machines

MDPI

Article

An Improved Approach for Real-Time Taillight Intention Detection by Intelligent Vehicles

Bingming Tong [1], Wei Chen [1], Changzhen Li [2], Luyao Du [1,*], Zhihao Xiao [1] and Donghua Zhang [3]

[1] School of Automation, Wuhan University of Technology, Wuhan 430070, China; 307311@whut.edu.cn (B.T.); greatchen@whut.edu.cn (W.C.); 320940@whut.edu.cn (Z.X.)
[2] School of Information Engineering, Wuhan University of Technology, Wuhan 430070, China; changzhen.li@whut.edu.cn
[3] Wuhan Zhongyuan Electronics Group Co., Ltd., Wuhan 430070, China; zhangdonghua@710g.com
* Correspondence: duluyao@whut.edu.cn; Tel.: +86-158-2715-3041

Abstract: Vehicle taillight intention detection is an important application for perception and decision making by intelligent vehicles. However, effectively improving detection precision with sufficient real-time performance is a critical issue in practical applications. In this study, a vision-based improved lightweight approach focusing on small object detection with a multi-scale strategy is proposed to achieve application-oriented real-time vehicle taillight intention detection. The proposed real-time detection model is designed based on YOLOv4-tiny, and a spatial pyramid pooling fast (SPPF) module is employed to enrich the output layer features. An additional detection scale is added to expand the receptive field corresponding to small objects. Meanwhile, a path aggregation network (PANet) is used to improve the feature resolution of small objects by constructing a feature pyramid with connections between feature layers. An expanded dataset based on the BDD100K dataset is established to verify the performance of the proposed method. Experimental results on the expanded dataset reveal that the proposed method can increase the average precision (AP) of vehicle, brake, left-turn, and right-turn signals by 1.81, 15.16, 40.04, and 41.53%, respectively. The mean average precision (mAP) can be improved by 24.63% (from 62.20% to 86.83%) at over 70 frames per second (FPS), proving that the proposed method can effectively improve detection precision with good real-time performance.

Keywords: intelligent vehicle; taillight intention; real-time detection; multi-scale feature

Citation: Tong, B.; Chen, W.; Li, C.; Du, L.; Xiao, Z.; Zhang, D. An Improved Approach for Real-Time Taillight Intention Detection by Intelligent Vehicles. *Machines* **2022**, *10*, 626. https://doi.org/10.3390/machines10080626

Academic Editors: Antonios Gasteratos and Ioannis Kostavelis

Received: 28 June 2022
Accepted: 27 July 2022
Published: 29 July 2022

1. Introduction

The driving safety of vehicles greatly concerns both the public and researchers. With the development of artificial intelligence algorithms [1–3] and internet-of-vehicles technology [4,5], intelligent vehicles can perceive their surrounding environment more effectively. In real traffic scenes, most information transmitted between vehicles is signaled by vehicle lights. Therefore, vehicle taillight signals represent indispensable information in driving behavior decisions such as lane-changes [6,7] and overtaking.

Over the last decade, vehicle taillight intention detection has been widely studied, combining knowledge-based methods with statistical machine learning models. These traditional methods have performed well owing to the handcrafted features of taillight color and structure designed in the specific scenarios studied. However, in the real world, traditional methods may not be able to adapt to changing environments consistently and effectively due to changes in the driving environment (i.e., lighting conditions, occlusion, road slope). As a result, the generalization ability and robustness of the traditional methods need to be further improved.

In recent years, deep learning has received more attention in the field of computer vision. Traditional algorithms such as HOG [8], SIFT [9], and LBP [10,11] based on handcrafted features have turned to machine learning techniques based on deep neural networks.

Convolutional Neural Networks (CNN) have been widely applied in the field of computer vision for image classification (e.g., GhostNet [12], ResNet [13]), semantic segmentation (e.g., Mask R-CNN [14]), and object detection (e.g., SSD [15], Faster R-CNN [16]). In the real world, taillight intention detection is essentially regarded as a type of multi-object detection, which can be achieved using a CNN-based object detector. In general, CNN-based object detectors are mainly divided into two categories. The first consists of two-stage detectors based on region proposal generation, represented by the R-CNN series [14,16,17]; the proposal generation approach improves detection precision at the cost of more inference time. The second category consists of one-stage detectors that refactor object detection into a regression task, represented by the YOLO series [18–22] and SSD. These algorithms directly calibrate grids on the output layers of different sizes, then generate the category score of the object in regressed grids.

Successes have been achieved in applying deep learning to vehicle taillight intention detection. Nevertheless, there is room for improvement. Meanwhile, in order to improve the precision of vehicle taillight intention detection, the high quality of datasets should be guaranteed. Numerous standard datasets (e.g., BDD [23], KITTI [24], and Bosch small traffic lights [25]) have been established for the transportation industry. However, to the best of our knowledge, there is no public dataset dedicated to the detection of vehicle taillights.

In this paper, an end-to-end method is proposed to detect rear braking and turn signals from video streams in real-time. To this end, YOLOv4-tiny is used as the base model, and three strategies are adopted to balance efficiency and effectiveness with less computational resource occupancy.

The main contributions can be summarized as follows:

- An expanded dataset based on BDD100K is established for vehicle taillight intention detection, which includes 3316 challenging images under different roads, time periods, lighting, and weather conditions in real traffic scenes.
- A lightweight model for real-time vehicle taillight intention detection is proposed. An SPPF module that can be used to enrich the deep semantic information is combined with a CSPDarknet53-tiny backbone to improve the performance of taillight intention detection.
- A multi-scale detection strategy based on the lightweight model is proposed to expand the receptive field focusing on small objects. PANet is leveraged to utilize contextual information to further improve the resolution of small objects such as turn signals.

The rest of the article is organized as follows. In Section 2, related works on taillight detection are reviewed. Section 3 presents the experimental dataset. In Section 4, our proposed method is described in detail. Section 5 provides the experimental results and analysis of the proposed method performed on the modified dataset. Finally, conclusions are drawn in Section 6.

2. Related Works

Traditional knowledge-based taillight detection methods utilize handcrafted features to match and classify them with statistical machine learning classifiers. Among them, Chen et al. [26] proposed a vision-based method employing a fast radial symmetry transform algorithm to match the taillights symmetrically for daytime brake light detection. Cui et al. [27] developed a layered framework for detecting vehicle taillight signals. The first layer detected vehicles with deformable part models, while the second layer extracted taillight candidates using clustering techniques. The last layer used sparse representation to estimate taillight states.

Leading up to the current study, researchers have been gradually shifting their research interest in vehicle taillight intention detection from traditional knowledge-based methods to deep learning methods. These can be roughly divided into two categories.

Methods in the first category split vehicle taillight intention detection into vehicle localization and taillight state estimation. Nava et al. [28] leveraged the YOLO detector to detect the front vehicle. Then, the brake signal state was determined through the SVM

classifier. Zhong et al. [29] first localized each vehicle using a Fast R-CNN detector with kernel correlation tracking. The position of the brake light area was obtained by a fine-tuned fully convolutional networks and the SVM classifier was used to determine the brake light state. Vancea et al. [30] presented a convolutional neural network architecture composed of a Faster R-CNN for detecting vehicles and a sub-network for classifying the obtained pairs of taillights within the vehicle. This type of approach uses a deep learning model to detect vehicles in real scenes, followed by taillight detection using a separate classifier (i.e., statistical machine learning, deep learning) without end-to-end detection.

The second category of methods performs vehicle taillight intention detection in an end-to-end fashion. Hsu et al. [31] proposed a method to build two classifiers based on the CNN-LSTM model to identify vehicle taillight signals. Brake signal spatial features were extracted through the CNN. For turn signals, the differences between images in consecutive frames with ROI regions were input to the model for state recognition. Frossard et al. [32] proposed an architecture to detect taillight signals directly. A fully convolutional network was added to the CNN-LSTM model in this architecture to generate masks, removing the influence of unnecessary spatial features. Lee et al. [33] improved detection precision by integrating an attention model in their CNN-LSTM network, where the learning process of the attention model was selectively focused on the spatial and temporal features. However, these approaches have only been validated on sequences of cropped vehicle images.

Different from the above methods, the proposed method models taillight detection as a regression problem and improves the performance of YOLOv4-tiny for small objects. With a multi-scale strategy, the proposed method can achieve effective performance in real taillight detection scenarios. Meanwhile, the proposed method seamlessly integrates YOLOv4-tiny, SPPF, and PANet to provide an end-to-end trainable network for taillight detection.

3. Dataset Establishment

In this section, we establish an expanded dataset based on BDD100K with three categories for vehicle taillights intention detection, including vehicle, brake signal, and turn signal. The turn signal category is further classified into left and right turn signals during the decoding process.

3.1. Data Acquisition

An expanded dataset with 3316 images was established based on selected images from the BDD100K dataset. The collection work sought to emphasize different roads, time periods, lighting, and weather conditions. All the images were from a car-front camera. Samples of the dataset are shown in Figure 1.

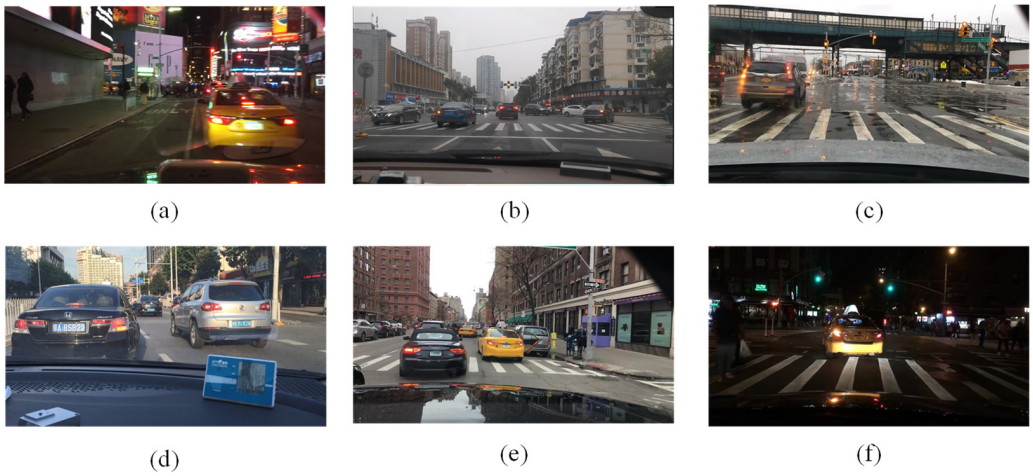

Figure 1. Samples of the vehicle taillight detection dataset. (**a**) Multi-light source scenario. (**b**) Crossroad scenario. (**c**) Rain scenario. (**d**) Lane change scenario. (**e**) Simple scenario. (**f**) Night scenario.

The datasets used in machine learning methods often include training, test, and validation sets. Consistent with the YOLOV4-tiny ratio [22], 2685, 332, and 299 images were selected as the training set, test set, and validation set, respectively. The training set was used to train the method weights. The test set was used to verify the detection performance of the proposed method on unseen data. The validation set was used to prevent over-fitting and tune the hyper-parameters.

3.2. Data Augmentation

The richness of the dataset plays a vital role in experimental results. To increase the richness of the dataset, data augmentation operations can be performed on images. Mirroring of the images can enlarge the sample size of the dataset. When the camera encounters dark lighting and electromagnetic interference, noise appears as a random distribution of bright spots on the image. Neural networks are not robust to noise, and it can be advantageous to employ models which have learned using noisy data [34]. Therefore, adding impulse noise to dataset images aims to improve the robustness of the method. Mosaic data augmentation on the images can be further employed to enrich the dataset. New training images are formed by flipping, scaling, and changing the color space of the four images to enrich the background of detected objects, thereby enhancing the generalization ability of the trained method.

3.3. Data Annotation

The images classified by the three categories (vehicle, brake, turn) were labelled manually after being numbered. Bounding boxes were drawn manually according to the VOC data format. LabelImg software, which provides the location and category information of objects, was used for data annotation. Images with the occlusion areas of objects more than 50% and unlabeled objects tend to be extremely small in an image. The work of data pre-processing was carried out through the above stages. The dataset statistics and the number of objects in each category are shown in Table 1.

Table 1. Vehicle taillight detection dataset.

	Vehicle	**Brake**	**Turn**
Training set	6538	1662	1734
Test set	808	201	231
Validation set	728	179	197
Total	8074	2042	2162

4. Proposed Method

In this section, an improved method to detect vehicle taillight intention more effectively by combining YOLOv4-tiny architecture with three modifications is proposed.

4.1. YOLOv4-Tiny Model

Aiming at the real-time requirements of vehicle taillight intention detection, our method was designed based on YOLOv4-tiny. YOLOv4-tiny, which combines two detection layers for feature fusion, is a lightweight model based on the CSPDarknet-tiny backbone.

Because YOLOv4-tiny is a lightweight model, its precision performance has difficulty meeting the detection requirements. To address this issue, three strategies were applied to improve its precision performance. First, the SPPF module was adapted for the YOLO-tiny model to extract different scale features from the same layer. Second, a 52×52 scale output layer was added to provide fine-grained features in multi-scale detection. Third, PANet was employed for feature fusion in the multi-scale output layer.

4.2. SPPF Module

The SPPF module decreases computational resource occupancy in the spatial pyramid pooling (SPP) module [35] by replacing the 5 × 5, 9 × 9, and 13 × 13 max-pooling layers with three 5 × 5 max-pooling layers. The max-pooling layer size represents the number of parameters required for calculation. Therefore, the parameter quantity of the SPPF model is less than half that of the SPP model. The SPPF module schematic is illustrated in Figure 2.

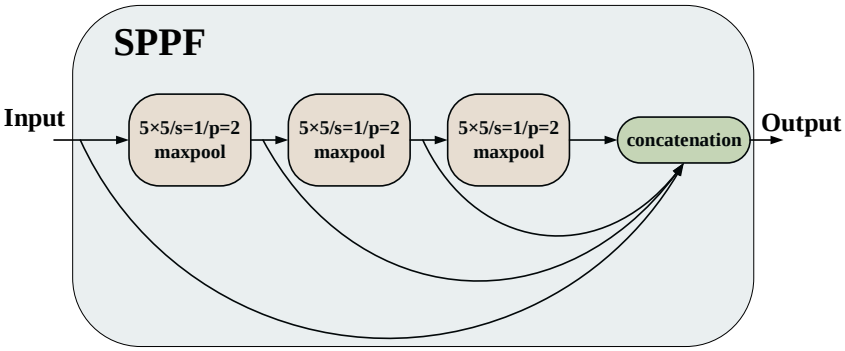

Figure 2. Schematic of the SPPF module.

In our proposed method, the SPPF module is inserted after the last convolutional layer of the CSPDarknet53-tiny backbone. The SPPF module was applied in YOLOv4-tiny to obtain features of different scales through multiple pooling, effectively avoiding incomplete cropping and shape distortion of image objects caused by the cropping and scaling operations of convolution as well as increasing the receptive field. The input feature layer undergoes three instances of 5 × 5 max pooling, and is connected to the input layer after each pooling. After concatenation, the output of the SPPF module is used for 13 × 13 scale detection.

4.3. Multi-Scale Detection

Multi-scale detection is utilized in YOLOv4-tiny to detect objects of different sizes. However, in YOLOV4-tiny, only two scale feature layers (26 × 26, 13 × 13) are extracted for feature fusion to form the feature pyramid network (FPN) [36]. It is difficult for YOLOv4-tiny to regress the final position information of objects as small as 26 × 26 scale [37].

The backbone network increases the network layer to extract semantic information as a way of reducing the spatial scale [38]. Thus, it is generally believed that backbone networks contain higher spatial resolution feature maps on shallow feature layers and represent rich semantic information on deep feature layers. Shallow feature maps usually focus on small object detection, and deeper features are conducive to large object detection.

The size distribution of the object detection bounding box (BBOX) used in the dataset is shown in Figure 3. Notably, the dataset covered objects with large, medium, and small sizes. Small object detection is indispensable in vehicle taillight intention detection.

Based on the rationales above, we added an additional 52 × 52 detection scale in YOLOv4-tiny to improve detection precision for small objects. Feature maps from shallower layers have higher resolution with detailed spatial information, and outputs from deeper layers have smaller resolution with rich semantic features. Multi-scale detection can simultaneously utilize richer deep semantic information and shallow higher-resolution image spatial features. Specifically, the improved method was able to predict the BBOXs of three different scales, namely, 13 × 13, 26 × 26, and 52 × 52. The outputs of the 13 × 13 and 26 × 26 detection layers contained rich semantic information, whereas small object features were ignored during down-sampling. The 52 × 52 output divided more grid cells in the image, meaning that smaller image areas could be detected.

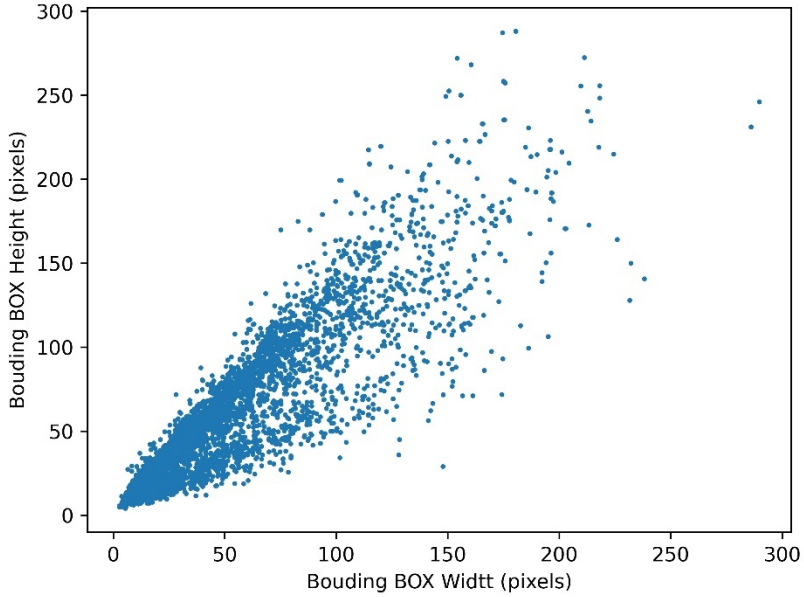

Figure 3. Objects size distribution of the vehicle taillight dataset.

4.4. PANet

PANet was first proposed in [39] to boost information flow in instance segmentation frameworks. PANet is considered an enhanced version of FPN. Here, it was used to enhance the representation capability of the proposed method by fusing down-up and top-down path augmentation. The PANet structure is shown in Figure 4.

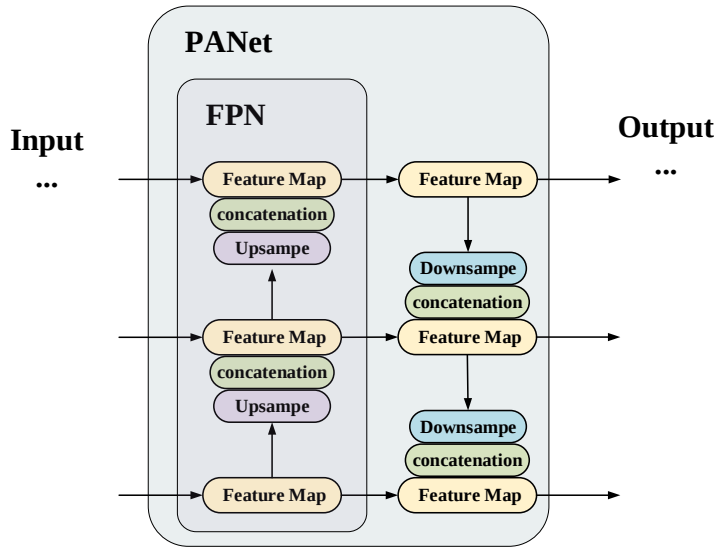

Figure 4. Schematic of the PANet module.

We used PANet to construct a feature pyramid with connections between feature layers to leverage contextual information passed both down-up and top-down. To improve the precision of small object detection, PANet was implemented in two aspects. The first aimed to fully exploit the deeper spatial information and shallower semantic information of the CSPdarknet_tiny network. The second was adopted to accommodate the feature fusion scale changes induced by the additional detection layer.

PANet selects different feature maps when predicting different objects, avoiding hard matching of object size and network depth. An additional 52×52 feature layer is extracted from the backbone network for multi-scale detection. PANet fully integrates the semantic

information from multi-scale detection to provide important features for small object detection, further improving the detection results.

4.5. Designed Model

The designed model for our method is shown in Figure 5. Am image with a pixel of 416 × 416 is input to the CSPDarknet-tiny backbone in the form (R, G, B). Feature maps with sizes of 52 × 52, 26 × 26, and 13 × 13 are extracted from the backbone network for multi-cross feature fusion and multi-scale detection.

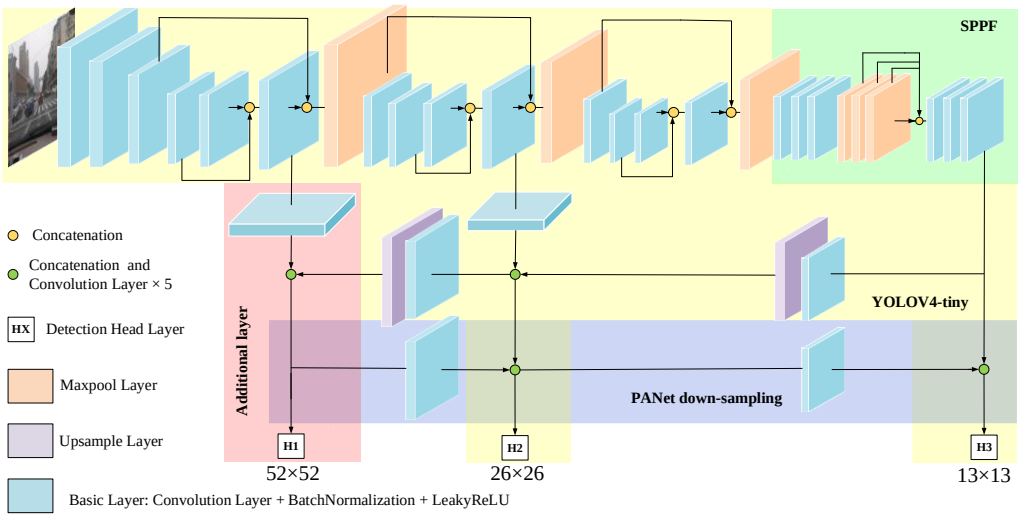

Figure 5. Network architecture of the designed model.

4.5.1. Multi-Cross Feature Fusion

The multi-cross feature fusion process can be divided into three steps. First, the 13 × 13 feature layer is pooled and concatenated three times through the SPPF module. Second, the pooled 13 × 13 feature layer is up-sampled twice to 52 × 52 scale and concatenated along the channel dimension with the corresponding feature layer extracted from the backbone network. Finally, the 52 × 52 feature layer is down-sampled twice by convolution transformation, and channel-wise concatenation is performed with the corresponding size feature layer. The maps that have undergone feature fusion are then sent to the corresponding detection head layers to output the results.

4.5.2. Loss Function

The loss function in the proposed method consists of three parts, regression loss, category probability loss, and confidence loss, which can be written as

$$\begin{aligned} Loss &= Loss_{loc} + Loss_{cls} + Loss_{conf} \\ &= \sum_{i=0}^{s^2} \sum_{j=0}^{B} 1_{ij}^{obj} CIoU \\ &+ \sum_{i=0}^{s^2} \sum_{j=0}^{B} \left(-1_{ij}^{obj}\right) \left[C_i^j \log\left(\hat{C}_i^j\right) + \left(1 - C_i^j\right) \log\left(1 - \hat{C}_i^j\right)\right] \\ &+ \sum_{i=0}^{s^2} \sum_{j=0}^{B} \left(-1_{ij}^{noobj}\right) \left[C_i^j \log\left(\hat{C}_i^j\right) + \left(1 - C_i^j\right) \log\left(1 - \hat{C}_i^j\right)\right] \\ &+ \sum_{i=0}^{s^2} \sum_{j=0}^{B} \left(-1_{ij}^{obj}\right) \sum_{c \in classes} \left[P_{i,c}^j \log\left(\hat{P}_{i,c}^j\right) + \left(1 - P_{i,c}^j\right) \log\left(1 - \hat{P}_{i,c}^j\right)\right], \end{aligned} \tag{1}$$

Here, there are S^2 grids in the input image, and each grid generates B BBOXs; 1_{ij}^{obj} and 1_{ij}^{noobj} represent the j-th BBOX in grid i, which contains and excludes objects, respectively;

$P_{i,c}^{j}$ and $\hat{P}_{i,c}^{j}$ represent the predicted value and true probability indicating that the object belonging to category c in grid i, respectively; C_i^j refers to the confidence of the predicted BBOX containing the object; and $\hat{C}_i^j$ denotes the intersection over union (IoU) of the real object and the BBOX. The IoU is a measure that quantifies the degree of overlap between two boxes, and is defined as follows:

$$\mathrm{IoU}(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cup B|}, \tag{2}$$

where $|\,|$ means the cardinality of the set and A and B represent two separate boxes.

The complete intersection over union (CIoU) loss function [40] was chosen for the regression loss. CIoU considers the distance between the target and the predicted box, the overlap rate, the scale, and the penalty terms, making the object BBOX regression more stable.

4.5.3. Decoding Process

Benefiting from the end-to-end approach, the input image is directly transformed into predicted coordinates and categories.

Specifically, the three detection layers have grids of 52 × 52, 26 × 26, and 13 × 13, respectively. Each grid is configured with three anchor boxes, outputting a total of 3 × (5 + 3) = 24 pieces of feature information. Each anchor box will have four coordinates, one confidence score, and three category (Vehicle, Brake, and Turn) probabilities. During the decoding process in Algorithm 1, anchor boxes for which the confidence score is less than 0.5 are ignored. The redundant information is filtered out by the Soft-NMS [41] algorithm. Then, the three output overlays are mapped to the input image size. Detected turn signals are further classified into Left-Turn (LT) and Right-Turn (RT) with the corresponding IoU vehicle box. Finally, taillight information that is not in the vehicle box is discarded.

Algorithm 1 Output Decoding Process

Input: O^1, O^2, and O^3 correspond to the output of the 13 × 13, 26 × 26, 52 × 52 detection layers.

Output: $O^m \in R^{l \times 6}$, l represents the number of results. Each result contains 4 coordinate, 1 confidence score, and 1 category symbol.

for n = 1 to 3 **do**
  $O^n \leftarrow \mathrm{Soft-NMS}(O^n - O^n_{\mathrm{confidence}<0.5})$
end for
$O^m \leftarrow \mathrm{map}(O^1, O^2, O^3)$
for $O^m_{\mathrm{category=Turn}}$ in O^m **do**
  for $O^m_{\mathrm{category=Vehicle}}$ in O^m **do**
    $O^m_{\mathrm{left}}, O^m_{\mathrm{right}} \leftarrow O^m_{\mathrm{category=Vehicle}}$
    if $\mathrm{IoU}(O^m_{\mathrm{right}}, O^m_{\mathrm{category=Turn}}) > 0.5$
      $O^m_{\mathrm{category=Right-Turn}} \leftarrow O^m_{\mathrm{category=Turn}}$
    else if $\mathrm{IoU}(O^m_{\mathrm{left}}, O^m_{\mathrm{category=Turn}}) > 0.5$
      $O^m_{\mathrm{category=Left-Turn}} \leftarrow O^m_{\mathrm{category=Turn}}$
    end if
  end for
  $O^m \leftarrow O^m - O^m_{\mathrm{category=Turn}}$
end for

5. Experiment and Result Analysis

In this section, the experimental evaluation metrics and environments are introduced. Then, the effectiveness of our method is verified. Ablation experiments are carried out to demonstrate the effect of different improvements. Three state-of-the-art object detectors are compared with the proposed method in terms of detection precision and rate. In

addition, the proposed method is deployed on real traffic scene detection to verify its stability and practicality.

5.1. Evaluation Metrics

The precision (P) and recall (R) are employed as two metrics for a more fair and reasonable comparison, as follows:

$$P = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}, \tag{3}$$

$$R = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}. \tag{4}$$

F1-score is the harmonic mean of precision and recall. It is a comprehensive evaluation index, as follows:

$$F1 - score = \left(\frac{P^{-1} + R^{-1}}{2}\right)^{-1}. \tag{5}$$

The precision recall (PR) curve is a visual metric for evaluating the performance of object detection algorithms. The closer the PR curve of the algorithm is to the upper right corner of the graph, the better the performance. The area under the PR curve is AP, which is used to quantitatively evaluate the performance of the algorithms and can be defined as

$$AP = \int_0^1 P(R)dR\,. \tag{6}$$

The mAP is an indicator of the performance of the overall situation. Typically, mAP can be defined as

$$mAP = \frac{1}{4}\sum_{c \in \text{categories}} AP(c)\,. \tag{7}$$

5.2. Experimental Environment

The experimental hardware environment consisted of a Windows PC with an Intel(R) Core (TM) i7-10750H CPU @ 2.60GHz 2.59 GHz, 16 G DDR4 of RAM, and an NVIDIA GeForce RTX 2070 with 8 GB of memory. The relevant software environment is shown in Table 2.

Table 2. Version parameters of the environment.

Environment	Python	Torch	CUDA	cuDNN
version	3.8.8	1.70	10.2	8.2.1

In the training strategy, the input images from the dataset were resized to 416 × 416 pixels. The pretrained weights of CSPdarknet53-tiny were used. After this configuration, the training process was optimized using Adam [42] with a momentum of 0.937. The maximum number of training iterations was set to be 300. The batch size was 24. The initial learning rate was set to be 0.001 and to decrease at a rate of a minimum value of one tenth of the initial learning rate as the iteration period increased.

We denoted AP@0.5 as the average precision under IoU, from 0.5 to 0.95. The detection confidence was set to 0.5, which is the default detection confidence of YOLOv4-tiny.

5.3. Ablation Experimental

The proposed method was compared with the baseline YOLOv4-tiny on the collected dataset. Ablation analyses were performed to investigate the impact of each modification. The detailed results are outlined in Tables 3 and 4.

Table 3. Precision, recall, and F1-score results of the ablation experiment on the test dataset.

Method	Precision (%)	Recall (%)	F1-Score (%)
YOLOv4-tiny [22]	74.72	53.09	60.25
YOLOv4-tiny-SPPF	77.27	58.35	65.75
YOLOv4-tiny-SPPF-3l	83.50	80.96	82.25
YOLOv4-tiny-SPPF-PANet-3l (ours)	91.78	87.52	89.50

Table 4. AP results of the ablation experiment on the test dataset.

Method	mAP (%) @0.5	AP (%) @0.5			
		Vehicle	Brake	LT	RT
YOLOv4-tiny [22]	62.20	96.48	74.76	35.59	41.95
YOLOv4-tiny-SPPF	65.18	97.86	78.06	39.54	45.25
YOLOv4-tiny-SPPF-3l	76.07	96.65	90.27	51.79	65.58
YOLOv4-tiny-SPPF-PANet-3l (ours)	86.83	98.29	89.92	75.63	83.48

YOLOv4-tiny-SPPF refers to the YOLOv4-tiny framework with employed SPPF module, in which AP performance gains for Vehicle, Brake, LT, and RT were 1.38, 3.3, 3.95, and 3.30%, respectively. The increase in mAP illustrates that it is beneficial to enrich the features of a feature layer by multiple pooling and concatenation of the same feature layer in the SPPF module.

YOLOv4-tiny-SPPF-3l extracted an additional 56×56 detection layer from the backbone network into the method, in which AP increased by 12.21, 12.25, and 20.33% for Brake, LT, and RT, respectively, while the AP for Vehicle dropped 1.21%. The experimental results show that multi-scale detection greatly improves the detection ability for small objects without contributing to large-sized objects.

As can be seen from Table 3, the recall rate of YOLOv4-tiny-SPPF-3l is greatly improved. The detection performance of small objects greatly benefits from the additional prediction layer. Early high-resolution CNN layer feature maps provide valuable information for locating small objects.

On the basis of YOLOv4-tiny-SPPF-3l, our method (YOLOv4-tiny-SPPF-PANet-3l) further integrated PANet. As shown in Table 4, the AP of our method increased by 1.64, 23.84, and 17.90% for Vehicle, LT, and RT respectively, while the AP for Brake fell by 0.35%. Because PANet fuses features from three detection layers of different sizes top-down and down-up, it can make full use of the contextual information of the backbone. Compared to the up-sampling feature fusion of the FPN in the baseline, the bidirectional sampling process of PANet fuses more fine-grained shallow features. The results show that the mechanism greatly enhances the detection ability for small objects, whereas the contribution for the situation of medium objects is moderate.

The PR curves for the four objects (Vehicle, Brake, LT, and RT) of the vehicle taillight detection dataset are shown in Figure 6. Compared with the PR curves of the baseline, the PR curves of all four categories obtained by our method are improved. For turn signals, our PR curves fully encompass the baseline, which verifies the detection ability of the proposed method for small objects.

The ultimate method, YOLOv4-tiny-SPP-PANet-3l, yielded a 24.63% mAP performance gain over the baseline YOLOv4-tiny on the test set. The AP of Vehicle, Brake, LT, and RT increased by 1.81, 15.16, 40.04, and 41.53%, respectively. Although significant progress has been made at the expense of smaller computation consumption, the improved method continues to meet the real-time requirements of most actual applications. The proposed method runs at 73 FPS on an RTX-2070 GPU, which is apparently lower than the 144 FPS of the YOLOv4-tiny. The memory size of the method is 47.0 MB, which meets the storage condition limitations of the vehicle platform.

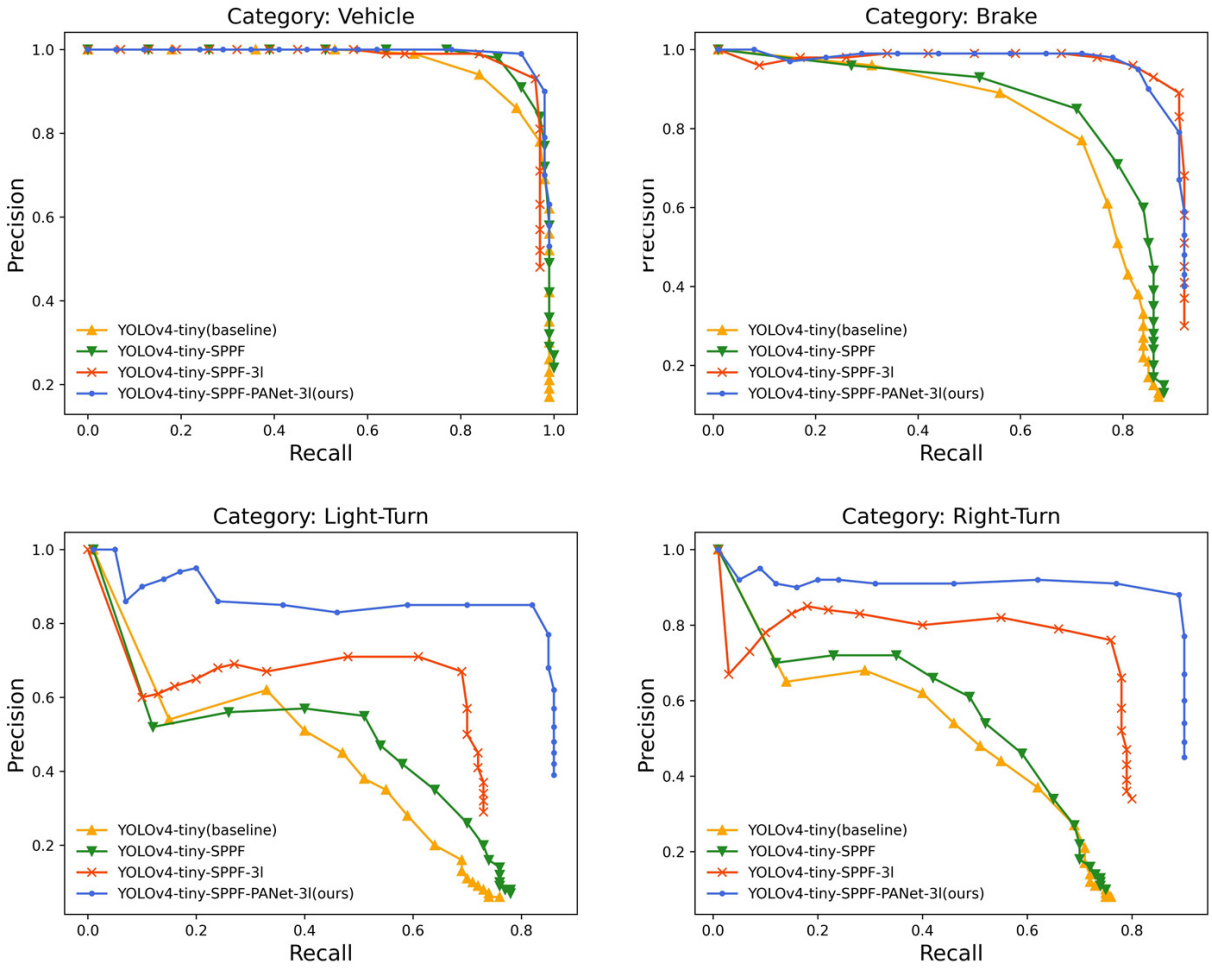

Figure 6. PR curves of four categories. YOLOv4-tiny [22] is chosen as the baseline.

5.4. Comparison of Detection Performance with Other Algorithms

This study evaluated three state-of-the-art object detectors, SSD, Faster R-CNN, and YOLOv4, on the test set.

We can conclude from Table 5 that the precision of these detection algorithms in vehicle detection is above 90% and YOLOv4 (92.13% mAP) has the most outstanding performance. The proposed method (86.83% mAP) possesses better performance than SSD (56.11% mAP) and Faster R-CNN (77.40% mAP). The main reason is that the SSD and Faster R-CNN detectors are not sensitive to small objects, especially turn signals. As the vehicle detection performance of these detection algorithms is good enough, the gap in the overall performance of the detection algorithm is specifically reflected in the precision of the detection of braking and turn signals.

Table 5. Comparison experiment results for vehicle taillight detection.

Method	mAP (%) @0.5	AP (%) @0.5			
		Vehicle	Brake	LT	RT
SSD [15]	56.11	91.69	73.40	28.51	30.87
Faster-RCNN [16]	77.40	98.13	89.92	59.37	62.20
YOLOv4 [21]	92.13	99.45	94.92	84.41	89.74
YOLOv4-tiny(baseline) [22]	62.20	96.48	74.76	35.59	41.95
YOLOv4-tiny-SPPF-3l-PANet (ours)	86.83	98.29	89.92	75.63	83.48

As can be seen from Table 6, SSD, the baseline, and our method are lightweight and have advantages in terms of detection rate, computation, and memory size. For SSD, this detector is the most lightweight, with a detection speed up to 80 FPS. This is mainly due to the fact that the backbone of the SSD, Mobilenetv2, is a lightweight classification network specifically designed for mobile platforms. The floating-point operations per second (FLOPTS) of Faster-RCNN is greater than 300 G. More importantly, Faster R-CNN cannot be used for real-time detection owing to its inference speed of 9 FPS. The memory size of YOLOv4 is as high as 244 MB, which is high for the limited in-vehicle platform. As a lightweight version of YOLOv4, the detection speed of YOLOv4-tiny is up to 144 FPS. However, the detection performance of YOLOv4-tiny is not satisfactory. Viewed this way, our method can balance detection precision, computation overhead, memory size, and inference time.

Table 6. Detection speed and model memory size.

Method	Input Size	Backbone	FLOPS	Memory Size (MB)	FPS
SSD [15]	300	Mobilenetv2	2.53 G	15.2	81
Faster-RCNN [16]	600	VGG16	369.96 G	108.0	9
YOLOv4 [21]	416	CSPDarknet	59.77 G	244.0.	35
YOLOv4-tiny(baseline) [22]	416	CSPDarknet-tiny	6.83 G	22.4	144
YOLOv4-tiny-SPPF-3l-PANet(ours)	416	CSPDarknet-tiny	9.39 G	47.0	73

5.5. Visualization of Detection Results

For a visual evaluation of the proposed method, Figures 7 and 8 show the detection performance of the proposed method compared to the baseline.

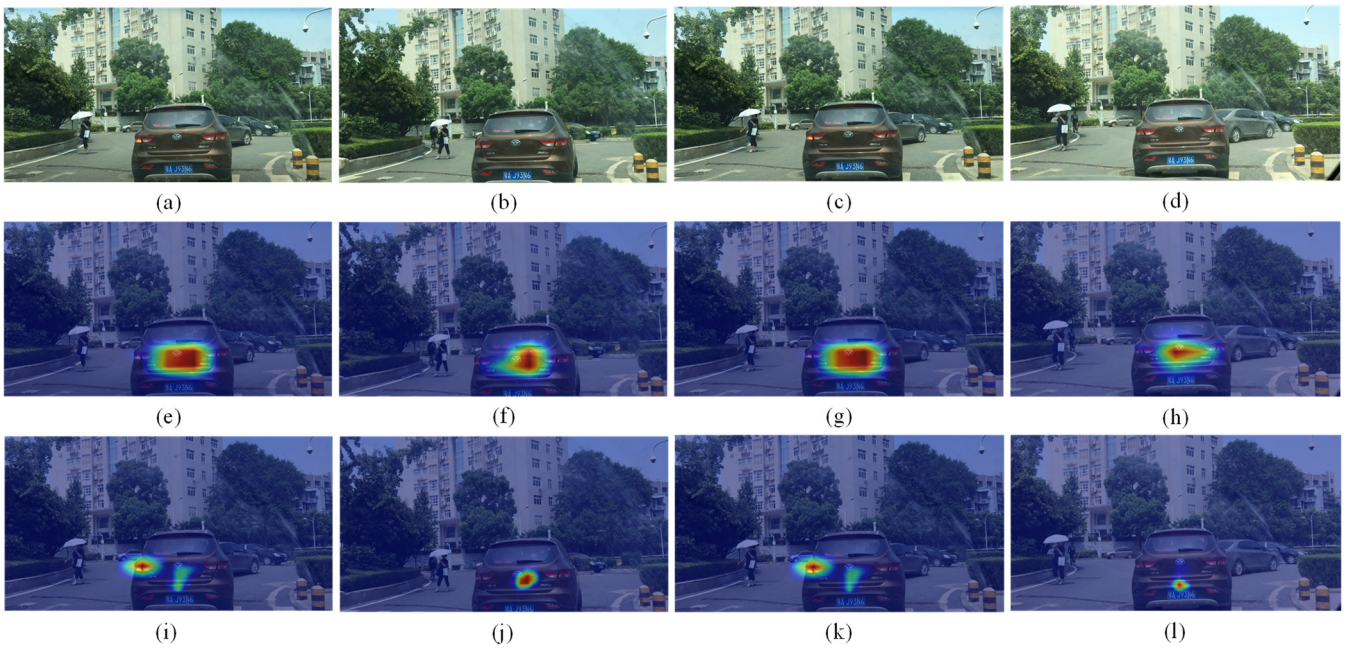

Figure 7. Visualization of interest regions for the baseline and the proposed method. (**a**–**d**) four consecutive pictures of left-turn signals from the preceding vehicle. (**e**–**h**) the interest regions of the baseline. (**i**–**l**) the interest regions of the proposed method.

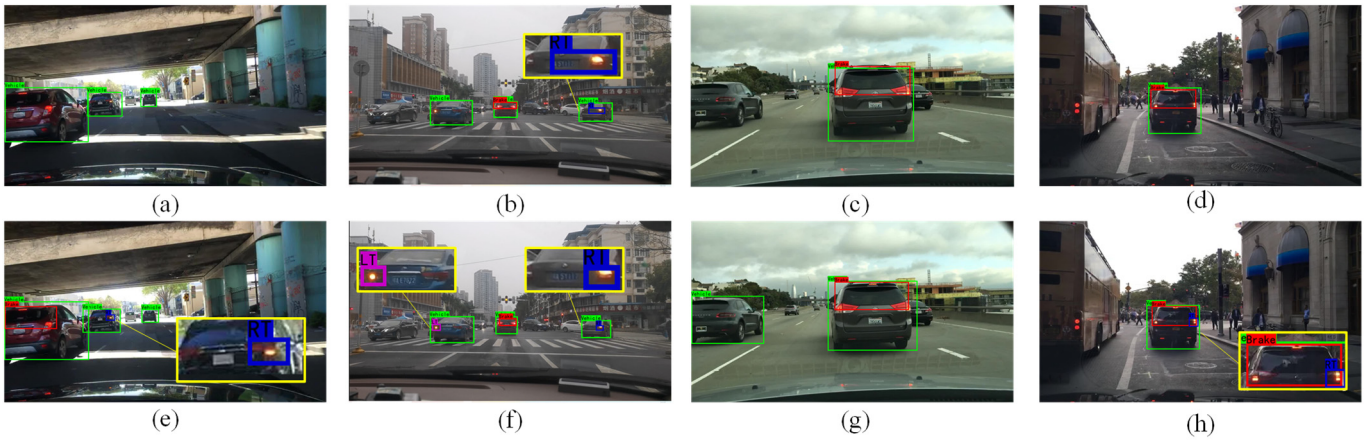

Figure 8. Detection results of the baseline and proposed method on the test set. (**a**–**d**) the detection results of the four test images in the baseline. (**e**–**h**) the detection results of the four test images in the proposed method.

Figure 7a–d are four consecutive turn signal pictures, with (a) and (c) having left turn signals and (b) and (d) having no turn signals. Figure 7e–l are the interest regions of the baseline and the proposed method, respectively. We found that the baseline can only focus on the vehicle object in the images. Our proposed method can recognize the turn signal while focusing on the vehicle object, indicating that the proposed method can detect turn signals better. Figure 8a–h shows the detection results of the baseline and the proposed method, respectively, on the test set.

Based on the discussion above, the proposed method showed better performance in detection ability than YOLOv4-tiny. This experimentally observed excellent performance should be ascribed to the feature-rich mechanism and multi-scale detection head strategy adopted in this article. In Figure 8, there are many turn signals and brake lights in the intersection driving traffic detection scene and the vehicle lane change scene. In these detection scenes, the vehicles are densely distributed, providing a good opportunity to test the object detection performance of the employed algorithms. Comparative analysis showed that our method greatly outperforms the baseline YOLOv4-tiny in the detection of vehicle taillights. Meanwhile, its ability to detect large objects (vehicles) is improved to a certain extent as well.

6. Conclusions

This study developed an improved real-time vehicle taillight intention detection approach. The proposed real-time detection method combines YOLOv4-tiny with a multi-scale detection strategy and integrates the SPPF and PANet modules. Multi-scale detection and multi-cross feature fusion are utilized to improve the detection precision of detection objects, especially for small objects. An expanded dataset consisting of selected images of BDD100K with collected images in real scenes was established. Based on the established real-scene dataset, experimental results showed that the overall detection precision reached 86.83%, a 24.63% improvement compared to the original YOLOv4-tiny. The proposed method improved the detection precision of vehicle, brake, left-turn, and right-turn lights by 1.81, 15.16, 40.04, and 41.53%, respectively, verifying the effectiveness of employing additional refined detection layers. Meanwhile, the ablation experimental results illustrate the advantages of the PANet over FPN in multi-cross feature fusion and small objects detection in vehicle taillight intention detection. Compared with other detectors, our method shows good real-time detection performance with relatively high precision, demonstrating its application potential in vehicle taillight intention detection.

Author Contributions: Conceptualization, B.T. and L.D.; methodology, B.T., W.C., L.D., C.L., Z.X. and D.Z.; software, B.T. and L.D.; validation, B.T., Z.X. and L.D.; formal analysis, W.C., C.L. and D.Z.;

data curation, B.T. and Z.X.; writing—original draft preparation, B.T. and L.D.; writing—review and editing, W.C. and L.D.; visualization, B.T. and L.D.; supervision, W.C. and L.D.; project administration, W.C. and L.D.; funding acquisition, W.C., C.L. and D.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded in part by the National Key R&D Program of China (No. 2018YFB0105205), in part by Hubei Province Technological Innovation Major Project (No. 2019AAA025), in part by the National Natural Science Foundation of China (No. 52102399), and in part by the Fundamental Research Funds for the Central Universities (No. 2022IVA039).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

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

References

1. Feng, D.; Haase-Schütz, C.; Rosenbaum, L.; Hertlein, H.; Gläser, C.; Timm, F.; Wiesbeck, W.; Dietmayer, K. Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 1341–1360. [CrossRef]
2. Cao, Z.; Guo, H.; Song, W.; Gao, K.; Chen, Z.; Zhang, L.; Zhang, X. Using reinforcement learning to minimize the probability of delay occurrence in transportation. *IEEE Trans. Veh. Technol.* **2020**, *69*, 2424–2436. [CrossRef]
3. Badue, C.; Guidolini, R.; Carneiro, R.V.; Azevedo, P.; Cardoso, V.B.; Forechi, A.; Jesus, L.; Berriel, R.; Paixão, T.M.; Mutz, F.; et al. Self-driving cars: A survey. *Expert Syst. Appl.* **2021**, *165*, 113816. [CrossRef]
4. Xu, G.; Bai, H.; Xing, J.; Luo, T.; Xiong, N.; Cheng, X.; Liu, S.; Zheng, X. SG-PBFT: A secure and highly efficient distributed blockchain PBFT consensus algorithm for intelligent Internet of vehicles. *J. Parallel Distrib. Comput.* **2022**, *164*, 1–11. [CrossRef]
5. Yang, P.; Xiong, N.; Ren, J. Data Security and Privacy Protection for Cloud Storage: A Survey. *IEEE Access* **2020**, *8*, 131723–131740. [CrossRef]
6. Du, L.; Chen, W.; Pei, Z.; Zheng, H.; Fu, S.; Chen, K.; Wu, D. Learning-Based Lane-Change Behaviour Detection for Intelligent and Connected Vehicles. *Comput. Intell. Neurosci.* **2020**, *2020*, 8848363. [CrossRef] [PubMed]
7. Du, L.; Chen, W.; Ji, J.; Pei, P.; Tong, B.; Zhen, H. A Novel Intelligent Approach to Lane-Change Behavior Prediction for Intelligent and Connected Vehicles. *Comput. Intell. Neurosci.* **2022**, *2022*, 9516218. [CrossRef] [PubMed]
8. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), San Diego, CA, USA, 20–25 June 2005; pp. 886–893. [CrossRef]
9. Lowe, D.G. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
10. Ojala, T.; Pietikainen, M.; Harwood, D. Performance evaluation of texture measures with classification based on Kullback discrimination of distributions. In Proceedings of the 12th International Conference on Pattern Recognition, Jerusalem, Israel, 9–13 October 1994; pp. 582–585. [CrossRef]
11. Ojala, T.; Pietikainen, M. A Comparative Study of Texture Measures with Classification Based on Feature Distributions. *Pattern Recognit.* **1996**, *29*, 51–59. [CrossRef]
12. Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C.; Xu, C. GhostNet: More Features from Cheap Operations. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 1577–1586. [CrossRef]
13. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [CrossRef]
14. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *42*, 386–397. [CrossRef] [PubMed]
15. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In Proceedings of the 2016 European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37. [CrossRef]
16. Ren, S.; He, K.; Girshick, R. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 1137–1149. [CrossRef] [PubMed]
17. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1440–1448. [CrossRef]
18. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788. [CrossRef]
19. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6517–6525. [CrossRef]

20. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
21. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
22. Darknet: Open Source Neural Networks in Python. Available online: https://github.com/AlexeyAB/darknet (accessed on 19 July 2022).
23. Yu, F.; Chen, H.; Wang, X.; Xian, W.; Chen, Y.; Liu, F.; Madhavan, V.; Darrell, T. BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 2633–2642. [CrossRef]
24. Geiger, A.; Lenz, P.; Urtasun, R. Are we ready for autonomous driving? The KITTI vision benchmark suite. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 3354–3361. [CrossRef]
25. Behrend, K.; Novak, L.; Botros, R. A deep learning approach to traffic lights: Detection, tracking, and classification. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 1370–1377. [CrossRef]
26. Cui, Z.; Yang, S.W.; Tsai, H.M. A Vision-Based Hierarchical Framework for Autonomous Front-Vehicle Taillights Detection and Signal Recognition. In Proceedings of the 2015 IEEE 18th International Conference on Intelligent Transportation Systems, Gran Canaria, Spain, 15–18 September 2015; pp. 931–937. [CrossRef]
27. Chen, H.T.; Wu, Y.C.; Hsu, C.C. Daytime Preceding Vehicle Brake Light Detection Using Monocular Vision. *IEEE Sens. J.* **2015**, *16*, 120–131. [CrossRef]
28. Nava, D.; Panzani, G.; Savaresi, S.M. A Collision Warning Oriented Brake Lights Detection and Classification Algorithm Based on a Mono Camera Sensor. In Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, 27–30 October 2019; pp. 319–324. [CrossRef]
29. Zhong, G.; Tsai, Y.; Chen, Y.; Mei, X.; Prokhorov, D.; James, M.; Yang, M. Learning to tell brake lights with convolutional features. In Proceedings of the 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), Rio de Janeiro, Brazil, 1–4 November 2016; pp. 1558–1563. [CrossRef]
30. Vancea, F.I.; Nedevschi, S. Semantic Information Based Vehicle Relative Orientation and Taillight Detection. In Proceedings of the 2018 IEEE 14th International Conference on Intelligent Computer Communication and Processing (ICCP), Cluj-Napoca, Romania, 6–8 September 2018; pp. 259–264. [CrossRef]
31. Hsu, H.K.; Tsai, Y.H.; Mei, X.; Lee, K.H.; Nagasaka, N.; Prokhorov, D.; Yang, M.H. Learning to tell brake and turn signals in videos using CNN-LSTM structure. In Proceedings of the 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), Yokohama, Japan, 16–19 October 2017; pp. 1–6. [CrossRef]
32. Frossard, D.; Kee, E.; Urtasun, R. DeepSignals: Predicting Intent of Drivers Through Visual Signals. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 9697–9703. [CrossRef]
33. Lee, K.H.; Tagawa, T.; Pan, J.M.; Gaidon, A.; Douillard, B. An Attention-based Recurrent Convolutional Network for Vehicle Taillight Recognition. In Proceedings of the 2019 IEEE Intelligent Vehicles Symposium (IV), Paris, France, 9–12 June 2019; pp. 2365–2370. [CrossRef]
34. Tiago, S.N.; Gabriel, B.P.; Welinton, A.C.; Moacir, P. Deep Convolutional Neural Networks and Noisy Images. In Proceedings of the 22th Iberoamerican Congress on Pattern Recognition (CIARP), Valparaiso, Chile, 7–10 November 2017; pp. 416–424. [CrossRef]
35. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904–1916. [CrossRef] [PubMed]
36. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 936–944. [CrossRef]
37. Cai, Y.; Luan, T.; Gao, H.; Wang, H.; Chen, L.; Li, Y.; Sotelo, M.A.; Li, Z. YOLOv4-5D: An Effective and Efficient Object Detector for Autonomous Driving. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 1–13. [CrossRef]
38. Deng, C.; Wang, M.; Liu, L.; Liu, Y.; Jiang, Y. Extended Feature Pyramid Network for Small Object Detection. *IEEE Trans. Multimed.* **2022**, *24*, 1968–1979. [CrossRef]
39. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path Aggregation Network for Instance Segmentation. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768. [CrossRef]
40. Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression. In Proceedings of the 34th AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; pp. 12993–13000. [CrossRef]
41. Bodla, N.; Singh, B.; Chellappa, R.; Davis, L.S. Soft-NMS—Improving Object Detection with One Line of Code. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 5562–5570. [CrossRef]
42. Kingma, D.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2015**, arXiv:1412.6980.