long and short focal length fusion ranging method, four sets of comparative experiments were carried out. In Experiment 1, the vehicle in front was gradually away from the self-vehicle, and the self-vehicle was stationary, which is the case where the pixel width of the vehicle tracked for the first time is too large in the ranging algorithm. In Experiment 2, the vehicle in front was getting closer to the self-vehicle, which is that the pixel width of the vehicle tracked for the first time is small, and the license plate cannot be detected accurately. In Experiments 3 and 4, the target vehicle is in the side lane, and the self-vehicle and the target vehicle are moving relatively. The comparisons between the long and short focal length fusion ranging results and the vehicle position-based ranging result and the actual distance are shown in Figures 10–13, where the vehicle position-based ranging algorithm used the method proposed in [31], and the actual distance was detected by the radar.

First of all, it can be seen that the stability of the distance estimation method proposed in this paper is higher than the method based on the vehicle position. Especially in the dynamic environments of Experiment 2, when the distance is above 30 m, the results of the position-based method appear huge fluctuation, as shown in Figure 11, which completely deviates from the actual distance. In the static environment of Experiment 1, the results of the position-based method also show a significant deviation when the distance is greater than 60 m, as shown in Figure 10. In Experiment 3, the results of the position-based method appear fluctuation when the distance is greater than 40 m, as shown in Figure 12.

Then, in Experiments 1 and 2, the target vehicle was in the lane directly in front, and the ranging results are always accurate. However, in Experiments 3 and 4, it can be found that after a while, the ranging results show apparent deviation. Considering that the target vehicles of Experiments 3 and 4 were in the side lanes, due to the characteristics of the detection algorithm, the bounding box extracted by the detection algorithm is not only the rear of the vehicle but also

(a)

(b)

Figure 8: Detection and tracking results.

the side of the vehicle. When the vehicle on the side lane is close to the self-vehicle, the pixel width of the vehicle output by the detection algorithm is greater than the actual vehicle pixel width. As a result, the actual vehicle width calculated by Equation (5) is larger than the true actual vehicle width. As the target vehicle moved away, the error of the vehicle pixel width output by the detection algorithm became smaller, which makes the ranging results larger than the actual distance. In Figure 12, the ranging results of the proposed algorithm start to be significantly larger after 35 m. Similarly, in Figure 13, the ranging results are smaller than the ground truth in the later period.

Although some errors occurred in Experiments 3 and 4, the accuracy and robustness of the algorithm proposed in this paper are significantly better than the position-based ranging algorithm. Table 3 shows the comparison of ranging performance. The proposed method can reduce both the

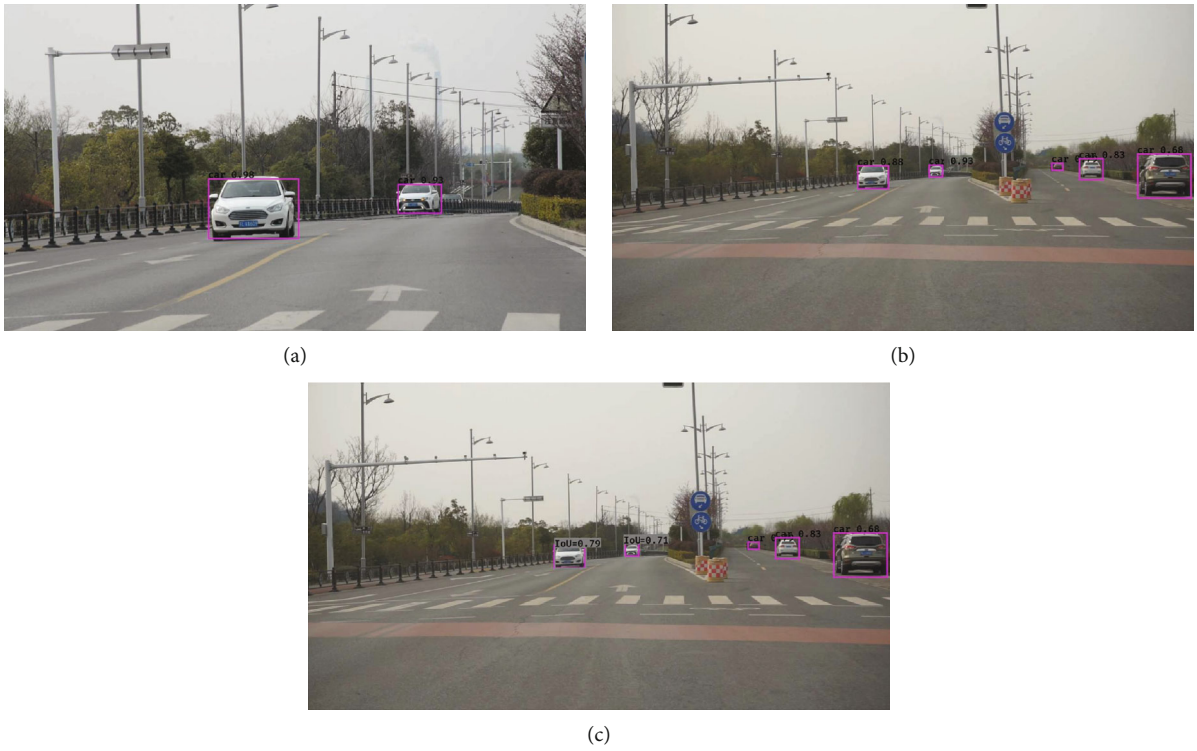

Figure 9: Long and short focal length vehicle detection box matching.

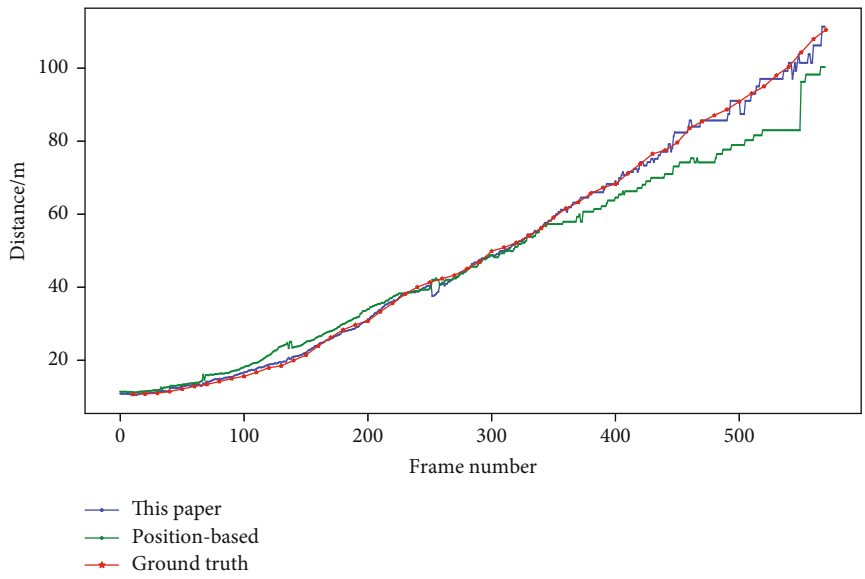

Figure 10: Experiment 1: same lane, self-vehicle stops.

mean error μ_d and the standard deviation σ_d. The mean error percentage μ_e shows that the relationship between the mean error percentage and the distance is small. The mean error percentage is less than 4%.

Figure 14 is the results of vehicle detection based on Lightweight YOLOv3 and long and short focal length camera fusion ranging proposed in this paper. The proposed method can accurately detect vehicles and estimate the distance between vehicles under various light and road conditions. In further analysis, the long and short focal length camera fusion ranging method proposed in this paper is based entirely on vehicle and license plate detection results, and the error is mainly derived from the inaccurate detection boxes, including the side vehicle detection box width beyond

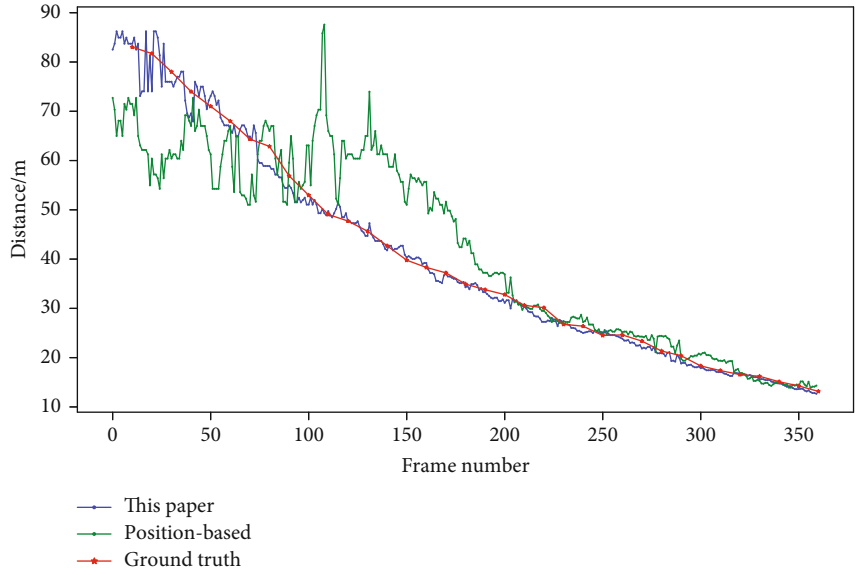

Figure 11: Experiment 2: same lane, move relatively.

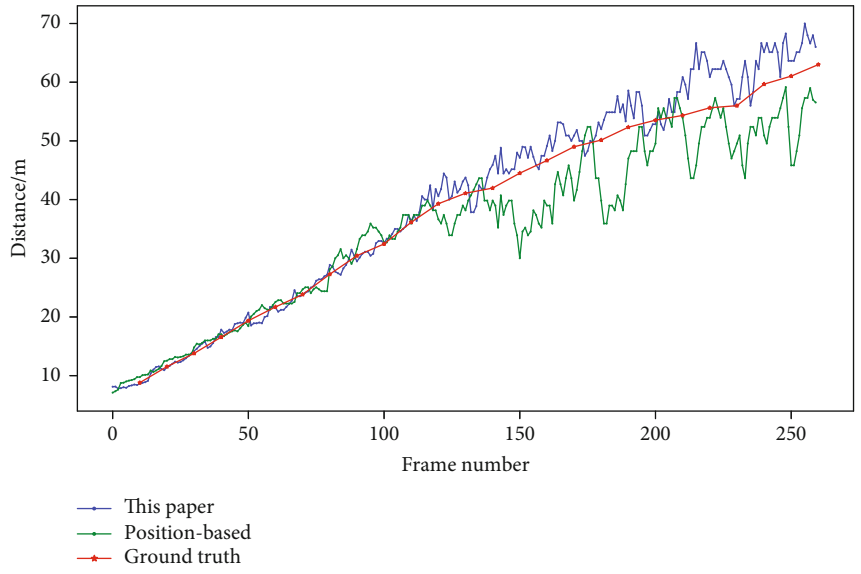

Figure 12: Experiment 3: different lanes, move relatively.

the actual vehicle width, distortion of side license plate width and the detection box of the distant vehicle width does not change significantly.

5. Conclusion

This paper proposes the Lightweight YOLO object detection model and the long and short focal length camera fusion ranging method. The lightweight network Shufflenet is integrated into the YOLOv3 network, and the Lightweight YOLO network is constructed. The parameter quantity is only one quarter of the original network, and the improved generalized IoU loss is added to the loss function. The precision and recall of the proposed method are slightly lower than YOLOv3, but the calculation speed is greatly improved, which achieved the demand for real-time detection. Besides, the fusion ranging method uses the license plate width to calculate the actual vehicle width and estimates the distance. Through the method of the long and short focal length fusion matching, it is possible to detect the license plate with a long distance and expand the range of ranging. The integration of the tracking algorithm also can detect the license plate once to determine the width of the tracked vehicle and reduces the calculation amount of the fusion matching algorithm. The experimental results show that the Lightweight YOLO object detection model and the long and short focal length

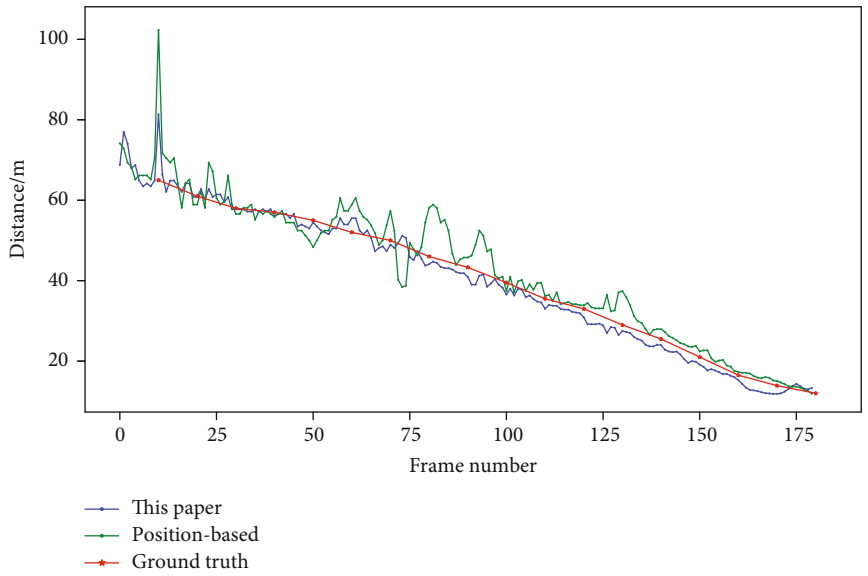

Figure 13: Experiment 4: different lanes, move relatively.

Table 3: Comparison of ranging results.

Distance	Position-based			This paper		
	μ_d (m)	σ_d (m)	μ_e (%)	μ_d (m)	σ_d (m)	μ_e (%)
0–40 m	2.19	2.88	9.30	0.79	0.68	3.77
>40 m	7.13	6.07	11.12	1.80	1.90	3.00
Total	4.77	5.41	10.25	1.32	1.54	3.37

Figure 14: Results of the detection and ranging algorithm.

cameras fusion ranging method can achieve stable real-time operation on the vehicle-embedded device. Subsequent research needs to improve the precision of the vehicle and license plate detection further and integrate other ranging methods to accurately sense vehicle position when the license plate cannot be detected.

Data Availability

All data included in this study are available upon request by contact with the corresponding author.

Conflicts of Interest

The authors declared no potential conflicts of interest with respect to the research, authorship, and publication of this article.

References

[1] Editorial Department of China Journal of Highway and Transport, "Review on China's automotive engineering research progress:2017," *China Journal of Highway and Transport*, vol. 30, no. 6, pp. 1–197, 2017.

[2] M. Rezaei and R. Klette, *Computer vision for driver assistance*, Springer International Publishing, 2017.

[3] Y. Cui, H. Xu, J. Wu, Y. Sun, and J. Zhao, "Automatic vehicle tracking with roadside LiDAR data for the connected-vehicles system," *IEEE Intelligent Systems*, vol. 34, no. 3, pp. 44–51, 2019.

[4] K. Yoneda, N. Hashimoto, R. Yanase, M. Aldibaja, and N. Suganuma, "Vehicle localization using 76GHz omnidirectional millimeter-wave radar for winter automated driving," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, pp. 971–977, Changshu, China, June 2018.

[5] E. Odat, J. S. Shamma, and C. Claudel, "Vehicle classification and speed estimation using combined passive infrared/ultrasonic sensors," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 5, pp. 1593–1606, 2017.

[6] Y. Fang, H. Zhao, H. Zha, X. Zhao, and W. Yao, "Camera and LiDAR fusion for on-road vehicle tracking with reinforcement learning," in *2019 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1723–1730, Paris, France, June 2019.

[7] A. Asvadi, L. Garrote, C. Premebida, P. Peixoto, and U. J. Nunes, "Multimodal vehicle detection: fusing 3D-LIDAR and color camera data," *Pattern Recognition Letters*, vol. 115, pp. 20–29, 2018.

[8] G. P. Stein, O. Mano, and A. Shashua, "Vision-based ACC with a single camera: bounds on range and range rate accuracy," in *IEEE IV2003 Intelligent Vehicles Symposium. Proceedings (Cat. No. 03TH8683)*, pp. 120–125, Columbus, OH, USA, June 2003.

[9] E. Dagan, O. Mano, G. P. Stein, and A. Shashua, "Forward collision warning with a single camera," in *IEEE Intelligent Vehicles Symposium, 2004*, pp. 37–42, Parma, Italy, June 2004.

[10] M. Ibarra-Arenado, T. Tjahjadi, J. Pérez-Oria, S. Robla-Gómez, and A. Jiménez-Avello, "Shadow-based vehicle detection in urban traffic," *Sensors*, vol. 17, no. 5, p. 975, 2017.

[11] S. S. Teoh and T. Bräunl, "Symmetry-based monocular vehicle detection system," *Machine Vision and Applications*, vol. 23, no. 5, pp. 831–842, 2012.

[12] L. W. Tsai, J. W. Hsieh, and K. C. Fan, "Vehicle detection using normalized color and edge map," *IEEE Transactions on Image Processing*, vol. 16, no. 3, pp. 850–864, 2007.

[13] T. K. Ten Kate, M. B. Van Leewen, S. E. Moro-Ellenberger, B. J. F. Driessen, A. H. G. Versluis, and F. C. A. Groen, "Mid-range and distant vehicle detection with a mobile camera," in *IEEE Intelligent Vehicles Symposium, 2004*, pp. 72–77, Parma, Italy, June 2004.

[14] K. Robert, "Night-time traffic surveillance: a robust framework for multi-vehicle detection, classification and tracking," in *2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance*, pp. 1–6, Genova, Italy, September 2009.

[15] Q. Zou, H. Ling, S. Luo, Y. Huang, and M. Tian, "Robust nighttime vehicle detection by tracking and grouping headlights," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 5, pp. 2838–2849, 2015.

[16] L. C. Liu, C. Y. Fang, and S. W. Chen, "A novel distance estimation method leading a forward collision avoidance assist system for vehicles on highways," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 4, pp. 937–949, 2017.

[17] Y. Wei, Q. Tian, J. Guo, W. Huang, and J. Cao, "Multi-vehicle detection algorithm through combining Harr and HOG features," *Mathematics and Computers in Simulation*, vol. 155, pp. 130–145, 2019.

[18] P. H. Batavia, D. E. Pomerleau, and C. E. Thorpe, "Overtaking vehicle detection using implicit optical flow," in *Proceedings of Conference on Intelligent Transportation Systems*, pp. 729–734, Boston, MA, USA, November 1997.

[19] Y. B. Chin, L. W. Soong, L. H. Siong, and W. W. Kit, "Extended fuzzy background modeling for moving vehicle detection using infrared vision," *IEICE Electronics Express*, vol. 8, no. 6, pp. 340–345, 2011.

[20] Y. Fang and B. Dai, "An improved moving target detecting and tracking based on optical flow technique and Kalman filter," in *2009 4th International Conference on Computer Science & Education*, pp. 1197–1202, Nanning, China, July 2009.

[21] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 1, pp. 142–158, 2016.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in *Computer Vision – ECCV 2014. ECCV 2014. Lecture Notes in Computer Science, vol 8691*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., pp. 346–361, Springer, Cham, 2014.

[23] S. REN, K. He, R. Girshick, and J. Sun, "Faster r-cnn: towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems 28 (NIPS 2015)*, pp. 91–99, Montreal, Canada, December 2015.

[24] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once:unified, real-time object detection," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, Las Vegas, NV, USA, June 2016.

[25] W. Liu, D. Anguelov, D. Erhan et al., "Ssd: Single shot multibox detector," in *Computer Vision – ECCV 2016. ECCV 2016. Lecture Notes in Computer Science, vol 9905*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., pp. 21–37, Springer, Cham, 2016.

[26] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7263–7271, Honolulu, HI, USA, July 2017.

[27] J. Sang, Z. Wu, P. Guo et al., "An improved YOLOv2 for vehicle detection," *Sensors*, vol. 18, no. 12, p. 4272, 2018.

[28] G. Toulminet, M. Bertozzi, S. Mousset, A. Bensrhair, and A. Broggi, "Vehicle detection by means of stereo vision-based obstacles features extraction and monocular pattern analysis," *IEEE Transactions on Image Processing*, vol. 15, no. 8, pp. 2364–2375, 2006.

[29] R. Adamshuk, D. Cravalho, J. H. Z. Neme et al., "On the applicability of inverse perspective mapping for the forward distance estimation based on the HSV colormap," in *2017 IEEE International Conference on Industrial Technology (ICIT)*, pp. 1036–1041, Toronto, ON, Canada, March 2017.

[30] J. Han, O. Heo, M. Park, S. Kee, and M. Sunwoo, "Vehicle distance estimation using a mono-camera for FCW/AEB systems," *International Journal of Automotive Technology*, vol. 17, no. 3, pp. 483–491, 2016.

[31] M. Rezaei, M. Terauchi, and R. Klette, "Robust vehicle detection and distance estimation under challenging lighting conditions," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 5, pp. 2723–2743, 2015.

[32] D. Zhao, Y. Yang, J. Huang, and Y. Liu, "Vehicle position estimation using geometric constants in traffic scene," in *Proceedings of 2014 IEEE International Conference on Service Operations and Logistics, and Informatics*, pp. 90–95, Qingdao, China, October 2014.

[33] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, http://arxiv.org/abs/1804.02767.

[34] N. Ma, X. Zhang, H. T. Zheng, and J. Sun, "Shufflenet v2: Practical guidelines for efficient cnn architecture design," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 116–131, Munich, Germany, September 2018.

[35] F. Chollet, "Xception: deep learning with depthwise separable convolutions," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1251–1258, Honolulu, HI, USA, July 2017.

[36] H. Rezatofighi, N. Tsoi, J. Y. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: a metric and a loss for bounding box regression," 2019, http://arxiv.org/abs/1902.09630.

[37] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 3464–3468, Phoenix, AZ, USA, September 2016.

[38] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[39] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[40] "NVIDIA JETSON AGX XAVIER," https://www.nvidia.com/en-us/autonomous-machines/jetson-agx-xavier/.

[41] "OV10635," https://www.ovt.com/sensors/OV10635/.

[42] AlexeyAB, *Yolo_mark*https://github.com/AlexeyAB/Yolo_mark/.

[43] M. Everingham, L. van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.