sustainability

MDPI

Article

Traffic Sign Detection Based on Lightweight Multiscale Feature Fusion Network

Shan Lin [1], Zicheng Zhang [1], Jie Tao [2,*], Fan Zhang [3], Xing Fan [1] and Qingchang Lu [1]

[1] School of Electronic and Control Engineering, Chang'an University, Xi'an 710064, China
[2] Zhejiang Institute of Mechanical and Electrical Engineering Co., Ltd., Hangzhou 310002, China
[3] School of Information Engineering, Chang'an University, Xi'an 710064, China
* Correspondence: jietao211@126.com

Abstract: Traffic sign detection is a research hotspot in advanced assisted driving systems, given the complex background, light transformation, and scale changes of traffic sign targets, as well as the problems of slow result acquisition and low accuracy of existing detection methods. To solve the above problems, this paper proposes a traffic sign detection method based on a lightweight multiscale feature fusion network. Since a lightweight network model is simple and has fewer parameters, it can greatly improve the detection speed of a target. To learn more target features and improve the generalization ability of the model, a multiscale feature fusion method can be used to improve recognition accuracy during training. Firstly, MobileNetV3 was selected as the backbone network, a new spatial attention mechanism was introduced, and a spatial attention branch and a channel attention branch were constructed to obtain a mixed attention weight map. Secondly, a feature-interleaving module was constructed to convert the single-scale feature map of the specified layer into a multiscale feature fusion map to realize the combined encoding of high-level semantic information and low-level semantic information. Then, a feature extraction base network for lightweight multiscale feature fusion with an attention mechanism based on the above steps was constructed. Finally, a key-point detection network was constructed to output the location information, bias information, and category probability of the center points of traffic signs to achieve the detection and recognition of traffic signs. The model was trained, validated, and tested using TT100K datasets, and the detection accuracy of 36 common categories of traffic signs reached more than 85%, among which the detection accuracy of five categories exceeded 95%. The results showed that, compared with the traditional methods of Faster R-CNN, CornerNet, and CenterNet, traffic sign detection based on a lightweight multiscale feature fusion network had obvious advantages in the speed and accuracy of recognition, significantly improved the detection performance for small targets, and achieved a better real-time performance.

Keywords: traffic engineering; traffic sign detection; convolutional neural network; multiscale feature fusion; attention mechanism

Citation: Lin, S.; Zhang, Z.; Tao, J.; Zhang, F.; Fan, X.; Lu, Q. Traffic Sign Detection Based on Lightweight Multiscale Feature Fusion Network. *Sustainability* **2022**, *14*, 14019. https://doi.org/10.3390/su142114019

Academic Editor: Lihui Zhang

Received: 30 August 2022
Accepted: 26 October 2022
Published: 27 October 2022

1. Introduction

Given the various problems existing in the process of road transportation [1], organizations in various countries have invested manpower, material resources, and financial resources to find solutions. Advanced driving assistance systems (ADASs) have become one of the key technologies to solve the problem. An ADAS can collect information about the movement of a vehicle and the surrounding environment and understand the information to help drivers make decisions; however, as an important part of an ADAS, traffic sign identification and recognition can timely convey guidance, restriction, or warning or can prompt information expressed by traffic signs within a certain range to drivers, which can ensure the safety of road driving to a certain extent.

In early studies, the feature-learning method has mainly been used, which mainly utilizes the inherent features of traffic signs and detects and identifies traffic signs based on

machine learning. In references [2–5], traffic signs are detected by feature-learning methods. However, these methods overuse the shapes and color features of traffic signs. Cases of clear color and standard shape have better detection and recognition effects. However, in complex environments, such as light transformation and shape distortion, this type of method has the problem of low detection and recognition rates. Yu et al. [6] designed a framework for traffic sign detection in complex scenes based on visual co-visuality that used three visual attention cues, contrast, center deviation, and symmetry, to detect traffic signs. The advantages of this method were the integration of bottom-up and top-down visual processing and no heavy-learning tasks. Yu et al. [7] proposed a traffic sign detection method based on saliency maps and Fourier descriptors. This method used frequency tuning to obtain a saliency map and used binary operation to obtain a binary image and capture traffic sign area. Based on a visual attention mechanism model, Zhang et al. [8] extracted edge and color information as early visual features, calculated each feature to obtain a visual saliency map, and then used a graph neural network and K-means algorithm to determine candidate regions containing traffic signs. The visual saliency method mainly relies on modeling of the visual attention mechanism. In complex environments, this kind of method has a certain improvement in the detection and recognition rates of traffic signs compared with the feature-learning method. However, this kind of method is relatively deficient in the extraction of deep features of traffic signs, such as the feature extraction of traffic signs of the same category in different weather environments.

Deep convolutional neural networks are widely used in traffic sign detection and recognition tasks and have achieved good results. Yin et al. [9] designed a novel structure combining intranetwork connections and residual connections and used an efficient GPU to accelerate convolution operations. Zhe et al. [10] created the TT100K traffic sign database based on Tencent Street View and labeled each traffic sign appearing in the images. Xie et al. [11] proposed a two-level cascaded convolutional neural network structure that could effectively improve the classification accuracy of traffic signs. Zhu et al. [12] proposed a novel framework for traffic sign detection and recognition that contained a fully convolutional network to guide traffic sign proposals and a deep convolutional neural network for target classification. Zhang [13] and Zuo [14] each applied Faster R-CNN to traffic sign detection and recognition and improved the efficiency of feature extraction through an RPN network and end-to-end model training. There are many kinds of traffic signs, including not only graphic signs but also text signs. To detect graphic signs and text signs at the same time, Luo et al. [15] first classified the region of interest into two categories to distinguish graphic and text signs and then identified specific categories through a deep feature extraction network. Zhu et al. [16] proposed a text-based traffic sign detection network that solved the multiscale problem of text detection by narrowing the text detection area. In practical applications, small traffic signs are prone to low resolution when imaging and different degrees of blur or deformation, so the images contain less information and the features are not easy to extract. In response to this problem, Peng et al. [17] introduced local regions into a detection network structure to provide more local information for the back-end recognition network and improve the detection and recognition rates of small traffic signs. Pei et al. [18] proposed a multiscale deconvolution network that flexibly used a multiscale CNN to form efficient and reliable local traffic sign recognition model training. Li [19] and Heng [20] et al. both used adversarial generative networks to enhance the detection and recognition capabilities of the original network. Xiang et al. [21] proposed an improved capsule network for traffic sign recognition in a multiscale capsule network. Traffic sign images are susceptible to the effects of shooting angle and distance, which change the size of the sign and seriously affect the feature extraction and classification processes. To address this problem, Xie et al. [22] proposed a traffic sign recognition method based on an LeNet-5 network by adjusting target size using bilinear interpolation and then using the adjusted image for feature extraction. In complex traffic scenarios, some objects are similar to traffic signs in appearance, and a detector may detect them incorrectly. To address this problem, Yuan et al. [23] found that

the locations of traffic signs had obvious statistical features when they were counted, so the location a priori information was introduced into a network to improve the accuracy of traffic sign detection. Lee et al. [24] proposed a novel traffic sign detection system that used CNN to simultaneously estimate the locations and precise boundaries of traffic signs. Kong et al. [25] combed the latest research results and proposed future development trends in machine learning and deep learning in traffic target detection. Zhou et al. [26] proposed a standard dataset and presented a high-resolution traffic sign recognition algorithm for complex environments. There are many small objects in traffic scenes, but their detection is still a challenge due to low resolution and limited information. For this reason, Lian et al. [27] proposed a small object detection method for traffic scenes based on attentional feature fusion to improve the small target detection accuracy.

To better apply convolutional neural networks in the field of traffic sign detection and recognition, various research scholars have proposed many convolutional neural networks with better performances but higher network complexity based on AlexNet [28], such as VGG [29], ResNet [30], DenseNet [31], etc. The transportation problem is complex, as well as nonlinear, and there is a lot of data information to be processed. The use of traditional deep neural networks leads to great difficulties in storing models in embedded devices due to the large limitation of storage space for hardware devices. In response to the high complexity of deep neural networks, which is difficult to be applied in real life, some scholars have tried to study lightweight networks to reduce complexity while maintaining the network effect and have made breakthrough progress. The research on lightweight networks is mainly divided into two directions, model compression and network structure design, in which the representative methods of model compression are knowledge distillation, pruning, quantization, and low-rank decomposition, and the representative networks of network structure design are SqueezeNet [32], MobileNet [33,34], ShuffleNet [35], and Xception [36].

In summary, a traffic sign detection and recognition algorithm based on deep learning can handle more complex application scenarios, but there are still some difficulties in the process of practical application. A traffic sign detection algorithm requires high real-time performance. Additionally, a traffic sign generally occupies a small proportion of the image, and there is a certain scale change. Based on a situation analysis of traffic sign detection and recognition, this paper researches and proposes a traffic sign detection and recognition algorithm based on lightweight multiscale feature fusion and an attention mechanism for intelligent driving applications. Firstly, the convolutional neural network C-MobileNetV3, which is based on the improved MobileNetV3 [37], is adopted as the feature extraction network, and the input images are convolved with different numbers of convolutional kernels to generate single-scale images of different depths. Then, feature maps of different scales are input into the feature-interleaving module, and the top-down connection, bottom-up connection, and layer-by-layer cascade modules are used to obtain multiscale feature fusion maps that fuse different scale information. Finally, the obtained multiscale feature fusion map is input into the detection network for foreground and background discrimination to complete the detection and recognition of traffic signs.

2. Methodology and Models

2.1. Feature Extraction

2.1.1. Lightweight Feature Extraction Network

To efficiently complete the extraction of feature maps, in this paper we selected a modified lightweight network based on MobileNetV3 as the base network for feature extraction and named it C-MobileNetV3. The basic structure of the network was a bottleneck [38]. The core idea was to reduce the computational complexity by using deep, separable convolution instead of standard convolution. In addition, a lightweight attention module was added to enhance the network learning ability. The bottleneck design consisted of two structures with a step size of 1 and a step size of 2, respectively. Among them, the structure with the step size of 1 was added with the residual connection. In addition, some bottlenecks were added with a squeeze-and-excitation (SE [32]) module, as shown in Figure 1.

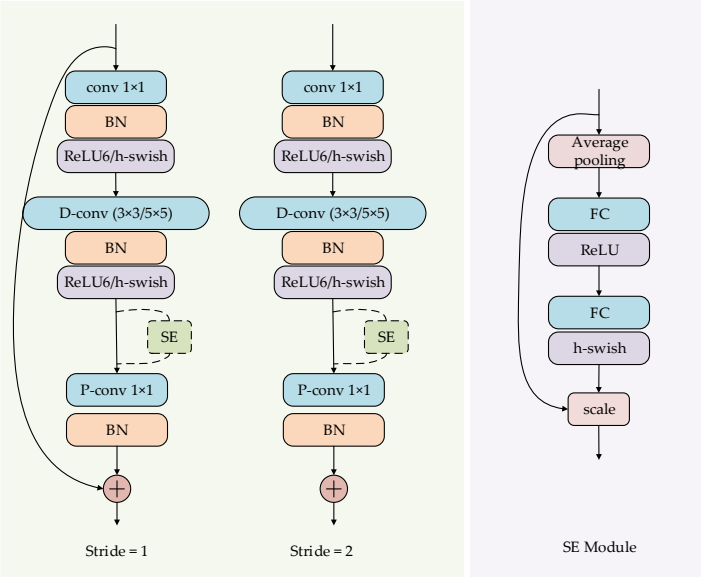

Figure 1. Bottleneck structure diagram.

The input image was first mapped from the low-dimensional space to the high-dimensional space by expanding the convolution of 1×1 to obtain enough information that the number of expansion channels could be set as desired. Next, channel-by-channel convolution was used to output the information of each channel, and point-by-point convolution was used to complete the joint coding between channels to compress the number of channels, which in turn made the network lighter. ReLU6 was used as the activation function for the first half of the model, and Hard-Swish was used for the second half. The overall framework of the lightweight feature extraction base network is shown in Figure 2. The input image was first convolved in 16 dimensions with a step size of 2 and a convolution kernel size of 3×3 to extract preliminary information and generate a $1024 \times 1024 \times 16$ feature map. Then, it was input into the feature extraction network and stacked by different bottleneck specifications.

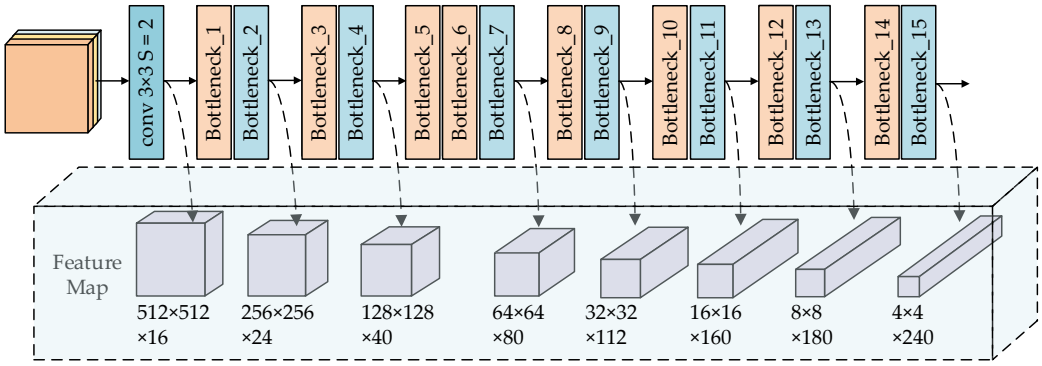

Figure 2. C-MobileNet overall framework.

2.1.2. Multiscale Feature Fusion Network

In this paper, we proposed a feature-interleaving module by borrowing the idea of BiFPN [39] for feature fusion. As shown in Figure 3, the network was generally composed of three parts: a top-down module, a bottom-up module, and a layer-by-layer connection module.

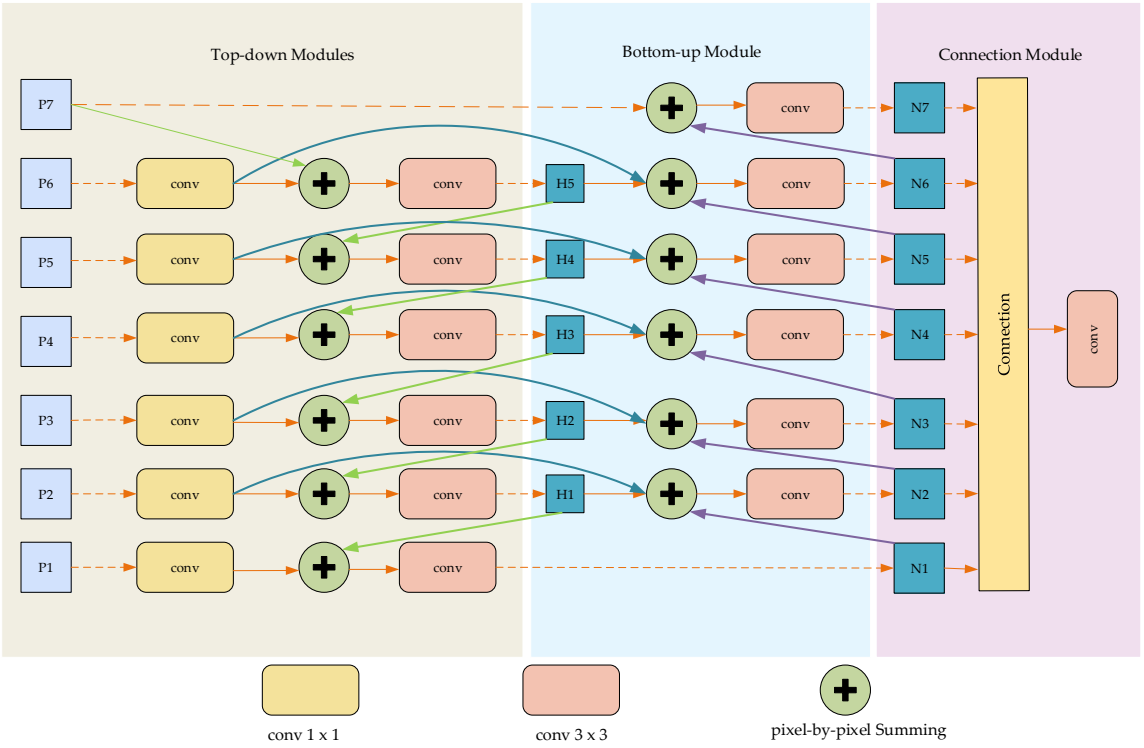

Figure 3. Feature-interleaving module.

1. Top-down module. To obtain a rich feature representation, the multiscale fusion module designed in this paper selected seven different sizes of feature maps for fusion, denoted as $\overrightarrow{P} = (P_1, P_2, \ldots, P_7)$. These feature maps were the feature maps output from Bottleneck_2, Bottleneck_4, Bottleneck_7, Bottleneck_9, Bottleneck_11, Bottleneck_13, and Bottleneck_15 in the lightweight feature extraction network, with sizes of $512 \times 512 \times 24$, $256 \times 256 \times 40$, $128 \times 128 \times 80$, $64 \times 64 \times 112$, $32 \times 32 \times 160$, $16 \times 16 \times 200$, and $8 \times 8 \times 240$, respectively.

In this paper, we used the summation operation for feature fusion and achieved consistency in the number of channels through the convolution kernel of $1 \times 1 \times 240$. The green arrow in the figure represents the bilinear interpolation. The size of the feature map after the bilinear interpolation became twice the previous one, and the transformed feature map could be fused with the underlying feature map. The convolution kernel of 3×3 was used to re-extract the features to ensure the stability of the features.

2. Bottom-up Module. To improve the limitations of unidirectional information flow, a bottom-up module was designed in this paper. The bottom-up module mainly consisted of three parts: max pooling, lateral connection, and cross-stage connection. The purple arrows in the figure represent max pooling, and show the cross-level connections. By cross-level connection, more features could be fused without additional cost. The feature map after max pooling, the feature map after cross-level connection, and the feature map obtained in the top-down module were added pixel by pixel, and the obtained results were used for feature extraction through convolution to obtain the output of the top-down module.
3. Layer-by-Layer Connection Module. To improve the efficiency of network detection, the connection module fused the feature map $\overrightarrow{N}(N_1, N_2, \ldots N_7)$ generated by the bottom-up module by summation and then performed feature re-extraction by convolution with a size of 3×3, a step size of 1, and a number of output channels to generate the fused feature map of $256 \times 256 \times 240$.

2.2. Hybrid Attention Module

As can be seen from the structure in Figure 4, the input feature map of the attention module [40] generated the channel domain attention feature map after the channel attention branch. After spatial-domain-attention-branching, the spatial attention feature map was generated, and then the two were added pixel by pixel to generate the hybrid attention feature map.

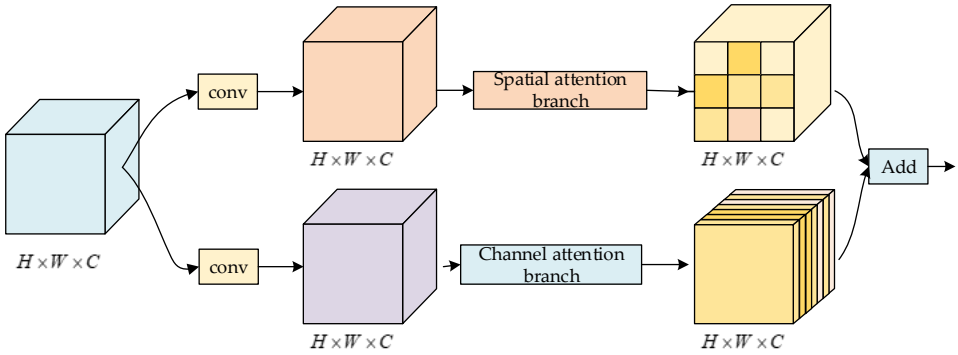

Figure 4. Hybrid attention module.

2.2.1. Spatial Attention Module

The structure diagram of the spatial attention module is shown in Figure 5, which omits the batch normalization (BN) layer and the sigmoid layer.

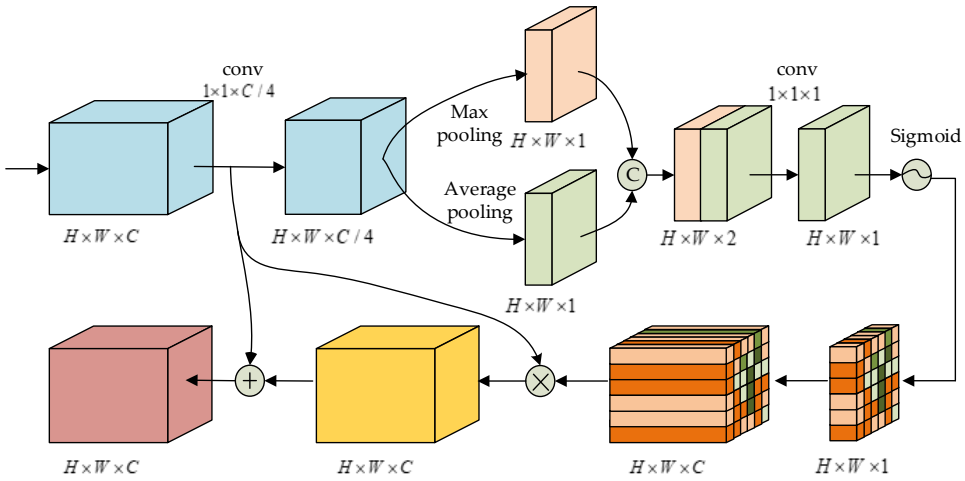

Figure 5. Spatial attention module.

The main steps of the spatial-attention-branching subnetwork calculation method were as follows:

1. Input the feature map $F \in R^{H \times W \times C}$, where the superscript indicates the height and width of the feature map and the number of channels.
2. Convolution of the input feature map of the size 3×3 to generate spatial feature map $F_S \in R^{H \times W \times C}$:

$$F_S = \mathrm{BN}(f^{(3 \times 3 \times C)}(F)) \quad (1)$$

where BN denotes the batch normalization operation and the convolution operation, and the superscript f denotes the convolution kernel size and the number of output channels.

3. Compression of the channel direction to generate the feature map $F_S{}' \in R^{H\times W\times C/4}$; its generation process is as follows:

$$F_S' = \mathrm{BN}(f^{(1\times 1\times C/4)}(F_S)) \tag{2}$$

4. Generation of spatial attention weight maps by pooling, convolution, and deformation operations $W_S \in R^{H\times W\times C}$:

$$W_s = \mathrm{ReaShape}(\mathrm{Sigmoid}(\mathrm{BN}(f^{(1\times 1\times 1)}(\mathrm{Concat}(\mathrm{MaxPool}_C(F_S') + \mathrm{AvgPool}_C(F_S')))))) \tag{3}$$

where $\mathrm{MaxPool}_C$ is the pixel maximum in the channel direction of the feature map; $\mathrm{AvgPool}_C$ is the pixel average in the channel direction of the feature map; Concat is the stitching of the feature map in the channel direction; Sigmoid is the activation function; and ReaShape is the deformation.

5. Acquisition of the spatial attention map by multiplying the spatial attention weight map $W_S \in R^{H\times W\times C}$ and the spatial feature map F_S; then, the spatial attention map is added to F_S to obtain the final output of the spatial attention subnet $F_{SW}^{H\times W\times C}$:

$$F_{SW} = F_S + (W_S \odot F_S) \tag{4}$$

2.2.2. Channel Attention Module

The channel domain attention mechanism used a network-learning method to obtain feature map channel weights, which complemented the spatial domain attention branch to achieve a reasonable allocation of computer resources. The structure of the channel-domain attention branch is shown in Figure 6.

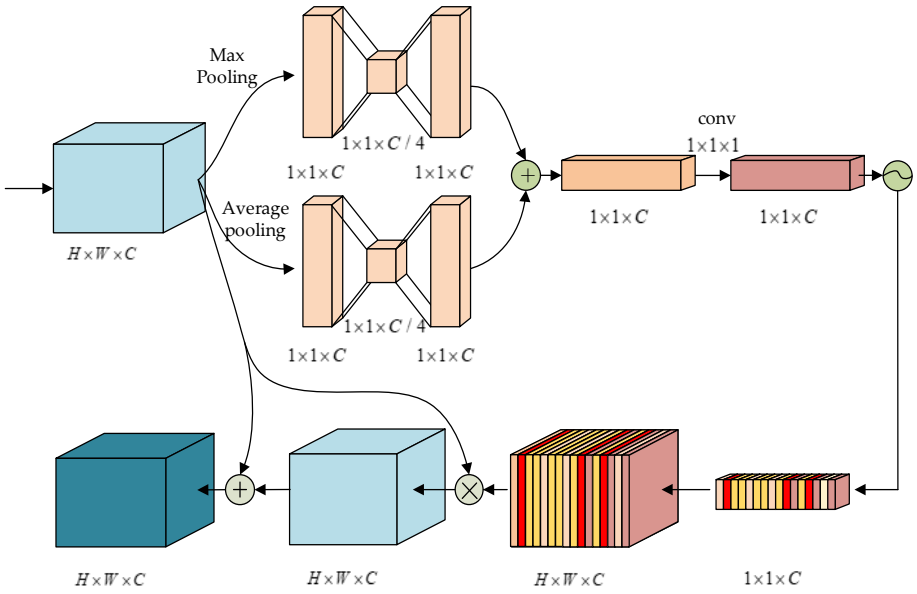

Figure 6. Channel attention module.

The main steps of the channel-domain-attention-branching subnetwork calculation method were:

1. Enter the feature map $F \in R^{H\times W\times C}$, where the superscript indicates the height and width of the feature map and the number of channels.
2. After the convolution operation of 3×3 is performed on the input feature map F, generate the channel domain basic feature map F_C:

$$F_C = \mathrm{BN}(f^{(3\times 3\times C)}(F)) \tag{5}$$

where BN denotes the batch normalization operation; f denotes the convolution operation; and the superscript denotes the convolution kernel size, as well as the number of output channels.

3. Calculate the global maximum pooling and global average pooling:

$$F_{CM} = \text{MaxPool}(F_C) \tag{6}$$

$$F_{CA} = \text{AvgPool}(F_C) \tag{7}$$

where MaxPool represents the pixel maximum value among the $H \times W$ data of each channel of the feature map, and AvgPool represents the pixel average value among the $H \times W$ data of each channel of the feature map.

4. Compress the channels through compressed convolution, and then the feature maps are updimensioned using C neurons to obtain the outputs of the global maximum pooling branch $F'_{CM} \in R^{1\times1\times C}$ and the global average pooling branch $F'_{CA} \in R^{1\times1\times C}$, respectively:

$$F'_{CM} = \text{BN}(f^{(1\times1\times C)}(\text{BN}(f^{(1\times1\times\frac{C}{4})}(F_{CM})))) \tag{8}$$

$$F'_{CA} = \text{BN}(f^{(1\times1\times C)}(\text{BN}(f^{(1\times1\times\frac{C}{4})}(F_{CA})))) \tag{9}$$

5. Obtain the fused feature maps of the global maximum pooling branch and the global average pooling branch by adding F'_{CM} and F'_{CA} pixel by pixel to obtain the channel domain attention weight map:

$$W_C = \text{ReaShape}(\text{Sigmoid}(\text{BN}(f^{(1\times1\times1)}(F'_{CM} + F'_{CA})))) \tag{10}$$

where $+$ denotes pixel-by-pixel summation, Sigmoid denotes the activation function, and ReaShape denotes upsampling.

6. Obtain the channel domain attention map by multiplying the channel domain attention weight map $W_C \in R^{H\times W\times C}$ with the channel domain basis feature map F_C; then, add the channel domain attention map with F_S to obtain the final output of the channel domain attention subnetwork $F_{CW}^{H\times W\times C}$:

$$F_{CW} = F_C + (W_C \odot F_C) \tag{11}$$

where $\odot$ indicates multiplying pixel by pixel, and $+$ indicates adding pixel by pixel.

2.3. Detection Network

Based on the idea of CenterNet [41], this paper used a target detection method without an anchor frame to detect traffic signs. This paper used the center point of the traffic sign truth box in the image to represent the current target, that is, a traffic sign was detected as a point. The actual prediction box of the object was obtained by predicting the center point offset and width of the target, and a heat map represented the classification information. The detection network framework is shown in Figure 7. The output of the multiscale fusion feature extraction network was used as the input of the detection network, and the input was input into three branches that predicted the heat map, width and height, and position offset of the traffic sign center point. Then, the obtained results were mapped to the original image size by suppressing the center point with a higher retention score through the maximum value, and finally, the target position and category were obtained.

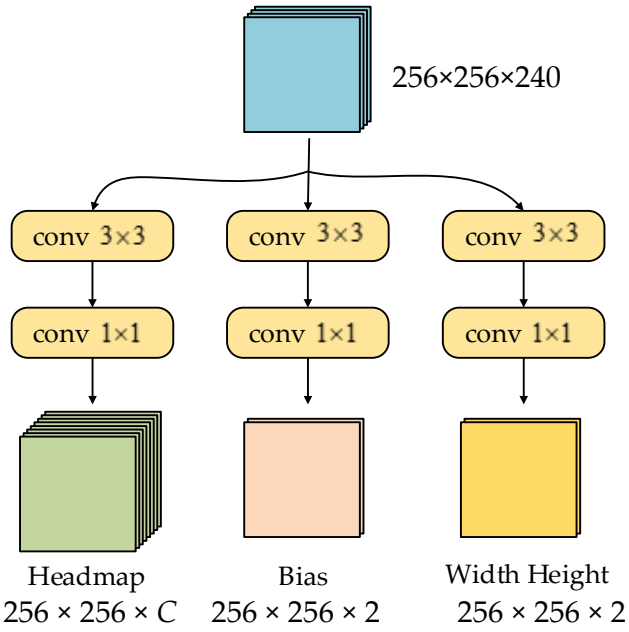

Figure 7. Detection network structure diagram.

The loss function for the heat map portion was denoted as follows:

$$L_K = \frac{1}{N}\sum_{xyz}\begin{cases}(1-\hat{Y}_{xyz})^{\alpha}\log(\hat{Y}_{xyz}), Y_{xyz}=1\\(1-Y_{xyz})^{\beta}(\hat{Y}_{xyz})^{\alpha}\log(1-\hat{Y}_{xyz}),\ \text{otherwise}\end{cases} \tag{12}$$

where $\hat{Y}_{xyz}$ represents the predicted value of the heat map at the channel c position (x, y); Y_{xyz} represents the true value of the corresponding position, which was calculated based on the Gaussian distribution of the centroid of the true value; N represents the number of traffic signs in the picture; the α parameters are used to control the loss weights of the hard and easy classification samples; and the β parameters are used for weight. In the experiments, α was taken as 2, and β was taken as 4.

The bias loss function was denoted as follows:

$$L_{off} = \frac{1}{N}\sum_{p}\left|\hat{O}_p - O_p\right| \tag{13}$$

where O_p is the actual bias, and $\hat{O}_p$ represents the bias predicted by the network.

The length and width predicted loss value was calculated as follows:

$$L_{size} = \frac{1}{N}\sum_{k=1}^{N}\left|\hat{S}_{pk} - S_k\right| \tag{14}$$

where $\hat{S}_{pk}$ refers to the predicted width and height, and S_k refers to the actual width and height.

The total loss function of the network was as follows:

$$L_{\text{det}} = L_k + \lambda_{size}L_{size} + \lambda_{off}L_{off} \tag{15}$$

where $\lambda_{size} = 0.1$ and $\lambda_{off} = 1$ represent the weights of different partial loss functions.

3. Experiments and Results

3.1. Experiment Platform

The dataset used in this chapter was TT100K, and the evaluation metrics used were precision, recall, PR curve, index, and frames per second (FPS). The experiments were conducted with an Intel(R) Xeon(R) E5-2680v3 processor, with a 2.5 GHz GPU and an NVIDIA Geforce GTX 1080ti graphics card using Python3.6 as the programming language and Pycharm as the software, based on Pytorch 1.1.0 deep learning platform, CUDA 9.0 and CUDnn7.1 was used to implement the algorithms in this paper.

3.2. Ablation Study

The models for conducting the ablation experiments were classified as follows.

Model 1: The network contained C-MobileNet with a detection network.

Model 2: The network contained C-MobileNet, a spatial domain attention module, and a detection module.

Model 3: The network contained C-MobileNet, a channel domain attention module, and a detection module.

Model 4: The network contained C-MobileNet, a hybrid attention module, and a detection module.

Model 5: The network contained C-MobileNet, a feature-interleaving module, and a detection module.

Model 6: The network contained C-MobileNet, a feature-interleaving module, a spatial domain attention module, and a detection module.

Model 7: The network contained C-MobileNet, a feature-interleaving module, a channel domain attention module, and a detection module.

Model 8: The network contained C-MobileNet, a feature-interleaving module, a hybrid attention module, and a detection module.

The experimental results are shown in Table 1.

Table 1. Ablation study.

	Average Accuracy (%)			**Recall (%)**			***F*1 (%)**		
	Small	**Mid**	**Large**	**Small**	**Mid**	**Large**	**Small**	**Mid**	**Large**
Model 1	82.7	91.1	89.8	78.6	87.9	87.5	80.6	89.5	88.6
Model 2	83.3	91.7	90.4	80.8	88.6	88.2	82.0	90.1	89.2
Model 3	83.4	91.8	90.6	81.5	89.8	89.0	82.4	90.7	89.8
Model 4	84.7	93.1	91.9	82.6	91.3	89.8	83.6	92.2	90.8
Model 5	84.1	91.8	91.3	81.6	90.8	88.9	82.8	91.3	90.1
Model 6	84.6	92.7	91.4	84.5	92.8	91.7	84.6	92.7	91.5
Model 7	85.0	92.8	91.9	85.2	93.8	93.1	85.1	93.3	92.5
Model 8	86.7	93.7	92.8	89.8	94.9	94.7	88.2	94.3	93.7

An analysis of the horizontal information in the table shows that the recognition effects of medium-sized traffic signs and large-sized traffic signs were better than those of small-sized traffic signs. This was due to the clear edges and obvious features of medium-sized and large-sized traffic signs, while the small-sized traffic signs had fuzzier features, which led to poorer recognition results. Through a longitudinal comparison of the information in the table, it can be seen that, for small-sized traffic signs, supplementation with the attention module and the multiscale feature fusion module could effectively improve the recognition effect.

Specifically, comparing Model 1, Model 2, Model 3, and Model 4 showed that the channel domain attention module and the spatial domain attention module could improve the detection effect to a certain extent, while the hybrid attention module, as well as the hybrid attention module combining the two, had more significant improvement effects on the detection task of traffic signs. Comparing Model 1 with Model 5, it can be seen that the introduction of the feature-interleaving module in the model improved the recognition

effect from an overall perspective, especially for small sizes, where the improvement effect could reach 1.4%. Model 6, Model 7, and Model 8 added the attention module based on Model 5. Comparing Model 5, Model 6, and Model 7, it can be seen that adding the channel domain attention module and spatial domain attention module to the network improved the detection effect of traffic signs in a certain range. Comparing Model 6, Model 7, and Model 8 shows that using both the spatial domain attention module and the channel domain attention module further improved the detection effect of the models.

Figure 8 shows the detection plots of different models for traffic signs.

Figure 8. Ablation study results.

3.3. Structural Experiments

To verify the effectiveness of each module in the proposed network, we designed the following experiments to demonstrate the effectiveness of the proposed method.

The models for conducting structural experiments were classified as follows:

Model 1: MobileNetV1 was chosen as the base network for lightweight feature extraction, and other parts of the network remained unchanged.

Model 2: ShuffleNet was selected as the base network for lightweight feature extraction, and other parts of the network remained unchanged.

Model 3: MobileNetV2 was selected as the base network for lightweight feature extraction, and other parts of the network remained unchanged.

Model 4: BiFPN was selected for the multiscale fusion module, and other parts of the network remained unchanged.

Model 5: NAS-FPN [42] was selected for the multiscale fusion module, and other parts of the network remained unchanged.

Model 6: PANet [43] was selected for the multiscale fusion module, and other parts of the network remained unchanged.

Model 7: CBAM [44] was selected as the attention module, and other parts of the network remained unchanged.

Model 8: DANet [45] was selected as the attention module, and other parts of the network remained unchanged.

Model 9: The network proposed in this paper was used.

Experiments were conducted on the above models, and the data results are shown in Table 2.

Table 2. Structural experiments.

	Average Accuracy (%)			Recall (%)			*F*1 (%)		
	Small	**Mid**	**Large**	**Small**	**Mid**	**Large**	**Small**	**Mid**	**Large**
Model 1	84.5	92.6	92.1	82.7	91.8	88.9	83.5	92.2	90.5
Model 2	84.9	92.9	92.5	83.6	92.2	89.4	84.2	92.5	91.0
Model 3	85.6	93.2	91.8	87.7	93.5	92.1	86.6	93.3	91.9
Model 4	84.9	92.0	91.6	83.9	89.3	88.5	84.4	90.6	90.0
Model 5	85.2	92.3	91.8	84.2	91.7	91.2	84.7	82.0	91.5
Model 6	85.7	92.8	92.3	86.2	93.9	94.6	85.9	93.3	93.0
Model 7	85.2	93.2	92.5	89.3	94.6	93.6	87.2	93.9	93.0
Model 8	86.0	93.3	91.5	89.5	94.8	94.4	87.7	94.0	92.9
Model 9	86.7	93.7	92.8	89.8	94.9	94.7	88.2	94.3	93.7

According to the cross-sectional information in the analysis table, it can be seen that the recognition effects of medium-sized traffic signs and large-sized traffic signs were better than that of small-sized traffic signs. By comparing Model 1, Model 2, Model 3, and Model 7 longitudinally, it can be seen that the accuracy rate of the lightweight network using the C-Mobilenetv3 proposed in this paper was higher. Compared with MobileNetV1, ShuffleNet, and MobileNetV2, the accuracy rate of small-scale traffic signs was improved by 1.2–3.2%, that of medium-scale traffic signs was improved by 1–2.6%, and that of large-scale traffic signs was improved by 1–2.8%.

A longitudinal comparison of Model 4, Model 5, Model 6, and Model 7 shows that the multiscale fusion module using the feature-interleaving module proposed in this paper had a significant improvement in the recognition effect. Compared with BiFPN, NAS-FPN, and PANet, the accuracy rate of small-scale traffic signs was improved by 1–1.8%, that of medium-scale traffic signs was improved by 0.7–1.5%, and that of large-scale traffic signs was improved by 0.4–1.4%.

A longitudinal comparison of Model 7, Model 8, and Model 9 shows that, compared to CBAM and DANet, the accuracy rate of small-scale traffic signs improved by 0.7–1.5%, that of medium-scale traffic signs improved by 0.2–0.3%, and that of large-scale traffic signs improved by 0.3–0.5%.

3.4. Evaluation of TT100K

3.4.1. Comparative Experiment of Similar Algorithm

To verify the proposed algorithm, Traffic Sign Detection and Recognition Network Combining Multiscale Features and Hybrid Attention Mechanism (MFHA-TSDR), comparative experiments were conducted with similar algorithms, including Faster R-CNN, CornerNet [46], and CenterNet, which were selected based on the TT100K dataset in the same hardware environment, as shown in Table 3, and Figure 9.

Table 3. Algorithm comparison experiments.

	Average Accuracy (%)			Recall (%)			FPS
	Small	**Mid**	**Large**	**Small**	**Mid**	**Large**	
Faster R-CNN	76.5	88.7	88.2	76.2	89.3	88.6	12.0
CornerNet	79.6	91.2	91.0	78.3	90.4	91.8	4.3
CenterNet	82.7	91.0	90.8	78.7	91.5	93.6	7.9
Ours	86.7	95.5	95.0	92.8	97.1	96.5	10.8

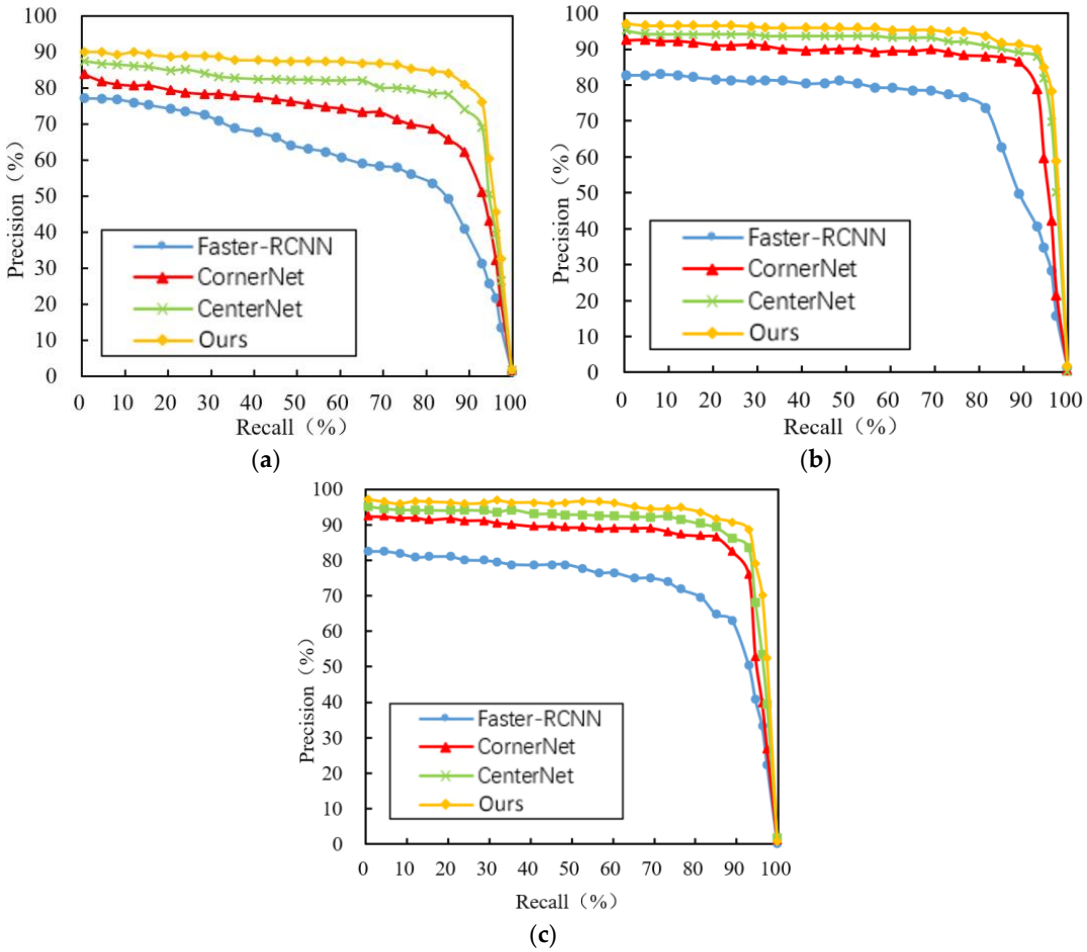

Figure 9. PR curve: (**a**) PR plots for small-sized signs; (**b**) PR plots for medium-sized signs; (**c**) PR plots for large-sized signs.

By analyzing the information in the table, it can be seen that the comprehensive performance of the proposed network for traffic sign detection in this paper was excellent, and the accuracy and recall rates of each scale were well-improved compared to other networks. By comparing the FPS values, it can be seen that the proposed network in this paper also had a great advantage in speed due to the use of a lightweight network. Overall, the proposed network could meet the requirements for both accuracy and speed of traffic sign detection and reduce the false detection rate caused by scale differences.

Analyzing the recognition effect graphs of each algorithm for different scales of traffic signs shows that our proposed algorithm had higher recognition accuracy, while CornerNet and CenterNet were second, and Faster R-CNN was less effective. Especially for the detection of small-scale traffic signs, the detection effect of Faster R-CNN was significantly lower than those of the other two algorithms. In contrast, the proposed network achieved good detection results for all three scales of traffic signs with better robustness.

Above was the quantitative analysis of the three algorithms; next, some of the actual detection result plots were selected for qualitative analysis, as shown in Figure 10. Among them, column a shows the original images in the TT100K dataset, column b shows the detection results of Faster R-CNN, column c shows the detection results of CornerNet, column d shows the detection results of CenterNet, and column shows the detection results of MFHA-TSDR (the proposed algorithm in this paper).

Figure 10. Experiment results.

Comparing the detection results in the first row, we can see that all four methods obtained good detection results for traffic signs with larger scales, while for small-scale targets, Faster R-CNN did not detect them, and there was a missed detection situation. The traffic signs presented in the images in the second row are of large scale, and the results show that all four methods detected them correctly. In the third row, the traffic signs are dense, and Faster R-CNN had a large number of missed detections. CornerNet also had some missed detections, and CenterNet had no missed detections, but the "minimum speed limit of 80 km/h" was wrongly detected as "The proposed network detects all traffic signs completely". The traffic signs in the fourth row of images are traffic signs with serious deformation, and the proposed network could detect the traffic signs with

serious deformation for "drive on the right". In the fifth row, the traffic signs are shaded by trees, and Faster R-CNN failed to detect the "speed limit" and "no parking" traffic signs, while CornerNet and CenterNet failed to detect the "no vehicle" signs that were shaded by leaves. Faster R-CNN failed to detect "speed limit" and "no parking" traffic signs, and CornerNet and CenterNet failed to detect "no long-time or temporary parking" traffic signs that were obscured by leaves. The algorithm proposed in this paper detected "no long-time or temporary parking" traffic signs that were heavily obscured by leaves and shadows and achieved the highest detection rate and accuracy.

In summary, all four algorithms could detect traffic signs well when the traffic sign features in the image were obvious. Faster R-CNN, CornerNet, and CenterNet had missed or false detection when the traffic signs in the images were small-scale, densely laid out, deformed, and obscured, while the proposed network in this paper demonstrated better robustness.

3.4.2. Different Types of Traffic Sign Detection

The traffic sign detection and recognition network proposed in this paper could complete the detection and recognition of 221 types of traffic signs, which mostly covers the traffic signs that may be encountered in daily traffic scenarios. In this paper, we selected some traffic signs that appeared more frequently in the TT100K dataset for data analysis, and the traffic sign detection and recognition network proposed in this paper could achieve more than an 80% recognition rate for all kinds of traffic signs. The results are shown in Table 4.

Table 4. Different classes of detection precision.

Classes	p3	p5	p6	p10	p12	p19	p23	p26	p27	pg
Precision(%)	87.0	94.1	86.5	86.2	76.2	93.2	92.1	90.2	91.2	90.2
Classes	ph4	ph4.5	ph5	pl20	pl30	pl40	pl50	pl60	pl70	pl80
Precision(%)	80.2	90.5	69.5	82.3	90.3	90.0	85.5	90.2	86.8	92.0
Classes	pl100	pl120	pm20	pm30	pm55	pn	pne	pr40	w13	w32
Precision(%)	96.5	97.5	90.5	90.9	93.9	86.9	96.8	93.6	70.2	90.2
Classes	w55	w57	w59	i2	i4	i5	il60	il80	il100	ip
Precision(%)	88.5	91.9	87.5	84.6	86.5	87.3	92.2	96.5	95.3	85.6

According to the information in the table, it can be seen that the detection accuracy of the network proposed in this paper was different for different categories of traffic signs, and most of them could reach more than 85%. However, the accuracy of the target in the category of height limit 5 (ph5) was 69.5%, which was not very good. After analyzing the test, we concluded that the reason was that the signs in this category of height limit were similar to many other signs, including signs of the same type of height limit such as height limit 5.3 (ph5.3), which had similar appearances and extracted similar features, so the recognition accuracy was not high; secondly, for the "120km/h" (pl120), these traffic signs had obvious features and were easy to distinguish, so the detection accuracy was the highest, reaching 97.5%. The detection accuracy of "no entry (pne)" was the second most accurate, reaching 96.8%. Overall, the algorithm proposed in this paper achieved good detection results for all kinds of traffic signs.

4. Discussion

The traffic sign detection method based on a lightweight multiscale feature fusion network proposed in this paper had better advantages than similar algorithms and was tested in real scenarios with good results, but there are still some limitations when applying it to actual assisted driving systems. An automated driving system is large and complex, and further research can be carried out on the following aspects in the future.

(1) The meanings of traffic signs are expressed through shapes, colors, graphics, and words, and traffic signs with the same meanings in different countries or regions may have different shapes, colors, graphics, and words. Therefore, it is a challenge for a network to recognize not only the existing data categories in the training set, but also the data not previously seen. Therefore, in the next step of study, the migration capability of the network should investigated so that the network can be more widely used.
(2) When testing the network, experiments were conducted only under daily conditions, such as normal conditions, the presence of occlusion, and uneven lighting. However, during actual driving, traffic sign detection results in bad weather are more important for driving safety. Therefore, in future study, more challenging environments, such as the presence of fog, haze, rain, or snow, should be selected for experimental analysis.
(3) In the process of driving, a driver pays different attention to the traffic signs he sees, paying attention to ones that require attention and ignoring unwanted information. For example, if a driver is at an intersection and his route is straight, while the traffic signs for a right-turn road may also exist in the driver's field of vision, the driver automatically ignores the information that is not meaningful for behavioral decision making, which reduces the burden of information processing in the human brain. Therefore, the next step of research should try to construct a network that gives different attention to traffic signs in images to further simplify information redundancy and make the network more suitable for complex traffic environments.

5. Conclusions

A traffic sign detection and recognition network based on lightweight multiscale feature fusion and an attention mechanism was proposed to address the problems of poor real-time performance, the single scale of extracted information, and unreasonable allocation of computer resources in the currently proposed convolutional neural network for traffic sign recognition.

(1) To address the problems that traffic sign detection requires high real-time performance and that the existing convolutional neural networks had many redundancies, a lightweight feature extraction network was designed and a key-point detection method was adopted instead of the original anchor frame traversal detection method. In addition, a feature-interleaving module was designed to realize the multiscale extraction of feature information for the problem that traffic sign sizes in a traffic scene map were variable and the semantic information obtained by the existing network was single.
(2) To improve the detection effect when a traffic sign occupied a small image size, was densely arranged, or had too much background information in the image, a hybrid attention module was designed and constructed, which was divided into a spatial attention branch and a channel attention branch, giving different weights to different locations in space and different channels, respectively.
(3) Experiments showed that the algorithm in this paper achieved 85% recognition accuracy for different scale targets and most categories. Compared with Faster R-CNN, ConerNet, and CenterNet, the check-all rate and check-accuracy rate of the algorithm in this paper were significantly higher, and a better real-time performance was achieved. Therefore, the proposed network in this paper was robust, had high recognition accuracy, and achieved a good real-time performance.

Author Contributions: S.L.: Conceptualization, methodology, validation, and writing-original draft preparation; Z.Z.: Validation and writing-original draft preparation; J.T.: Methodology; F.Z.: Validation; X.F. and Q.L.: Writing-review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: We thank Zhejiang Institute of Mechanical and Electrical Engineering Co., Ltd., for supporting this research.

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

References

1. Lu, Q.-C.; Zhang, L.; Xu, P.-C.; Cui, X.; Li, J. Modeling network vulnerability of urban rail transit under cascading failures: A Coupled Map Lattices approach. *Reliab. Eng. Syst. Saf.* **2022**, *221*, 108320. [CrossRef]
2. Zhang, W.; Wang, Q.; Fan, H.; Tang, Y. Contextual and Multi-Scale Feature Fusion Network for Traffic Sign Detection. In Proceedings of the 2020 10th Institute of Electrical and Electronics Engineers International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), Xi'an, China, 10–13 October 2020.
3. Sun, W.; Du, H.; Zhang, X.; Zhao, Y.; Yang, C. Traffic Sign Recognition Method based on Multi-layer Feature CNN and Extreme Learning Machine. *J. Univ. Electron. Sci. Technol. China* **2018**, *47*, 343–349.
4. Wu, L.; Li, H.; He, J.; Chen, X. Traffic Sign Detection Method Based on Faster R-CNN. *J. Phys. Conf. Ser.* **2019**, *1176*, 32045. [CrossRef]
5. Li, H.; Sun, F.; Liu, L.; Wang, L. A Novel Traffic Sign Detection Method via Color Segmentation and Robust Shape Matching. *Neurocomputing* **2015**, *169*, 77–88. [CrossRef]
6. Yu, L.; Xia, X.; Zhou, K. Traffic Sign Detection Based on Visual Co-saliency in Complex Scenes. *Appl. Intell.* **2019**, *49*, 764–790. [CrossRef]
7. Yu, C.; Hou, J.; Hou, C. Traffic Sign Detection Based on Saliency Map and Fourier Descriptor. *Comput. Eng.* **2017**, *43*, 28–34.
8. Zhang, F.; Ji, R.; Jiao, S.; Qi, K. A Novel Saliency Computation Model for Traffic Sign Detection. In Proceedings of the 2017 2nd International Conference on Image, Vision and Computing (ICIVC), Chengdu, China, 2–4 June 2017.
9. Yin, S.; Deng, J.; Zhang, D.; Du, J.-Y. Traffic Sign Recognition Based on Deep Convolutional Neural Network. *Optoelectron. Lett.* **2017**, *13*, 476–480. [CrossRef]
10. Zhu, Z.; Liang, D.; Zhang, S.; Huang, X.; Li, B.; Hu, S. Traffic-Sign Detection and Classification in the Wild. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
11. Xie, K.; Ge, S.; Ye, Q.; Luo, Z. Traffic Sign Recognition Based on Attribute-Refinement Cascaded Convolutional Neural Networks. In Proceedings of the Pacific Rim Conference on Multimedia, Xi'an, China, 15–16 September 2016.
12. Zhu, Y.; Zhang, C.; Zhou, D.; Wang, X.; Bai, X.; Liu, W. Traffic Sign Detection and Recognition Using Fully Convolutional Network Guided Proposals. *Neurocomputing* **2016**, *214*, 758–766. [CrossRef]
13. Zhang, Z.; Zhou, X.; Chan, S.; Chen, S.; Liu, H. Faster R-CNN for Small Traffic Sign Detection. In *CCF Chinese Conference on Computer Vision*; Springer: Singapore, 2017; pp. 155–165.
14. Zuo, Z.; Yu, K.; Zhou, Q.; Wang, X.; Li, T. Traffic Signs Detection Based on Faster R-CNN. In Proceedings of the 2017 IEEE 37th International Conference on Distributed Computing Systems Workshops (ICDCSW), Atlanta, GA, USA, 5–8 June 2017; pp. 286–288.
15. Luo, H.; Yang, Y.; Tong, B.; Wu, F.; Fan, B. Traffic Sign Recognition Using A Multi-Task Convolutional Neural Network. *IEEE Trans. Intell. Transp. Syst.* **2018**, *19*, 1100–1111. [CrossRef]
16. Zhu, Y.; Liao, M.; Yang, M.; Liu, W. Cascaded Segmentation-Detection Networks for Text-Based Traffic Sign Detection. *IEEE Trans. Intell. Transp. Syst.* **2018**, *19*, 209–219. [CrossRef]
17. Cheng, P.; Liu, W.; Zhang, Y.; Ma, H. LOCO: Local Context Based Faster R-CNN for Small Traffic Sign Detection. In Proceedings of the International Conference on Multimedia Modeling, Bangkok, Thailand, 5–7 February 2018; Springer: Cham, Switzerland; pp. 329–341.
18. Pei, S.; Tang, F.; Ji, Y.; Fan, J.; Ning, Z. Localized Traffic Sign Detection with Multi-scale Deconvolution Networks. In Proceedings of the 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), Tokyo, Japan, 23–27 July 2018; pp. 1–7.
19. Li, J.; Liang, X.; Wei, Y.; Xu, T.; Feng, J.; Yan, S. Perceptual Generative Adversarial Networks for Small Object Detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1951–1959.
20. Heng, L.; Qing, K. Traffic Sign Image Synthesis with Generative Adversarial Networks. In Proceedings of the 24th International Conferenceon Pattern Recognition, Beijing, China, 20–24 August 2018; pp. 2540–2545.
21. Xiang, C.; Zhang, L.; Tang, Y.; Zou, W.; Xu, C. MS-CapsNet: A Novel Multi-Scale Capsule Network. *IEEE Signal Process. Lett.* **2018**, *25*, 1850–1854. [CrossRef]
22. Zhang, J.; Xie, Z.; Sun, J.; Zou, X.; Wang, J. A Cascaded R-CNN With Multiscale Attention and Imbalanced Samples for Traffic Sign Detection. *IEEE Access* **2020**, *42*, 29742–29754. [CrossRef]
23. Yuan, Y.; Zhi, X.; Qi, W. An Incremental Framework for Video-Based Traffic Sign Detection, Tracking, and Recognition. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 1918–1929. [CrossRef]

24. Lee, H.; Kim, K. Simultaneous Traffic Sign Detection and Boundary Estimation using Convolutional Neural Network. *IEEE Trans. Intell. Transp. Syst.* **2018**, *19*, 1652–1663. [CrossRef]
25. Kong, X.; Zhang, J.; Deng, L.; Liu, Y. Research Advances on Vehicle Parameter Identification Based on Machine Vision. *China J. Highw. Transp.* **2021**, *34*, 13–30.
26. Zhou, K.; Zhan, Y.; Fu, D. Learning Region-Based Attention Network for Traffic Sign Recognition. *Sensors* **2021**, *21*, 686. [CrossRef]
27. Lian, J.; Yin, Y.; Li, L.; Wang, Z.; Zhou, Y. Small Object Detection in Traffic Scenes Based on Attention Feature Fusion. *Sensors* **2021**, *21*, 3031. [CrossRef]
28. Krizhevsky, A.; Sutskever, I.; Hinton, G. *ImageNet Classification with Deep Convolutional Neural Networks*; NIPS. Curran Associates Inc.: New York, NY, USA, 2012; pp. 1097–1105.
29. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556, 1–14.
30. 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.
31. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely Connected Convolutional Networks. *arXiv* **2018**, arXiv:1608.06993, 1–9.
32. Iandola, F.N.; Han, S.; Moskewicz, M.W.; Ashraf, K.; Dally, W.J.; Keutzer, K. SqueezeNet: AlexNet-level Accuracy with 50× Fewer Parameters and <0.5MB Model Size. *arXiv* **2016**, arXiv:1602.07360, 1–13.
33. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* **2017**, arXiv:1704.04861, 1–9.
34. Zhang, M.; Xu, S.; Song, W.; He, Q.; Wei, Q. Lightweight underwater object detection based on yolo v4 and multi-scale attentional feature fusion. *Remote Sens.* **2021**, *13*, 4706. [CrossRef]
35. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. *arXiv* **2017**, arXiv:1707.01083, 1–9.
36. Chollet, F. Xception: Deep Learning with Depthwise Separable Convolutions. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1800–1807.
37. Howard, A.; Sandler, M.; Chen, B.; Wang, W.; Chen, L.-C.; Tan, M.; Chu, G.; Vasudevan, V.; Zhu, Y.; Pang, R.; et al. Searching for MobileNetV3. *arXiv* **2019**, arXiv:1905.02244, 1–11.
38. Howard, A.; Zhmoginov, A.; Chen, L.C.; Sandler, M.; Zhu, M. Inverted Residuals and Linear Bottlenecks: Mobile Networks for Classification, Detection and Segmentation. *arXiv* **2019**, arXiv:1801.04381, 1–14.
39. Tan, M.; Pang, R.; Le, Q.V. EfficientDet: Scalable and Efficient Object Detection. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 10778–10787.
40. Guo, M.-H.; Xu, T.-X.; Liu, J.-J.; Liu, Z.-N.; Jiang, P.-T.; Mu, T.-J.; Zhang, S.-H.; Martin, R.R.; Cheng, M.-M.; Hu, S.-M. Attention Mechanisms in Computer Vision: A survey. *Comput. Vis. Media* **2022**, *8*, 331–368. [CrossRef]
41. Duan, K.; Bai, S.; Xie, L.; Qi, H.; Huang, Q.; Tian, Q. Centernet: Keypoint Triplets for Object Detection. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 6569–6578.
42. Ghiasi, G.; Lin, T.Y.; Le, Q.V. NAS-FPN: Learning Scalable Feature Pyramid Architecture for Object Detection. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 7029–7038.
43. 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, Salt Lake City, UT, USA, 18–22 June 2018; pp. 8759–8768.
44. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. CBAM: Convolutional Block Attention Module. *arXiv* **2018**, arXiv:1807.06521, 1–17.
45. Fu, J.; Liu, J.; Tian, H.; Li, Y.; Bao, Y.; Fang, Z.; Lu, H. Dual Attention Network for Scene Segmentation. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 3141–3149.
46. Law, H.; Deng, J. CornerNet: Detecting Objects as Paired Keypoints. *Int. J. Comput. Vis.* **2020**, *128*, 642–656. [CrossRef]